Definiens

Developer XD 2.0.4

Reference Book

DEEPER INSIGHTS
FASTER RESULTS
BETTER DECISIONS

www.definiens.com

Definiens Documentation:

Developer XD 2.0.4

Reference Book

Imprint

Published by:

Definiens AG, Bernhard-Wicki-Straße 5, 80636 München, Germany
Phone: +49 89 2311 800 • Fax: +49 89 2311 8090
Web: www.definiens.com

Dear User,

Thank you for using Definiens software. We appreciate being of service to you with image intelligence solutions. At Definiens we constantly strive to improve our products. We therefore appreciate all comments and suggestions for improvements concerning our software, training, and documentation. Feel free to contact us via web form on the Definiens support website www.definiens.com/support

Thank you.

Legal Notes

Regulatory affairs

Under certain circumstances the solutions and applications developed using Definiens Developer XD may fall under specific regulations (e.g. medical device or IVD regulations) in your country. Please ensure that you check and follow local regulations before using or taking your solution or application into commerce. If you require specific data about Definiens Developer XD, please contact your dealer or our sales staff.

* * *

Typeset by Wikipublisher

Day of print: 05 September 2012

Contents

Introduction

Symbols and Expressions

Basic Mathematical Notations

Basic mathematical symbols used in expressions.

$\therefore$	Therefore
$\varnothing$	Empty set
$a \in A$	a is an element of a set A
$b \notin B$	b is not an element of set B
$A \subset B$	Set A is a proper subset of set B
$A \not\subset B$	Set A is not a proper subset of set B
$A \subseteq B$	Set A is a subset of set B
$A \cup B$	Union of sets A and B
$A \cap B$	Intersection of sets A and B
$A \setminus B$	A symmetric difference of sets A and B
$\#A$	The size of set A
$\exists$	Exists, at least one
$\forall$	For all
$\Rightarrow$	It follows
$\Leftrightarrow$	Equivalent
$\sum_{i=1}^{u}$	Sum over index i
$[a,b]$	Interval with $\{x \mid a \leq x \leq b\}$

Image Layer and Scene

Expressions used to represent image layers and scenes.

$k = 1, \ldots, K$	Image layer k
$t = 1, \ldots, T$	Thematic layer t
(x, y, z, t)	Co-ordinates of a pixel/voxel
u	Size of a pixel/voxel in co-ordinate system unit
(sx, sy, sz, st)	Scene extent of a scene s
$c_k(x, y, z, t)$	Intensity value of image layer k at pixel/voxel (x, y, z, t)
$c_k^{\max}$	Brightest possible intensity value of image layer k
$c_k^{\min}$	Darkest possible intensity value of image layer k
c_k^{range}	Data range of image layer k
$\bar{c}_k$	Mean intensity of image layer k
σ_k	Standard deviation of intensity values of image layer k
$N_4(x, y)$	4-pixel/voxel neighbors (x, y)
$N_8(x, y)$	8-pixel/voxel neighbors (x, y)
$N_6(x, y, z)$	6-pixel/voxel neighbors (x, y, z)
$N_{26}(x, y, z)$	26-pixel/voxel neighbors (x, y, z)

Region

Expressions used to represent regions.

R	Region R
(x_G, y_G, z_G, t_G)	Origin of region R
$[R_x, R_y, R_z, R_t]$	Extent of region R
$c_k^{\max}(R)$	Brightest possible intensity value of image layer k within region R
$c_k^{\min}(R)$	Darkest possible intensity value of image layer k within region R
$\bar{c}_k(R)$	Mean intensity of image layer k within region R
$\sigma_k(R)$	Standard deviation of image layer k within region R

Pixel Set

Expressions representing layer intensity.

S	Set of pixels
$\bar{c}_k(S)$	Mean intensity of image layer k of a set S
$\sigma_k(S)$	Standard deviation of intensity values of image layer k of a set S
$\bar{c}(S)$	Brightness
w_k^B	Brightness weight of image layer k
$\bar{\Delta}_k(v, O)$	Mean difference of an image object v to image objects in a set O

Image Object

Expressions representing an image object as a set of pixels.

u, v	Image object
O	Set of image objects
P_v	Set of pixels of an image object v
$\#P_v$	Total number of pixels contained in P_v
$\bar{c}_k(v)$	Mean intensity of image layer k forming an image object v
$\sigma_k(v)$	Standard deviation of intensity values of image layer k of all pixels forming image object v
P_v^{Inner}	Set of inner border pixels of P_v
P_v^{Outer}	Set of outer border pixels of P_v
$v_i(P)$	Image object v in image object level i in pixel P
$m(v)$	Classification of image object v
B_v	Bounding box of an image object v
$B_v(d)$	Extended bounding box of an image object v with distance d
$x_{\text{min}}(v)$	Minimum x co-ordinate of v
$x_{\text{max}}(v)$	Maximum x co-ordinate of v
$y_{\text{min}}(v)$	Minimum y co-ordinate of v
$y_{\text{max}}(v)$	Maximum y co-ordinate of v
b_v	Image object border length
$b(v, u)$	Length of common border between v and u

Image Objects Hierarchy

Expressions used to represent relations between image objects.

u, v	Image objects
$U_v(d)$	Superobject of an image object v at a distance d
$S_v(d)$	Sub-objects of an image object v at a distance d
$V_i, i = 1, \ldots, n$	Image object level
N_v	Direct neighbors of an image object v
$N_v(d)$	Neighbors of an image object v at a distance d
$e(u, v)$	Neighborhood relation between the image objects u and v

Class-Related Set

Expressions used to represent relations between classes.

M	Set of classes $M = \{m1, \ldots, m_a\}$
m	A class, $(m \in M)$
$N_v(d,m)$	Neighbors classified as m within a distance d
$S_v(d,m)$	Sub-objects classified as m with hierarchical distance d
$U_v(d,m)$	Superobject classified as m with hierarchical distance d
$V_i(m)$	All image objects at level i classified as m
$\phi(v,m)$	Fuzzy membership value of image object v to class m
$\varphi(v,m)$	Stored membership value of image object v to class m
$P_i(R,m)$	Pixels classified as m on image object level i within region R

Co-ordinate Systems Used in Definiens Software

Definiens software uses three co-ordinate systems:

- The pixel co-ordinate system is used for identifying pixel positions within a map
- The user co-ordinate system allows the use of geocoding information within a map
- The internal pixel co-ordinate system is used only for internal calculations by the Analysis Engine software.

Pixel Co-ordinate System

The pixel co-ordinate system is used to identify pixel position within an image. It is used for calculating position features such as x -center and y -center.

This co-ordinate system is oriented from bottom to top and from left to right. The origin position is (0, 0), which is at the bottom-left corner of the image. The co-ordinate is defined by the offset of the bottom-left corner of the pixel from the origin.

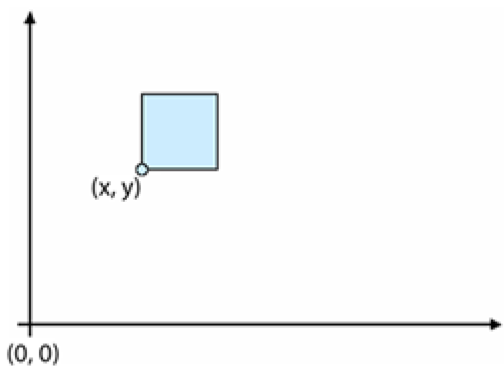


Figure 1. The pixel coordinate system

Pixel Co-ordinate Definition

$x_{min} + 1 = x_{max}$ $y_{min} + 1 = y_{max}$

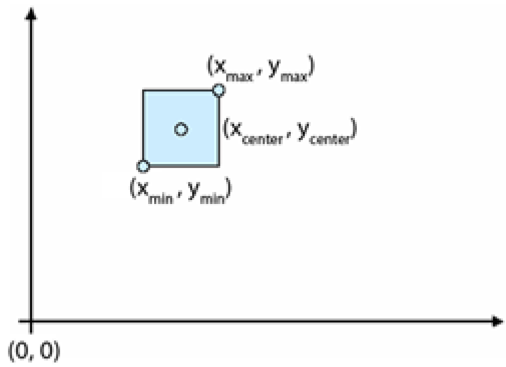


Figure 2. Coordinates of a pixel

User Co-ordinate System

The user co-ordinate system enables the use of geocoding information within a scene. The values of the separate user co-ordinate system are calculated from the pixel co-ordinate system. In the user interface, the user co-ordinate system is referred to as the co-ordinate system.

This co-ordinate system is defined by geocoding information:

- The bottom-left X position
- The bottom-left Y position
- Resolution the size of a pixel in co-ordinate system unit. For example, if the co-ordinate system is metric, the resolution is the size of a pixel in meters. If the co-ordinate system is lat/long, then the resolution is the size of a pixel in degrees
- Co-ordinate system name
- Co-ordinate system type.

The origin of the co-ordinate system is at the bottom-left corner of the image (x_0, y_0) . The co-ordinate defines the position of the bottom-left corner of the pixel within the user co-ordinate system.

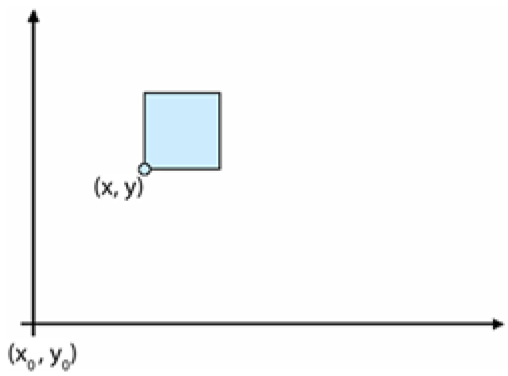


Figure 3. The user coordinate system

To convert a value from the pixel co-ordinate system to the user co-ordinate system and back, the following transformations are valid, where (x, y) are the co-ordinates in user co-ordinate system and u is the pixel size in units:

- $x = x_0 + x_{\text{pixel}} \times u$
- $x_{\text{pixel}} = (x - x_0)/u$
- $y = y_0 + y_{\text{pixel}} \times u$
- $y_{\text{pixel}} = (y - y_0)/u$

Image Layer Related Features

Scene

A scene is a collection of combined input image data, as represented in Definiens software. A scene comprises at least one image layer. In addition, it can include more image layers and thematic layers. You are likely to encounter this concept when importing image or other data into Definiens software.

A scene can also include metadata, such as image creation information or geo-information, and hold settings such as layer aliases, unit information, or geocoding. Within Definiens software, image layers, thematic layers, and metadata are loaded by reference to the respective data files. Each scene is represented by a related map.

- A 2D image in a 3D data set is called a slice.
- A 2D image in a time series data set is called a frame.
- A 4D data set consists of a series of frames where each frame is a 3D data set.

Depending on the related data set, a scene can be one of the following:

Data Set	Scene
2D image	A rectangular area in a 2D space
3D data set	A rectangular volume in a 3D space
4D data set	A series of rectangular volumes in a 4D space
Time series data set	A series of rectangular areas in a 2D + time space

Extents

A scene has an origin (x_0, y_0, z_0, t_0) , the extent sx in the x -direction, the extent sy in the y -direction, the extent sz in the z -direction, and the extent st in the t -direction.

The expression is (sx, sy, sz, st) , where:

- $\#(\text{pixels})_x$ is the number of pixels in the x -direction
- $\#(\text{pixels})_y$ is the number of pixels in the y -direction
- u is the size of a slice pixel in the co-ordinate system unit
- u_{slices} is the spatial distance between slices in the co-ordinate system unit
- u_{frames} is the temporal distance between slices in the time co-ordinate unit

The formulas for scene extent are:

- $sx = \#(\text{pixels})_x \times u$
- $sy = \#(\text{pixels})_x \times u$
- $sz = \#(\text{slices}) \times u_{\text{slices}}$
- $st = \#(\text{frames}) \times u_{\text{frames}}$

Co-ordinates

- x^{pxl} is the x offset in pixels
- y^{pxl} is the y offset in pixels
- z^{slices} is the z offset in slices
- t^{frames} is the t offset in frames

Co-ordinate formulas are:

- $x = x_0^{\text{geo}} + x^{\text{pxl}} \times u$
- $y = y_0^{\text{geo}} + y^{\text{pxl}} \times u$
- $z = z_0^{\text{geo}} + z^{\text{slices}} \times u_{\text{slices}}$
- $t = t_0^{\text{geo}} + t^{\text{frames}} \times u_{\text{frames}}$

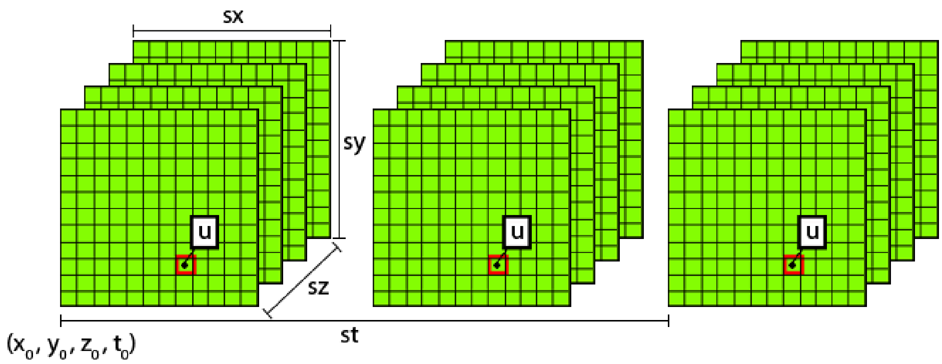


Figure 4. Coordinates of a multidimensional scene with four slices and three frames

Layers

Scenes can consist of an arbitrary number of image layers $(k = 1, ..., K)$ and thematic layers $(t = 1, ..., T)$.

Maps

A map represents the combination of a scene and an image object hierarchy. It is the structure that represents the data that the rule set operates on. Definiens software can deal with multiple maps and provides algorithms to rescale and copy maps, as well as to synchronize image objects between them.

Region

A region is a subset definition applicable to maps. A region is specified by its origin, which is the offset position of the region and its extent $[R_x, R_y, R_z, R_t]$ in all co-ordinate dimensions: $(x_G, y_G, z_G, t_G), [R_x, R_y, R_z, R_t]$

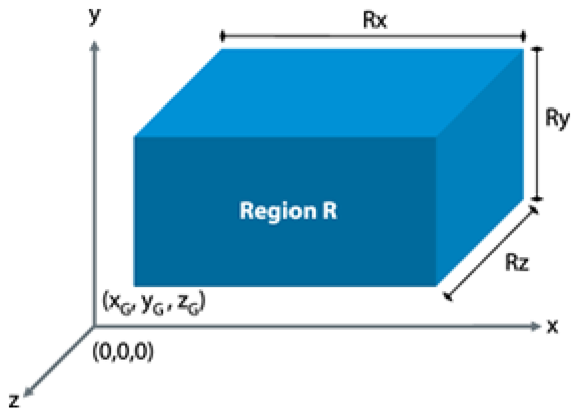


Figure 5. A three-dimensional region

Depending on map dimensions, the following region types exist:

Type	Expression
2D	$(x_G, y_G), [R_x, R_y]$
3D	$(x_G, y_G, z_G), [R_x, R_y, R_z]$
4D	$(x_G, y_G, z_G, t_G), [R_x, R_y, R_z, R_t]$
Time series	$(x_G, y_G, t_G), [R_x, R_y, R_t]$

Image Layer

A scene refers to at least one image layer of an image file. Image layers are referenced by a layer name (string), which is unique within a map.

The pixel/voxel value that is the layer intensity of an image layer k at pixel/voxel (x, y, z, t) is denoted as $c_k(x, y, z, t)$. The dynamic range of image layers is represented as follows:

- $c_k^{\min}$ is the smallest possible intensity value of an image layer k
- $c_k^{\max}$ is the largest possible intensity value of an image layer k
- c_k^{range} is the data range of image layer k with $c_k^{\text{range}} = c_k^{\max} - c_k^{\min}$

The dynamic range depends on the image layer data type. The supported image layer data types[1] are:

1. Full support for image layer data types is dependent on drivers

Type	$c_k^{\min}$	$c_k^{\max}$	c_k^{range}
8-bit unsigned (int)	0	255	256
16-bit unsigned (int)	0	65535	65536
16-bit signed (int)	−32767	32767	65535
32-bit unsigned (int)	0	4294967295	4294967296
32-bit signed (int)	−2147483647	2147483647	4294967295
32-bit float	1.17×10^{-38}	3.40×10^{38}	n/a

The mean value of all pixel/voxels in a layer is computed by:

$$\bar{c}_k = \frac{1}{sx \times sy} \sum_{(x,y)} c_k(x,y)$$

The standard deviation of all pixel/voxels in a layer is computed by:

$$\sigma_k = \sqrt{\frac{1}{sx \times sy}\left(\sum_{(x,y)} \left(c_k(x,y)\right)^2 - \frac{1}{sx \times sy} \sum_{(x,y)} c_k(x,y) \sum_{(x,y)} c_k(x,y)\right)}$$

Neighborhood

On raster pixel/voxels there are two ways to define the neighborhood: as 6-neighborhood or 26-neighborhood.

Image Layer Intensity on Pixel Sets

A fundamental measurement on a pixel set S and an image object v is the distribution of the layer intensity. Firstly, the mean intensity within the set is defined by:

$$\bar{c}_k(S) = \frac{1}{\#s} \sum_{(x,y)\exists S} c_k(x,y)$$

The standard deviation is defined as:

$$\sigma_k(S) = \sqrt{\frac{1}{\#S}\left(\sum_{(x,y)\exists S} (c_k(x,y))^2 - \frac{1}{\#S} \sum_{(x,y)\exists S} c_k(x,y) \sum_{(x,y)\exists S} c_k(x,y)\right)}$$

An overall intensity measurement is given by the brightness which is the mean value of $\bar{c}_k(S)$ for selected image layers.

$$\bar{c}(S) = \frac{1}{w^B} \sum_{k=1}^{K} w_k^B \bar{c}_k(S)$$

If v is an image object and O a set of other image objects, then the mean difference of the objects within O to an image object v is calculated by:

$$\bar{\Delta}_k(v,0) = \frac{1}{w} \sum_{u \varepsilon 0} w_u\left(\bar{c}_k(v) - \bar{c}_k(u)\right)$$

Image Object Related Features

Image Object

An image object is a set of pixels. The set of pixels of an image object v is denoted as P_v.

Image Object Size

At a basic level, the size of an image object is measured by the number of pixels in image object v. The number of pixels is denoted by $\#P_v$, the cardinality of the respective pixel set.

Bounding Box of an Image Object

The bounding box B_v of an image object v is the most common form of measurement of the extent of an image in a scene. It is the smallest rectangular area that encloses all pixels of v along x- and y-axes. The bounding box is therefore defined by the minimum and maximum values of the x and y co-ordinates of an image object v $\left(x_{\min}(v)\ , x_{\max}(v)\right.$ and $\left.y_{\min}(v)\ , y_{\max}(v)\right)$.

The extended bounding box $B_v(d)$ is created by enlarging the bounding box with the same number of pixels in all directions.

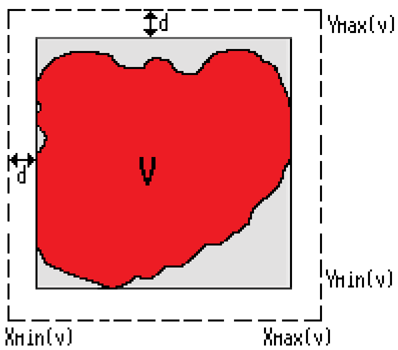


Figure 6. Bounding box of an image object v

Pixel Neighborhood and Connectivity of Image Objects

Two pixels are considered to be spatially connected if they neighbor each other in the pixel raster. On 2D raster images, the neighborhood between pixels can be defined as a 4- or 8-neighborhood:

With reference to the green pixel, the light-blue pixels are the 4-neighborhood. In combination with the dark blue pixels, they form the 8- or diagonal neighborhood of the green pixel. Definiens software uses the 4-neighborhood for all spatial neighborhood calculations. As it assumes Image objects to be a set of spatially connected pixels, an image

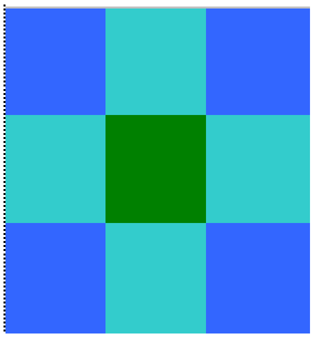


Figure 7. 4- and 8-neighborhoods

object is defined as connected if it is connected according to the 4-neighborhood principle.

Neighborhood Relations Between Image Objects

Two image objects are considered neighbors if they contain pixels that neighbor each other according to the 4-neighborhood principle. In other words, two image objects u and v are considered neighboring each other if this is at least on pixel $(x,y) \in P_v$ and one pixel $(x',y') \in P_u$ so that (x',y') is part of $N_4(x,y)$. The set of all image objects neighboring v is denoted by $N_v(d)$:

$$N_v = \{u \in V_i : \exists (x,y) \in P_v \exists (x',y') \in P_u : (x',y') \exists N_4(x,y)\}$$

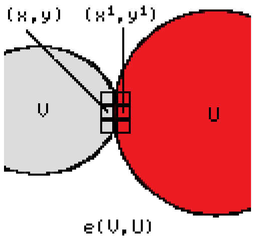


Figure 8. Neighborhood relation

The entire border line between u and v is called the neighborhood relation and is represented as {$e(u,v).$ The neighborhood relations between image objects are automatically determined and maintained by the Definiens software.

Border of an Image Object

The border length b_v of a two-dimensional image object v is calculated by the number of the elementary pixel borders. The border length of a three-dimensional image object is calculated from the borders of image object slices multiplied by the slice distance. The border of image object slices is counted by the number of the elementary pixel borders.

Similarly, the border length $b(v,u)$ of the neighborhood relation between two image objects v and u is calculated from the common borders of image object slices multiplied by the slice distance. The border of image object slices is counted by the number of the elementary pixel borders along the common border. Image objects are basically pixel sets. The number of pixels belonging to an image object v and its pixel set P_v is denoted by $\#P_v$.

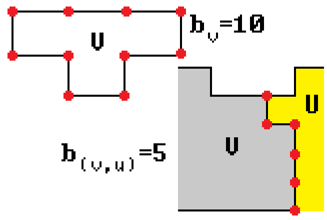


Figure 9. Border length of an image object v or between two objects v , u

Inner Border The set of all pixels in P_v belonging to the inner border pixels of an image object v is defined by P_v^{Inner} with $P_v^{\text{Inner}} = \{(x,y) \in P_v : \exists(x',y') \in N_4(x,y) : (x',y') \notin P_v\}$

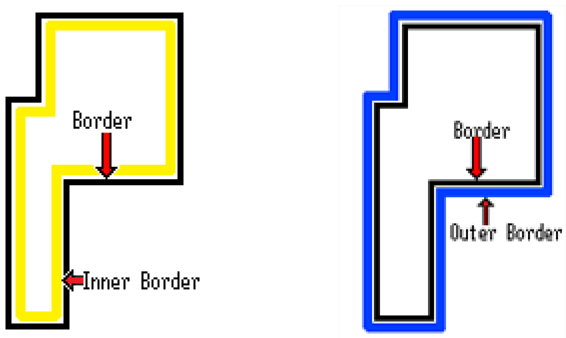


Figure 10. Inner and outer borders of an image object

Outer Border The set of all pixels in Pv belonging to the outer border pixels of an image object v is defined by P_v^{Outer} with $P_v^{\text{Outer}} = \{(x,y) \notin P_v : \exists(x',y') \in N_4(x,y) : (x',y') \in P_v\}$

Neighborhood Relations of 3D Image Objects RfB

Image objects in Definiens Developer XD 2.0.4 can be two-dimensional or three-dimensional. The two-dimensional neighborhood concept can be easily extended to three dimensions. That is, for 3D image objects, there is a 6-neighborhood resulting from the two additional neighboring pixels in the z dimension. This diagonal 3D neighborhood is also referred to as the 26-neighborhood.

3D objects are usually 6-connected and may extend across several slices. Several separated parts within a single slice are also possible. However, these parts must be connected along parts of other slices. If you look at a 3D object in the map view using one of the three 2D projections, a two-dimensional slice is displayed on the plane. You can use the MPR view or the 3D view to get a better impression of the shape of a 3D image object.

The software manages two- and three-dimensional neighborhood relations separately. This allows a correct determination of neighborhood relations in mixed mode situations where 2D and 3D image objects co-exist in the same scene. 2D image objects can only be connected by two-dimensional neighborhood relations while 3D image objects can be connected by both two-dimensional and three-dimensional neighborhood relations.

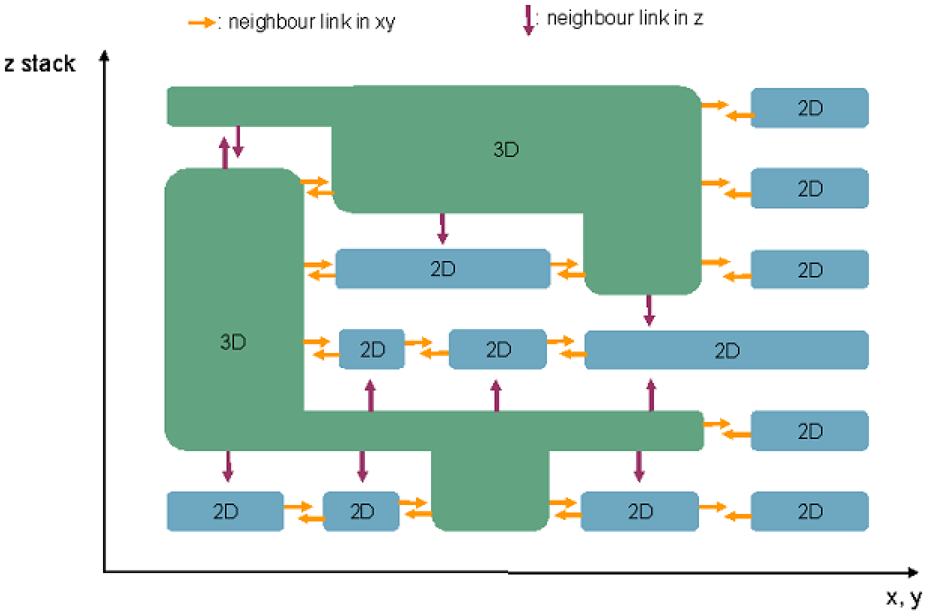


Figure 11. 2D and 3D neighborhood relations

Tracking Image Objects Through Time

There is no special image object type to represent the movement of an image object in time. Instead, image object links are used to track identical image objects at different points in time (a time series).

Disconnected Image Objects

In addition to normal 2D and 3D image objects, Definiens Developer XD 2.0.4 allows you to work with image objects that are spatially disconnected. While a connected image object covers one contiguous region of a scene, a disconnected image object can consist of arbitrary pixels in two or more potentially disconnected parts within a scene.

It is important to know that image objects can be defined as disconnected even though they are not disconnected in reality. An image object is defined as disconnected if it lacks information about its spatial connectivity. The major motivation for this lack of information is the high calculation effort that is necessary to ensure spatial connectivity. If, for example, you remove some pixels from an image object, it may divide into several parts. To connect the resulting image objects, a special algorithm needs to analyze the remaining objects and separate them properly into several sub-image objects. If the resulting image objects remain disconnected, they can be simply marked as there is no need for an analysis. Disconnected image objects are therefore useful for fast object processing when spatial information is not relevant. If you use threshold segmentation, for example, to separate an image into two disconnected image objects, you can save a lot of computing time and memory since the number of image objects is considerably reduced.

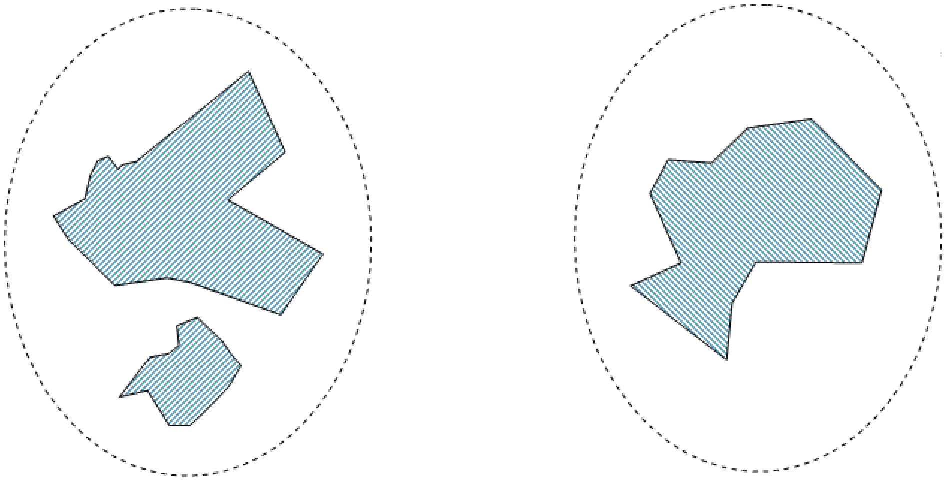


Figure 12. Disconnected image objects (left) and spatially connected image object (right)

The main purpose of working with disconnected image objects is the representation of simple "pixel crowds", where several properties that can be measured for normal image objects might not make sense or are meaningless. You should therefore note that the following information is not available for disconnected image objects:

- Neighborhood relations and all features based on these relations
- Shape features
- Polygons.

Image Object Hierarchy

Image objects are organized into levels, where each object on each level creates a partition of the scene S . This can be described using the following fundamental conditions. The totality of all image objects covers the entire scene:

$$\underset{v \exists V_i}{U} P_v = \{(x,y)\}$$

There are no image objects that overlap (having a non-empty intersection):

$$\underset{u,v \exists V_i}{\forall} P_u \cap P_v = \varnothing$$

The image object levels are hierarchically structured. This means that all image objects on a lower level are complete contained in exactly one image object of a higher level.

$$\underset{i<j}{\forall} \quad \underset{v \exists V_i}{\forall} \quad \underset{u \exists V_i}{\exists} \quad P_v \quad \subset \quad P_u$$

Level Distance

The level distance represents the hierarchical distance between image objects on different levels in the image object hierarchy. Starting from the current image object level, the level distance indicates the hierarchical distance of image object levels containing the respective image objects (sub-objects or superobjects).

Since each object has exactly 1 or 0 superobjects on the higher level, the superobject of v with a level distance d can be denoted as $U_v(d)$. Similarly, all sub-objects with a level distance d are denoted as $S_v(d)$.

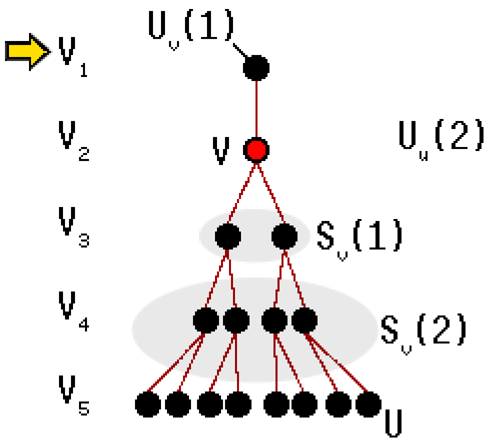


Figure 13. Image object hierarchy

Spatial Distance

The spatial distance represents the distance between image objects on the same level in the image object hierarchy. If you want to analyze neighborhood relations between image objects on the same image object level in the image object hierarchy, the feature distance expresses the spatial distance (in pixels) between the image objects. The default value is 0 (in other words, only neighbors that have a mutual border are considered). The set of all neighbors within a distance d are denoted by $N_v(d)$.

Distance Measurements Many features enable you to enter a spatial distance parameter. Distances are usually measured in pixel units. Because exact distance measurements between image objects are very processor-intensive, Definiens software uses approximation approaches to estimate the distance between image objects.

There are two different approaches: the center of gravity and the smallest enclosing rectangle. You can configure the default distance calculations.

Center of Gravity The Center of Gravity approximation measures the distance between the center of gravity between two image objects. This measure can be computed very efficiently but it can be inaccurate for large image objects.

Smallest Enclosing Rectangle The smallest enclosing rectangle approximation tries to correct the center of gravity approximation by using rectangular approximations of the image object to adjust the basic measurement delivered by the center of gravity.

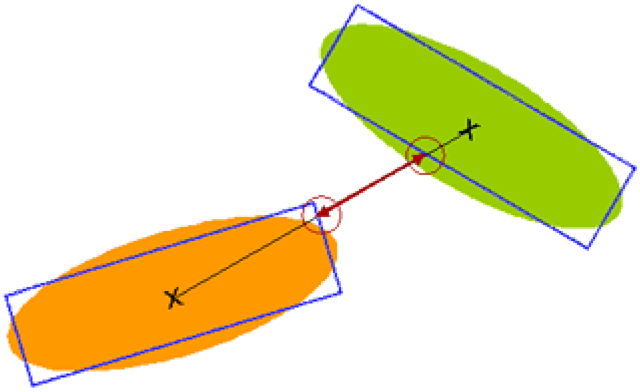


Figure 14. Distance calculation between image objects. (The black line is the center of gravity approximation. The red line is the smallest enclosing rectangle approximation)

We recommend using the center of gravity distance for most applications although the smallest enclosing rectangle may give more accurate results. A good strategy for exact distance measurements is to use center of gravity and try to avoid large image objects, for example, by creating border objects. To avoid performance problems, restrict the total number of objects involved in distance calculations to a small number.

You can edit the distance calculation in the algorithm parameters of the Set Rule Set Options algorithm and set the Distance Calculation option to your preferred value.

Class-Related Features

Class-Related Sets

Let $M = \{m1, \dots, m_a\}$ be a set of classes with m being a specific class and $m \in M$. Each image object has a fuzzy membership value of $\phi(v,m)$ to class m . In addition each image object also carries the stored membership value $\varphi(u,m)$ that is computed during execution of the last classification algorithm. By restricting a set of image objects O to only the image object that belong to class m , various class-related features can be computed:

$N_v(d,m) = \{u \in N_v(d) : \tilde{\varphi}(u,m) = 1\}$	Neighbors of class
$S_v(d,m) = \{u \in S_v(d) : \tilde{\varphi}(u,m) = 1\}$	Sub-objects of class
$U_v(d,m) = \{u \in U_v(d) : \tilde{\varphi}(u,m) = 1\}$	Superobjects of class
$V_i(m) = \{u \in V_i(m) : \tilde{\varphi}(u,m) = 1\}$	Image object levels of class
$P_i(R,m) = \{p \in R : (v_i(p)) = m\}$	Pixels of class

Example: The mean difference of layer k to a neighbor image object within a distance d and that image object belongs to a class m is defined as $\bar{\Delta}_k(v, N_v(d,m))$.

Shape-Related Features

Many of the Definiens form features are based on the statistics of the spatial distribution of the voxels that form an 3D image object. As a central tool to work with these statistics, Definiens Developer XD 2.0.4 uses the covariance matrix:

Parameters

- X : x co-ordinates of all voxels forming the image object
- Y : y co-ordinates of all voxels forming the image object
- Z : z co-ordinates of all voxels forming the image object.

Expression

$$C = \begin{pmatrix} \mathrm{Var}(X) & \mathrm{Cov}(XY) & \mathrm{Cov}(XZ) \\ \mathrm{Cov}(XY) & \mathrm{Var}(Y) & \mathrm{Cov}(YZ) \\ \mathrm{Cov}(XZ) & \mathrm{Cov}(YZ) & \mathrm{Var}(Z) \end{pmatrix}$$

Another frequently used technique to derive information about the form of image objects is the bounding box approximation. Such a bounding box can be calculated for each image object and its geometry can be used as the first clue to the image object itself.

The main information provided by the bounding box is its length a , its width b , its thickness c , its volume $a \times b \times c$ and its degree of filling f , which is the volume V filled by the image object divided by the total volume $a \times b \times c$ of the bounding box.

Shape Approximations Based on Eigenvalues

The shape approximations based on eigenvalues measures the statistical distribution of the voxel co-ordinates (x, y, z) of a set P_v .

The center of gravity of the set P_v is:

$$x_{\text{center}} = \frac{1}{\#P_v} \sum_{(x,y,z)} x$$

$$y_{\text{center}} = \frac{1}{\#P_v} \sum_{(x,y,z)} y$$

$$z_{\text{center}} = \frac{1}{\#P_v} \sum_{(x,y,z)} z$$

The variances of voxel co-ordinates are:

$$C_{xx} = \frac{1}{\#P_v} \sum_{(x,y,z)} x_i^2 - \left(\frac{1}{\#P_v} \sum_{(x,y,z)} x_i \right)^2 = E_{xx} - E_x^2$$

$$C_{yy} = \frac{1}{\#P_v} \sum_{(x,y,z)} y_i^2 - \left(\frac{1}{\#P_v} \sum_{(x,y,z)} y_i \right)^2 = E_{yy} - E_y^2$$

$$C_{zz} = \frac{1}{\#P_v} \sum_{(x,y,z)} y_i^2 - \left(\frac{1}{\#P_v} \sum_{(x,y,z)} y_i \right)^2 = E_{zz} - E_z^2$$

$$C_{xy} = \frac{1}{\#P_v} \sum_{(x,y,z)} x_i y_i - \left(\frac{1}{\#P_v} \sum_{(x,y,z)} x_i \right) \times \left(\frac{1}{\#P_v} \sum_{(x,y,z)} y_i \right) = E_{xy} - E_x E_y$$

$$C_{xz} = \frac{1}{\#P_v} \sum_{(x,y,z)} x_i z_i - \left(\frac{1}{\#P_v} \sum_{(x,y,z)} x_i \right) \times \left(\frac{1}{\#P_v} \sum_{(x,y,z)} z_i \right) = E_{xz} - E_x E_z$$

$$C_{yz} = \frac{1}{\#P_v} \sum_{(x,y,z)} y_i z_i - \left(\frac{1}{\#P_v} \sum_{(x,y,z)} y_i \right) \times \left(\frac{1}{\#P_v} \sum_{(x,y,z)} z_i \right) = E_{yz} - E_y E_z$$

The covariance matrix C of voxel co-ordinates is:

$$C = \begin{pmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{yz} \\ C_{zx} & C_{zy} & C_{zz} \end{pmatrix}$$

The diagonalization of the covariance matrix gives three eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$, which are the main, medium, and minor axes of an ellipsoid.

$\lambda \times e = C \times e$, where λ is the eigenvalue and e is the eigenvector.

Elliptic Approximation

The elliptic approximation uses the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ of the covariance matrix and computes an ellipsis with axis along the eigenvector e_1 with length a , and along the eigenvector e_2 with length b , and along the eigenvector e_3 with length c .

The formula is $a : b : c = \lambda_1 : \lambda_2 : \lambda_3$.

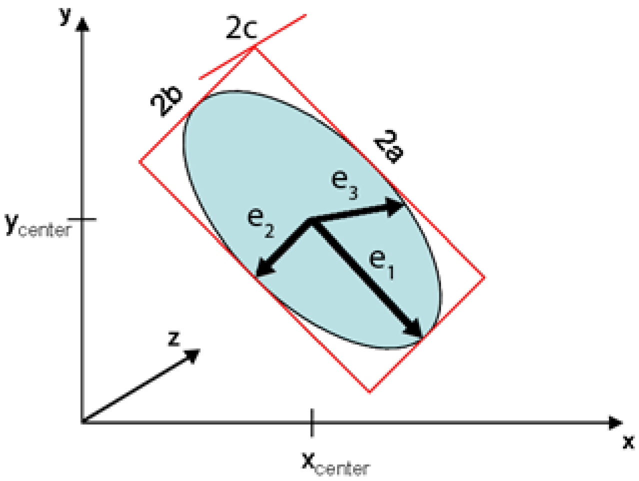


Figure 15. Elliptic approximation

About Algorithms

A single process executes an algorithm on an image object domain. It is the elementary unit of a rule set and provides a solution to a specific image analysis problem. Processes are the main working tools for developing rule sets. A rule set is a sequence of processes that are executed in a defined order.

The image object domain is a set of image objects. Every process loops through this set of image objects one by one and applies the algorithm to each single image object. This image object is referred to as the current image object.

Creating a Process

A single process can be created using the Edit Process dialog box, in which you can define:

- The method of the process from an algorithm list, for example Multiresolution Segmentation or Classification
- The image object domain on which an algorithm should be performed
- Detailed algorithm parameter settings.

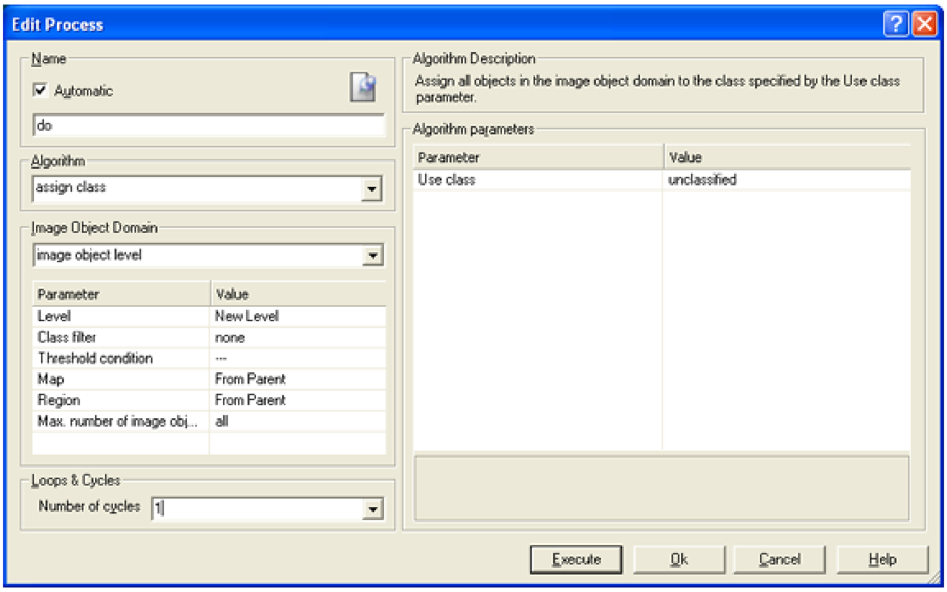


Figure 16. The Edit Process dialog box

Specifying Algorithm Parameters

Depending on the chosen algorithm, several parameters may be modified:

1. Define the individual settings of the algorithm in the Algorithms Parameters group box. If available, click a plus sign (+) button to expand the table to access additional parameters.

2. To edit values of parameters, select the parameter name or its value by clicking on it. Depending on the type of value, change the value in one of the following ways:
   - Edit values directly within the value field.
   - Click the ellipsis button located inside the value field. A dialog box opens, enabling you to configure the value.
   - Select the value from a drop-down list. For many parameters, you can select a feature by clicking the From feature item. or an existing variable. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK, or press Enter to open the Create Variable dialog box for additional settings.

1 Process-Related Operation Algorithms

The Process-Related Operation algorithms are used to control other processes.

1.1 Execute Child Processes

Execute all child (subordinate) processes of a process. To define a process using this algorithm as a parent process, you can right-click it and choose the Insert Child command in the context menu.

1.1.1 Supported Domains

Execute

Use the Execute Child Processes algorithm in conjunction with the Execute domain to structure your process tree. A process with this setting serves as a parent process, providing a container for a sequence of functionally related child processes.

Pixel Level

Applies the algorithm at the pixel level. Typically used for initial segmentations.

Image Object Level

Use the Execute Child Processes algorithm in conjunction with other image object domains (for example, the image object level domain) to loop over a set of image objects. All contained child processes will be applied to the image objects in the image object domain. In this case the child processes usually use one of the following as image object domain: current image object, neighbor object, super object, sub objects.

Current Image Object

Applies the algorithm to the current internally selected image object of the parent process.

Neighbor Image Object

Applies the algorithm to all neighbors of the current internally selected image object of the parent process. The size of the neighborhood is defined by the Distance parameter.

Super Object

Applies the algorithm to the superobject of the current internally selected image object of the parent process. The number of levels above in the image object level hierarchy is defined by the Level Distance parameter.

Sub Object

Applies the algorithm to the sub objects of the current internally selected image object of the parent process. The number of levels below in the image object level hierarchy is defined by the Level Distance parameter.

Maps

Applies the algorithm to all specified maps of a project. You can select this domain in parent processes with the Execute Child Process algorithm to set the context for child processes that use the map parameter From Parent.

Linked Objects

Applies the algorithm to all linked image objects of the current internally selected image object of the parent process.

Image Object List

Applies the algorithm to all image objects that were collected with the algorithm Update Image Object List.

Array

Options for the array function are visible when Array is selected as the Image Object Domain (for information on using arrays, please consult the user guide):

- **Array**: Select a predefined array
- **Array Type**: Displays the array type (this is not editable)
- **Variable**: The variable holding the value of the current array item when the process is executed
- **Index Variable**: The scene variable holding the index of the current array item when the process is executed

1.2 Execute Child As Series

When items are exported during a multimap analysis they are held in memory, which can often lead to out-of-memory problems during intensive analyses. The Execute Child as Series algorithm allows partial results to be exported as soon as they are available, freeing up memory for other tasks.

1.2.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Sub Object; Super Object; Linked Objects; Maps; Image Object List

1.2.2 Algorithm Parameters

Series Name

Enter the series name variables and strings are possible. All export algorithms receive a new parameter: Export Series (Yes/No)

1.3 If, Then and Else

If the conditions in the If algorithm are connected, all the defined conditions must be true to enter the Then path, otherwise the Else path is taken

- If no condition is defined, the Then path is chosen
- If there is more than one Then or Else condition, all are executed.

1.3.1 Supported Domains

Execute; Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Maps; Image Object List; Array

1.4 Throw

The Throw algorithm generates an error message (the Exception Message parameter) if a certain condition occurs, for example the number of objects exceeds a defined number.

1.4.1 Supported Domains

Execute; Pixel Level; Image Object Level: Current Image Object; Neighbor Image object; Super Object; Sub Objects; Linked Objects; Maps; Image Object List; Array

1.5 Catch

If an exception (generating an error message) occurs as defined in the Throw algorithm and a Catch algorithm has been defined, the error message is not generated. Instead the processes (and any child processes) defined in the Catch algorithm are executed. If there are Catch processes without an exception, the Catch process is not executed.

1.5.1 Supported Domains

Execute; Pixel Level; Image Object Level: Current Image Object; Neighbor Image object; Super Object; Sub Objects; Linked Objects; Maps; Image Object List; Array

1.6 Set Rule Set Options

The Set Rule Set Options algorithm lets you control certain settings for the rule set, or parts of rule sets. For example, you may want to apply particular settings to analyze large objects and change them to analyze small objects. In addition, because the settings are part of the rule set and not on the client, they are preserved when the rule set is run on a server.

You can save the rule set or the associated project to preserve the settings chosen in this algorithm, because saving the project also saves the rule set. The current settings display under Project Settings in the Options dialog box.

1.6.1 Supported Domains

Execute

1.6.2 Algorithm Parameters

Apply to Child Processes Only

If the value is No, settings apply globally, persisting after completion of execution. If Yes is selected, changes apply only to child processes.

Distance Calculation

- Smallest Enclosing Rectangle uses the smallest enclosing rectangle of an image object for distance calculations
- Center of Gravity uses the center of gravity of an image object for distance calculations
- Default reverts values to their defaults when the rule set is saved
- Keep Current maintains the existing settings when the rule set is saved

Current Resampling Method

- Center of Pixel initiates resampling from the center of the pixel.
- Upper Left Corner of Pixel initiates resampling from the upper-left corner of the pixel.
- Default reverts values to their defaults when the rule set is saved
- Keep Current maintains the existing settings when the rule set is saved

Evaluate Conditions on Undefined Features as 0

When a value is undefined, with respect to a condition you have specified, the software can evaluate it as false, or perform the evaluation based on a value of zero.

- **Yes**: Evaluate any condition with undefined feature as “false” (this is the default).
- **No**: Assign value “zero” to undefined feature before evaluation of any condition with undefined feature.
- **Default** reverts values to their defaults when the rule set is saved
- **Keep Current** maintains the existing settings when the rule set is saved.

Polygons Base Polygon Threshold

This value sets the degree of abstraction for the base polygons. The default is 1.25.

Polygons Shape Polygon Threshold

This value determines the degree of abstraction for the shape polygons. Shape polygons are independent of the topological structure and consist of at least three points. The threshold for shape polygons can be changed any time without the need to recalculate the base vectorization. The default value is one.

Polygons Remove Slivers

Remove Slivers is used to avoid intersection of edges of adjacent polygons and self-intersections of polygons.

Sliver removal becomes necessary with higher threshold values for base polygon generation. Note that the processing time to remove slivers is high, especially for low thresholds where it is not needed anyway.

- No enables the intersection of polygon edges and self-intersections
- Yes avoids the intersection of edges of adjacent polygons and self-intersections of polygons
- Default reverts values to their defaults when the rule set is saved
- Keep Current maintains the existing settings when the rule set is saved

Update Topology

Update Topology allows you to update neighborhood relations automatically or on demand.

Save Temporary Layers

Selecting Yes saves temporary layers to your hard drive. The default layer values in Definiens Developer XD 2.0.4 are:

- Yes for new rule sets.
- No for rule sets created with Definiens Enterprise Image Intelligence 7 or earlier.
- Yes for rule sets created with Definiens XD 1

Polygon Compatibility Mode

Set this option to use polygon compatibility option for polygons.

2 Segmentation Algorithms

Segmentation algorithms are used to subdivide entire images at a pixel level, or specific image objects from other domains into smaller image objects.

Definiens provides several different approaches to segmentation, ranging from very simple algorithms, such as chessboard and quadtree-based segmentation, to highly sophisticated methods such as multiresolution segmentation and contrast filter segmentation.

Segmentation algorithms are required whenever you want to create new image objects levels based on image layer information. But they are also a very valuable tool to refine existing image objects by subdividing them into smaller pieces for more detailed analysis.

2.1 Chessboard Segmentation

The Chessboard Segmentation algorithm splits the pixel domain or an image object domain into square image objects.

A square grid aligned to the image left and top borders of fixed size is applied to all objects in the domain and each object is cut along these gridlines.

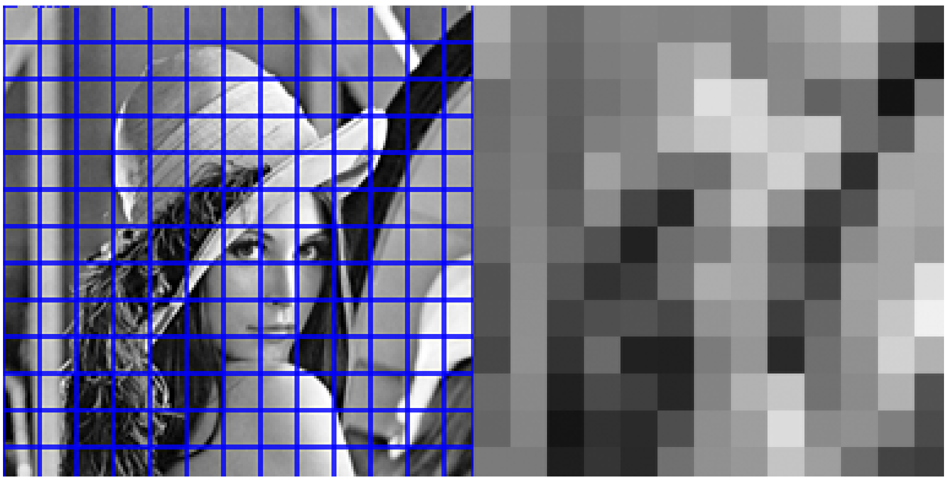


Figure 2.1. Result of chessboard segmentation with object size 20

2.1.1 Supported Domains

Pixel Level; Image Object Level: Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

2.1.2 Algorithm Parameters

Object Size

Object Size defines the size of the square grid in pixels. Variables are rounded to the nearest integer

Level Name

In the Level Name field, enter the name of a new image object level. This parameter is only available, if the domain Pixel Level is selected in the process dialog.

Overwrite Existing Level

This parameter is only available when Pixel Level is selected. It allows you to automatically delete an existing image level above the pixel level and replace it with a new level created by the segmentation.

Thematic Layer Usage

In the Thematic Layers field, specify the thematic layers to be considered in addition to segmentation. Each thematic layer used for segmentation will cause further splitting of image objects while enabling consistent access to its thematic information. You can segment an image using more than one thematic layer. The results are image objects representing proper intersections between the thematic layers. If you want to produce image objects based exclusively on thematic layer information, you can select a chessboard size larger than your image size.

2.2 Quadtree-Based Segmentation

The Quadtree-Based Segmentation algorithm splits the pixel domain or an image object domain into a quadtree grid formed by square objects.

A quadtree grid consists of squares with sides each having a power of two and aligned to the image left and top borders. It is applied to all objects in the domain and each object is cut along these gridlines. The quadtree structure is built so that each square has a maximum possible size and also fulfills the homogeneity criteria defined by the mode and scale parameters.

The maximum square object size is 256 x 256, or 65,536 pixels.

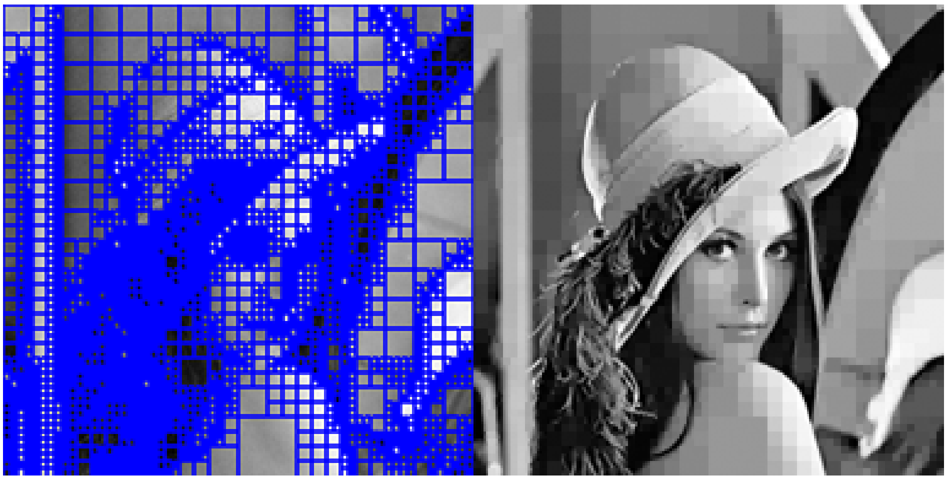


Figure 2.2. Result of quadtree-based segmentation with mode color and scale 40

2.2.1 Supported Domains

Pixel Level; Image Object Level: Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

2.2.2 Algorithm Parameters

Mode

- Color: The maximal color difference within each square image object is less than the Scale value
- Super Object Form: Each square image object must completely fit into the super-object. This mode only works with an additional upper image level

Scale

Scale defines the maximum color difference within each selected image layer inside square image objects. It is only used in conjunction with the Color mode.

Level Name

In the Level Name field, enter the name of a new image object level. This parameter is only available if the domain Pixel Level is selected in the process dialog.

Overwrite Existing Level

This parameter is only available when Pixel Level is selected. It allows you to automatically delete an existing image level above the pixel level and replace it with a new level created by the segmentation.

Image Layer Weights

Image layers can be weighted depending on their importance or suitability for the segmentation result. The higher the weight assigned to an image layer, the more weight will be assigned to that layer's pixel information during the segmentation process, if it is used. Consequently, image layers that do not contain the information intended for representation by the image objects should be given little or no weight. For example, when segmenting a geographical LANDSAT scene using multiresolution segmentation or spectral difference segmentation, the segmentation weight for the spatially coarser thermal layer should be set to 0 in order to avoid deterioration of the segmentation result by the blurred transient between image objects of this layer.

1. In the Algorithm parameters area, expand the Image Layer weights list and set the weight of the image layers to be considered by the algorithm. You can use both of the following methods:
    - Select an image layer and edit the weight value
    - Select Image Layer weights and click the ellipsis button located inside the value field to open the Image Layer Weights dialog box
2. Select an image layer in the list. To select multiple image layers press Ctrl.
3. Enter a new weight in the New value text box and click Apply.

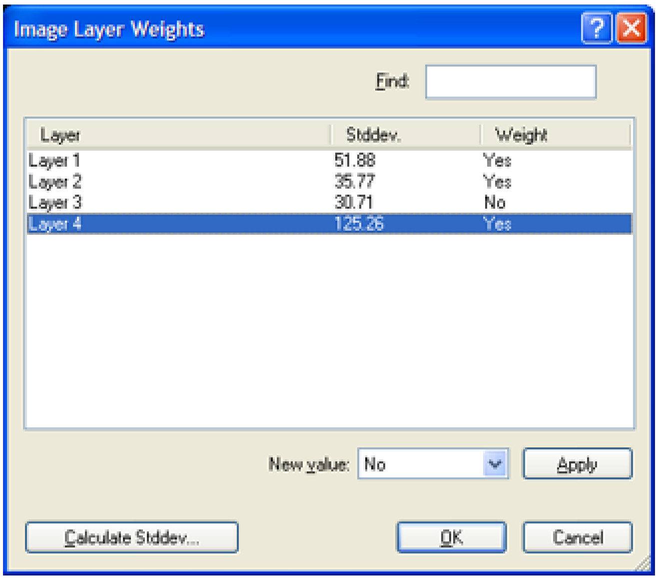


Figure 2.3. Image Layer Weights dialog box

Options

- Click the Calculate Stddev button to check the image layer dynamics. The calculated standard deviations of the image layer values for each single image layer are listed in the Stddev. column.
- To search for a specific layer, type the name into the Find text box.

2.2.3 Thematic Layer Weights

In the Thematic Layers field, specify the thematic layers to be considered in addition to segmentation. Each thematic layer used for segmentation will cause additional splitting of image objects while enabling consistent access to its thematic information. You can segment an image using more than one thematic layer. The results are image objects representing proper intersections between the thematic layers. If you want to produce image objects based exclusively on thematic layer information, you can select a chessboard size larger than your image size between the thematic layers.

2.3 Contrast Split Segmentation

The Contrast Split Segmentation algorithm segments an image or image object into dark and bright regions. It is based on a threshold that maximizes the contrast between the resulting bright objects (consisting of pixels with pixel values above the threshold) and dark objects (consisting of pixels with pixel values below the threshold).

The algorithm evaluates the optimal threshold separately for each image object in the image object domain. If the pixel level is selected in the image object domain, the algorithm first executes a chessboard segmentation, then performs the split on each square.

It achieves the optimization by considering different pixel values as potential thresholds. The test thresholds range from the minimum threshold to the maximum threshold, with intermediate values chosen according to the step size and stepping type parameter. If a test threshold satisfies the minimum dark area and minimum bright area criteria, the contrast between bright and dark objects is evaluated. The test threshold causing the largest contrast is chosen as the best threshold and used for splitting.

2.3.1 Supported Domains

Pixel Level; Image Object Level: Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

2.3.2 Settings

Chessboard Tile Size

This field is available only if pixel level is selected in the Image Object Domain. Enter the chessboard tile size (the default is 1,000).

Level Name

Select or enter the level to contain the results of the segmentation. Available only if the pixel level is in the image object domain.

Overwrite Existing Level

This parameter is only available when Pixel Level is selected. It allows you to automatically delete an existing image level above the pixel level and replace it with a new level created by the segmentation.

Minimum Threshold

Enter the minimum gray value to be considered for splitting. The algorithm calculates the threshold for gray values from the minimum threshold value to the maximum threshold value (the default is 0).

Maximum Threshold

Enter the maximum gray value to be considered for splitting. The algorithm calculates the threshold for gray values from the minimum threshold value to the maximum threshold value (the default is 255).

Step Size

Enter the step size by which the threshold increases from the minimum threshold to the maximum threshold. The value is either be added to the threshold or multiplied by the threshold, according to the selection in the Stepping Type field. The algorithm recalculates a new best threshold each time the threshold is changed by application of the values in the Step Size and Stepping Type fields, until the maximum threshold is reached. Higher values entered for step size tend to execute more quickly; smaller values tend to achieve a split with a larger contrast between bright and dark objects.

Stepping Type

Use the drop-down list to select one of the following:

- Add: Calculate each step by adding the value in the Scan Step field
- Multiply: Calculate each step by multiplying by the value in the Scan Step field

Image Layer

Select the image layer where the contrast is to be maximized.

Class for Bright Objects

Create a class for image objects brighter than the threshold or select one from the drop-down list. Image objects are not classified if the value in the Execute splitting field is No.

Class for Dark Objects

Create a class for image objects darker than the threshold or select one from the drop-down list. Image objects are not classified if the value in the Execute splitting field is No.

2.3.3 Advanced Settings

Contrast Mode

Select the method the algorithm uses to calculate contrast between bright and dark objects. The algorithm calculates possible borders for image objects (where a is the mean of bright border pixels and b is the mean of dark border pixels):

- Edge Ratio: $(a-b) \div (a+b)$
- Edge Difference: $(a-b)$
- Object Difference: The difference between the mean of all bright pixels and the mean of all dark pixels.

Execute Splitting

Select Yes to split objects with the best-detected threshold. Select No to simply compute the threshold without splitting.

Variable for Best Threshold

Enter a scene variable to store the computed pixel value threshold that maximizes the contrast.

Variable for Best Contrast

Enter a scene variable to store the computed contrast between bright and dark objects when splitting with the best threshold. The computed value is different for each contrast mode.

Minimum Relative Area Dark

Enter the minimum relative dark area. Segmentation into dark and bright objects only occurs if the relative dark area is higher than the value entered. Only thresholds that lead to a relative dark area larger than the value entered are considered as best thresholds. Setting this value to a number greater than 0 may increase speed of execution.

Minimum Relative Area Bright

Enter the minimum relative bright area. Only thresholds that lead to a relative bright area larger than the value entered are considered as best thresholds. Setting this value to a number greater than 0 may increase speed of execution.

Minimum Contrast

Enter the minimum contrast value threshold. Segmentation into dark and bright objects only occurs if a contrast higher than the value entered can be achieved.

Minimum Object Size

Enter the minimum object size in pixels that can result from segmentation. Only larger objects are segmented. Smaller objects are merged with neighbors randomly. (The default value of 1 effectively inactivates this option.)

2.4 Multiresolution Segmentation

The Multiresolution Segmentation algorithm locally minimizes the average heterogeneity of image objects for a given resolution of image objects. It can be executed on an existing image object level or the pixel level for creating new image objects on a new image object level.

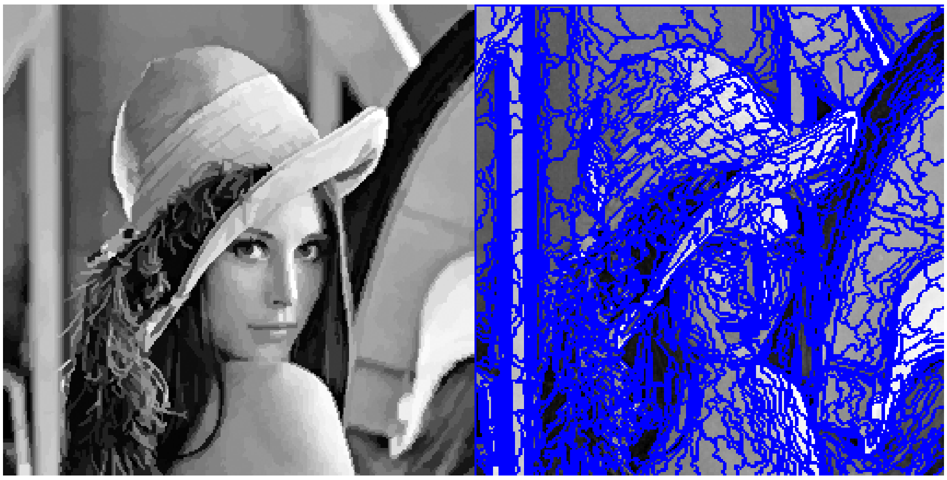


Figure 2.4. Result of multiresolution segmentation with scale 10, shape 0.1 and compactness 0.5

The multiresolution segmentation algorithm consecutively merges pixels or existing image objects. Thus it is a bottom-up segmentation algorithm based on a pairwise region merging technique. Multiresolution segmentation is an optimization procedure which, for a given number of image objects, minimizes the average heterogeneity and maximizes their respective homogeneity.

The segmentation procedure works according the following rules, representing a mutual-best-fitting approach:

1. The segmentation procedure starts with single image objects of one pixel and repeatedly merges them in several loops in pairs to larger units as long as an upper threshold of homogeneity is not exceeded locally. This homogeneity criterion is defined as a combination of spectral homogeneity and shape homogeneity. You

can influence this calculation by modifying the scale parameter. Higher values for the scale parameter result in larger image objects, smaller values in smaller image objects.

2. As the first step of the procedure, the seed looks for its best-fitting neighbor for a potential merger.
3. If best fitting is not mutual, the best candidate image object becomes the new seed image object and finds its best fitting partner.
4. When best fitting is mutual, image objects are merged.
5. In each loop, every image object in the image object level will be handled once.
6. The loops continue until no further merger is possible.

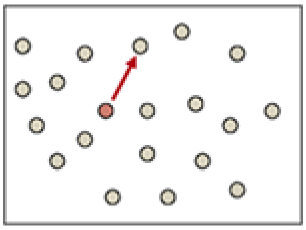


Figure 2.5. Each image object uses the homogeneity criterion to determine the best neighbor to merge with

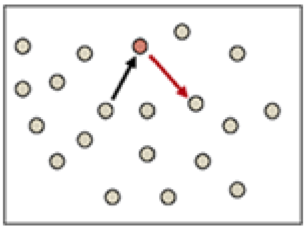


Figure 2.6. If the first image object's best neighbor (red) does not recognize the first image object (gray) as best neighbor, the algorithm moves on (red arrow) with the second image object finding the best neighbor

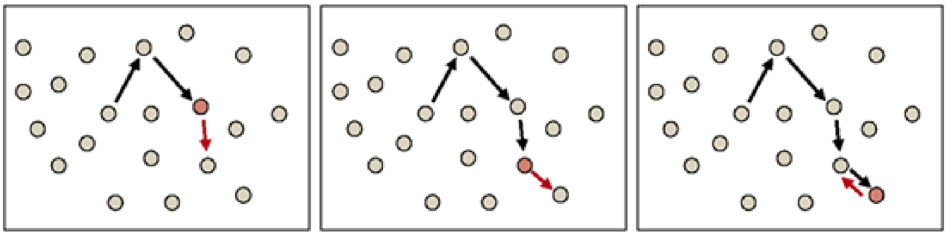


Figure 2.7. This branch-to-branch hopping repeats until mutual best fitting partners are found

The procedure continues with another image object's best neighbor. The procedure iterates until no further image object mergers can be realized without violating the maximum allowed homogeneity of an image object.

With any given average size of image objects, multiresolution segmentation yields good abstraction and shaping in any application area. However, it has higher memory requirements and significantly slower performance than some other segmentation techniques and therefore is not always the best choice.

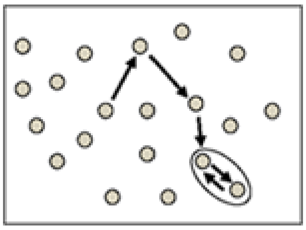


Figure 2.8. If the homogeneity of the new image object does not exceed the scale parameter, the two partner image objects are merged.

2.4.1 Supported Domains

Pixel level; Image Object Level

2.4.2 Level Settings

Level Name

The Level Name field lets you define the name for the new image object level. It is only available if a new image object level will be created by the algorithm. To create new image object levels, use either the image object domain pixel level in the process dialog or set the Level Usage parameter to Create Above or Create Below.

Overwrite Existing Level

This parameter is only available when Pixel Level is selected. It allows you to automatically delete an existing image level above the pixel level and replace it with a new level created by the segmentation.

Level Usage

Select one of the available modes from the drop-down list. The algorithm is applied according to the mode based on the image object level that is specified by the image object domain. This parameter is not visible if pixel level is selected as image object domain in the Edit Process dialog box.

- Use Current applies multiresolution segmentation to the existing image object level. Objects can be merged and split depending on the algorithm settings
- Use Current (Merge Only) applies multiresolution segmentation to the existing image object level. Objects can only be merged. Usually this mode is used together with stepwise increases of the scale parameter
- Create Above creates a copy of the image object level as super objects
- Create Below creates a copy of the image object level as sub objects.

2.4.3 Segmentation Settings

Image Layer Weights

Enter weighting values – the higher the weight assigned to an image layer, the more weight will be given to that layer's pixel information during the segmentation process.

You can also use a variable as a layer weight.

Thematic Layer Usage

Specify the thematic layers to be candidates for segmentation. Each thematic layer that is used for segmentation will lead to additional splitting of image objects while enabling consistent access to its thematic information. You can segment an image using more than one thematic layer. The results are image objects representing proper intersections between the thematic layers.

Scale Parameter

The Scale Parameter is an abstract term that determines the maximum allowed heterogeneity for the resulting image objects. For heterogeneous data, the resulting objects for a given scale parameter will be smaller than in more homogeneous data. By modifying the value in the Scale Parameter value you can vary the size of image objects.

> **TIP**: Always produce image objects of the biggest possible scale that still distinguish different image regions (as large as possible and as fine as necessary). There is a tolerance concerning the scale of the image objects representing an area of a consistent classification due to the equalization achieved by the classification. The separation of different regions is more important than the scale of image objects.

2.4.4 Composition of Homogeneity Criterion

The object homogeneity to which the scale parameter refers is defined in the Composition of Homogeneity criterion field. In this circumstance, homogeneity is used as a synonym for minimized heterogeneity. Internally, three criteria are computed: color, smoothness, and compactness. These three criteria for heterogeneity may be applied in many ways although, in most cases, the color criterion is the most important for creating meaningful objects. However, a certain degree of shape homogeneity often improves the quality of object extraction because the compactness of spatial objects is associated with the concept of image shape. Therefore, the shape criteria are especially helpful in avoiding highly fractured image object results in strongly textured data (for example radar data).

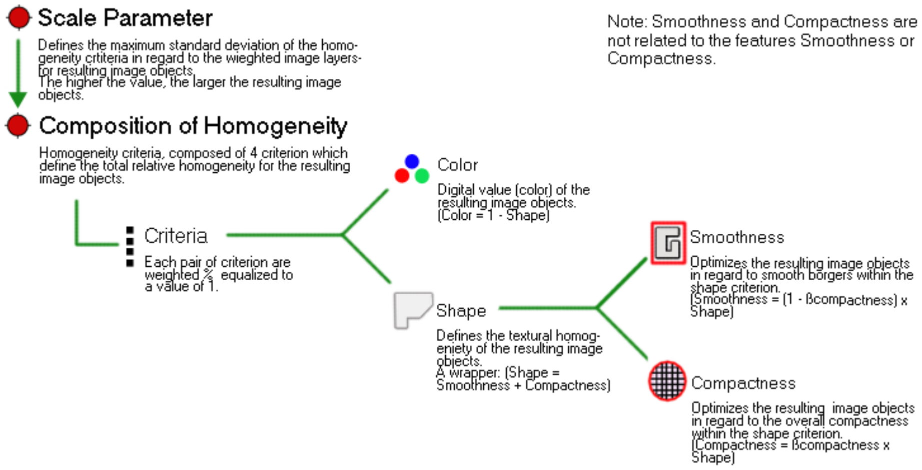


Figure 2.9. Multiresolution concept flow diagram

Shape

The value of the Shape field modifies the relationship between shape and color criteria; By modifying the Shape criterion,[1] you define the color criteria (color = 1 shape). In effect, by decreasing the value assigned to the Shape field, you define to which percentage the spectral values of the image layers will contribute to the entire homogeneity criterion. This is weighted against the percentage of the shape homogeneity, which is defined in the Shape field.

Changing the weight for the Shape criterion to 1 will result in objects more optimized for spatial homogeneity. However, the shape criterion cannot have a value larger than 0.9, due to the fact that without the spectral information of the image, the resulting objects would not be related to the spectral information at all. The slider bar adjusts the amount of Color and Shape to be used for the segmentation.

In addition to spectral information, the object homogeneity is optimized with regard to the object shape, defined by the Compactness parameter.

Compactness

The compactness criterion is used to optimize image objects with regard to compactness. This criterion should be used when different image objects which are rather compact, but are separated from non-compact objects only by a relatively weak spectral contrast. Use the slider bar to adjust the degree of compactness to be used for the segmentation.

1. A high value for the shape criterion operates at the cost of spectral homogeneity. However, spectral information is in the end the primary information contained in image data. Using the shape criterion too intensively may thus reduce the quality of segmentation results.

2.5 Spectral Difference Segmentation

The Spectral Difference Segmentation algorithm merges neighboring image objects according to their mean image layer intensity values. Neighboring image objects are merged if the difference between their layer mean intensities is below the value given by the maximum spectral difference.

This algorithm is designed to refine existing segmentation results, by merging spectrally similar image objects produced by previous segmentations. It cannot be used to create new image object levels based on the pixel level domain.

2.5.1 Supported Domains

Image Object Level

2.5.2 Level Settings

Level Usage

Define the image object level to be used: the current one or a new one to be created above the current.

Level Name

Define the Level Name of the image object level to be created. Not available if you use the current image object level.

2.5.3 Segmentation Settings

Maximum Spectral Difference

Define the maximum spectral difference in gray values between image objects that are used during the segmentation. If the difference is below this value, neighboring objects are merged.

Parameters

- $wn_{[k1,k2,k3]}$ are the normalized layer weights $w_{[k1,k2,k3]}$
  - $wn_{k1} = \dfrac{wk_1}{(w_{k1} + w_{k2} + w_{k3})}$
  - $wn_{k2} = \dfrac{wk_2}{(w_{k1} + w_{k2} + w_{k3})}$
  - $wn_{k3} = \dfrac{wk_3}{(w_{k1} + w_{k2} + w_{k3})}$

Expression

$$wn_{k1} \times abs(k1_1 - k1_2) + wn_{k2} \times abs(k2_1 - k2_2) + wn_{k3} \times abs(k3_1 - k3_2)$$

Image Layer Weights

Enter weighting values – the higher the weight assigned to an image layer, the more weight will be given to that layer's pixel information during the segmentation process.

You can also use a variable as a layer weight.

Thematic Layer Usage

Specify the thematic layers that are to be considered in addition for segmentation. Each thematic layer used for segmentation will lead to additional splitting of image objects, while enabling consistent access to its thematic information. You can segment an image using more than one thematic layer. The results are image objects representing proper intersections between the thematic layers.

2.6 Multi-Threshold Segmentation

Multi-Threshold Segmentation splits the image object domain based on pixel values. This creates image objects and classifies them based on user-created thresholds. It can also be used to create unclassified image objects based on pixel values thresholds.

2.6.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

2.6.2 Level Settings

Level Name

Available only if the image object domain is pixel level. Select an existing level or enter a name.

Overwrite Existing Level

This parameter is only available when Pixel Level is selected. It allows you to automatically delete an existing image level above the pixel level and replace it with a new level created by the segmentation.

Image Layer

Select an image layer to use for threshold evaluation.

Ensure Connected Objects

Select the type of image objects that are created (the image object domain must be pixel level):

- If Yes, all image objects are marked as connected
- If No, image objects can be marked as disconnected

Merge Image Objects First

This parameter specifies the behaviour if the Ensure Connected Objects parameter is set to Yes. It is only available if you select an image object domain that represents existing image objects.

- If Yes, all image objects with the same classification are first merged into one disconnected object and then separated again into connected image objects. By using this option you might lose some borders between image objects that existed before executing the algorithm. (This was the standard behaviour for Definiens XD 1.0 software.)
- If No, only disconnected image objects are further separated into connected image objects where necessary.

Min Object Size

Limits the minimum size of resulting image objects to a defined value.

Thresholds

You can set multiple thresholds to classify image objects based on pixel values. An additional threshold field is added as each threshold is created. Multiple thresholds must be in ascending order:

Class 1 Pixels below the threshold will be classified as the class selected or entered.

Threshold 1 Enter a pixel value below which pixels will be classified as defined in the Class field. Alternatively, you can select a feature or a variable. To create a new variable, type a name for the new variable and click OK to open the Create Variable dialog box.

2.7 Contrast Filter Segmentation

Pixel filters detect potential objects by contrast and gradient, and create suitable object primitives. An integrated reshaping operation modifies the shape of image objects to help form coherent and compact image objects. The resulting pixel classification is stored in an internal thematic layer. Each pixel is classified as one of the following classes: No Object, Object in First Layer, Object in Second Layer, Object in Both Layers and Ignored by Threshold. Finally a chessboard segmentation is used to convert this thematic layer into an image object level.

Contrast filter segmentation, as a first step, improves overall image analysis performance substantially.

2.7.1 Supported Domains

Pixel Level

2.7.2 Chessboard Settings

Configure the final chessboard segmentation of the internal thematic layer.

Object Size

Object Size

Level Name

Level name of the level to be created

Overwrite Existing Level

This parameter is only available when Pixel Level is selected. It allows you to automatically delete an existing image level above the pixel level and replace it with a new level created by the segmentation.

Thematic Layer

Thematic layer usage flags

2.7.3 Input Parameters

These parameters are identical for the first and second layers.

Layer

Choose the image layers to analyze. You can disable the second image layer by selecting No Layer. If you select No Layer, then the other parameters in the second layer group are inactive.

Scale 1–4

Several scales can be defined and analyzed at the same time. If at least one scale is tested positive, the pixels are classified as image objects. The default value is "no scale", indicated by a scale value of 0. To define a scale, edit the scale value.

The scale value n defines a square (gray) with a side length of $2d'(d' = \{\text{all pixels with distance to the current pixel} \leq |n| \times 2+1 \text{ but} \rightarrow (|n|-2) \times 2+1)\}$ with the current pixel in its center. The mean value of the pixels inside the outer square is compared with the mean value of the pixels inside a inner square (red) with a side length of $2d'(d' = \{\text{all pixels with distance to the current pixel} \leq |n| - 2 \times 2 + 1 \text{ but not the pixel itself}\})$. In the case of $|n| \leq 3$ it is just the pixel value.

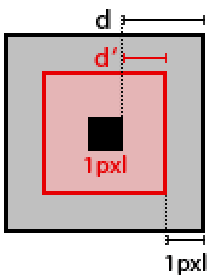


Figure 2.10. Scale testing of the contrast filter segmentation

Select a positive scale value to find objects that are brighter than their surroundings on the given scale; select a negative scale value to find darker objects.

Gradient

Use additional minimum gradient criteria for objects. Using gradients can increase the computing time for the algorithm. Set this parameter to 0 to disable the gradient criterion.

Lower Threshold

Pixels with layer intensity below this threshold are assigned to the Ignored by Threshold class.

Upper Threshold

Pixels with layer intensity above this threshold are assigned to the Ignored by Threshold class.

2.7.4 ShapeCriteria Settings

If you expect coherent and compact image objects, shape criteria parameters provide an integrated reshaping operation that modifies the shape of image objects by cutting off protruding parts and filling indentations and hollows.

ShapeCriteria Value

Protruding parts of image objects are declassified if a direct line crossing the hollow is smaller or equal than the ShapeCriteria value. Indentations and hollows of image objects are classified as the image object if a direct line crossing the hollow is smaller or equal than the shape criteria value. If you do not want any reshaping, set the value to 0.

Working on Class

Select a class of image objects for reshaping. The pixel classification can be transferred to the image object level using the classification parameters.

2.7.5 Classification Parameters

Enable Class Assignment

Select Yes or No in order to use or disable the classification parameters. If you select No, then the other parameters in the group are inactive.

No Objects

Pixels failing to meet the defined filter criteria are assigned the selected class.

Ignored By Threshold

Pixels with a layer intensity below or above the threshold values are assigned the selected class.

Object in First Layer

Pixels that match the filter criteria in first layer, but not the second layer, are assigned the selected class.

Objects in Both Layers

Pixels that match the filter criteria value in both layers are assigned the selected class.

Objects in Second Layer

Pixels that match the scale value in second layer, but not the first layer, are assigned the selected class.

2.8 Watershed Segmentation

The Watershed Segmentation algorithm implements a watershed transformation.

The algorithm takes in an image layer, that it regards as topographic map. It then simulates a rain drop falling down onto that map and pouring down into a local minima. All pixels, whose raindrops end up in the same local minima, form an image object (catchment basin). Image object borders relate to the watershed lines.

To overcome over-segmentation, the user can apply the ‘seed-based fusion’ principle, where each object can be a ‘seed’ or a ‘candidate’. Objects are merged until there are only seeds remaining. Seeds will only merge with candidate objects and candidate objects become seeds when defined criteria are met.

2.8.1 Examples

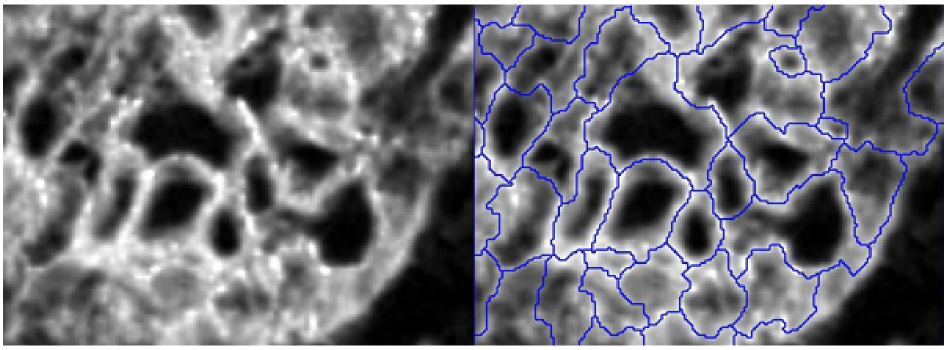


Figure 2.11. Original image and watershed segmentation

Figure 2.12. Original image – Edge image(Sobel) – and watershed segmentation

2.8.2 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

2.8.3 Algorithm Parameters

Layer

Select the layer for the algorithm to operate upon.

Invert Level

Specify whether to invert the layer before applying the watershed algorithm.

Connectivity

Select the pixel connectivity; choose from 4-connected or 8-connected.

Seed-Based Fusion

Seed Criterion Select a criterion for seed-based fusion. For no fusion, select <disabled>. The options are:

- **Ovfl. Area (pxl)**: Overflow Area – the number of pixels below the minimum border pixel
- **Ovfl. Height**: Overflow Height – minimum border pixel
- **Ovfl. Depth**: Overflow Depth – the difference between the minimum border and minimum object pixel
- **Ovfl. Volume**: Overflow Volume – the sum of the difference between the minimum border and the pixel, for all pixels below the minimum border pixel.

Threshold Choose a threshold for the seed criterion.

Seed Criterion Select an additional seed criterion if you wish.

Fuse Super Objects Select whether to fuse super objects.

3 Basic Classification Algorithms

Classification algorithms analyze image objects according to defined criteria and assign them to a class that best meets them.

3.1 Assign Class

Assign all objects of the image object domain to the class specified by the Use Class parameter. The membership value for the assigned class is set to 1 for all objects independent of the class description. The second and third-best classification results are set to 0.

3.1.1 Use Class

Select the class for the assignment from the drop-down list. You can also create a new class for the assignment.

3.2 Classification

Evaluates the membership value of an image object against a list of selected classes. The classification result of the image object is updated according to the class evaluation result. The three best classes are stored in the image object classification result. Classes without a class description are assumed to have a membership value of 1.

3.2.1 Active Classes

Erase Old Classification, If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

3.3 Hierarchical Classification

The Hierarchical Classification algorithm evaluates the membership value of an image object against a list of selected classes.

The classification result of the image object is updated according to the class evaluation result. The three best classes are stored as the image object classification result. Classes without a class description are assumed to have a membership value of 0. Class-related features are considered only if explicitly enabled by the according parameter.

This algorithm is optimized for applying complex class hierarchies to entire image object levels. This reflects the classification algorithm of eCognition Professional 4. When working with domain-specific classification in processes, the algorithms Assign Class and Classification are recommended.

3.3.1 Active Classes

Choose the list of active classes for the classification.

3.3.2 Use Class-Related Features

Enable to evaluate all class-related features in the class descriptions of the selected classes. If it is disabled these features will be ignored.

3.4 Remove Classification

Delete specific classification results from image objects.

3.4.1 Classes

Select classes that should be deleted from image objects.

3.4.2 Process

Enable to delete computed classification results created via processes and other classification procedures from the image object.

3.4.3 Manual

Enable to delete manual classification results from the image object.

4 Advanced Classification Algorithms

Advanced classification algorithms classify image objects that fulfill special criteria, such as being enclosed by another image object, or being the smallest or largest object in a set.

4.1 Find Domain Extrema

Find Domain Extrema classifies image objects with the smallest or largest feature values within the image object domain, according to an image object feature.

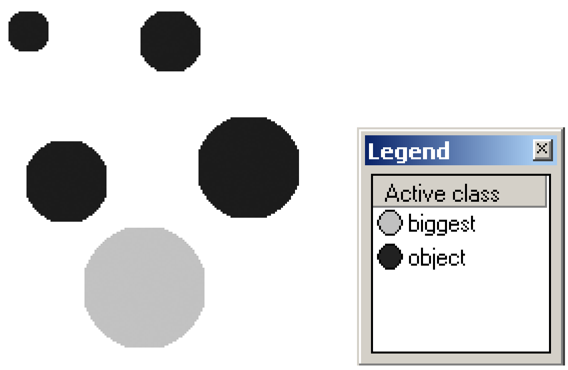


Figure 4.1. Result of Find Domain Extrema, with Extrema Type set to Maximum and Feature Set to Area

4.1.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

4.1.2 Extrema Settings

Extrema Type

Choose Minimum to classify image objects with the smallest feature values and Maximum for image objects with the largest feature values.

Feature

Choose the feature to use for finding the extreme values.

Accept Equal Extrema

This feature enables the algorithm to accept equal extrema. This parameter defines the behavior of the algorithm if more than one image object is fulfilling the extreme condition. If enabled, all image objects will be classified; if not, none of the image objects will be classified.

4.02 Compatibility Mode

Select Yes from the Value field to enable compatibility with older software versions (version 4.02 and older). This parameter will be removed from future versions.

4.1.3 Classification Settings

Specify the classification that will be applied to all image objects fulfilling the extreme condition. At least one class needs to be selected in the active class list for this algorithm.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification if There is no New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

4.2 Find Local Extrema

Find Local Extrema classifies image objects that fulfill a local extrema condition, according to image object features within a search domain in their neighborhoods.

Image objects with either the smallest or the largest feature value within a specific neighborhood will be classified according to the classification settings.

4.2.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

4.2.2 Search Settings

The search settings let you specify a search domain for the neighborhood around the image object.

Class Filter

Choose the classes to be searched. Image objects will be part of the search domain if they are classified with one of the classes selected in the class filter. Always add the class selected for the classification to the search class filter. Otherwise cascades of incorrect extrema resulting from the reclassification during the execution of the algorithm may appear.

Search Range

Define the search range in pixels. All image objects with a distance below the given search range will be part of the search domain. Use the drop-down arrows to select zero or positive numbers.

Connected

Enable to ensure that all image objects in the search domain are connected with the analyzed image object via other objects in the search range.

3.5 Compatibility Mode

Select Yes from the Value field to enable compatibility with older software versions (version 3.5 and older). This parameter will be removed in future versions.

4.2.3 Conditions

Define the extrema conditions.

Extrema Type

Choose Minimum for classifying image objects with the smallest feature values and Maximum for classifying image objects with largest feature values.

Feature

Choose the feature to use for finding the extreme values

Equal Extrema Condition

This parameter defines the behavior of the algorithm if more than one image object is fulfilling the extrema condition:

- Do Not Accept Equal Extrema means no image objects will be classified
- Accept Equal Extrema means all image objects will be classified
- Accept First Equal Extrema will result in the classification of the first of the image objects.

4.2.4 Classification Settings

Specify the classification that will be applied to all image objects fulfilling the extreme condition. At least one class needs to be selected in the active class list for this algorithm.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification if There is no New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

4.3 Find Enclosed by Class

Find and classify image objects that are completely enclosed by image objects belonging to certain classes.

If an image object is located at the border of the image, it will not be found and classified by Find Enclosed by Class. The shared part of the outline with the image border will not be recognized as the enclosing border.

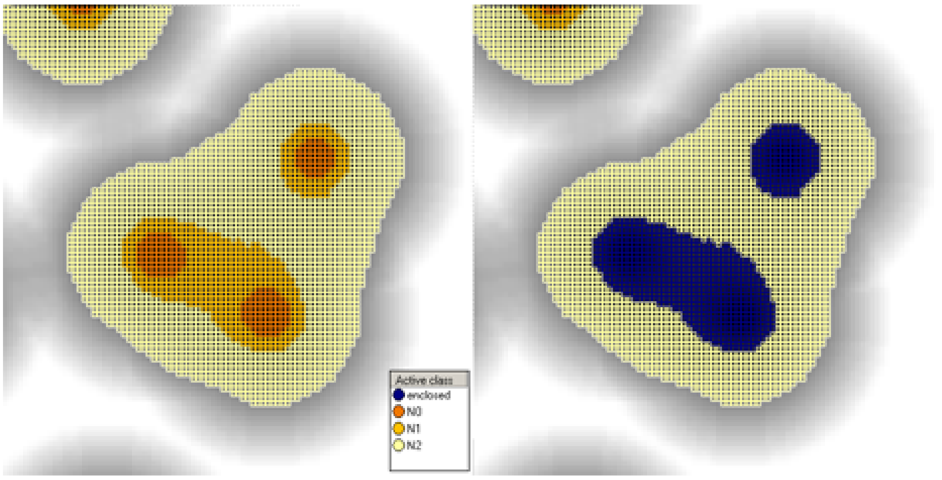


Figure 4.2. Left: Input of Find Enclosed by Class: Image Object Domain: Image Object Level, Class Filter: N0, N1. Enclosing class: N2. Right: Result of Find Enclosed by Class: Enclosed objects get classified with the class 'enclosed'. Note that the objects at the upper image border are not classified as enclosed

4.3.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

4.3.2 Search Settings

Enclosing Classes

Choose the classes that might be enclosing the image objects

Compatibility Mode

Select Yes from the Value field to enable compatibility with older software versions (version 3.5 and 4.0). This parameter will be removed with future versions.

4.3.3 Classification Settings

Choose the classes that should be used to classify enclosed image objects.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

4.4 Find Enclosed by Image Object

Find and classify image objects that are completely enclosed by image objects from the image object domain. Enclosed image objects located at the image border will be found and classified by the Find Enclosed by Image Object algorithm. The shared part of the outline with the image border will be recognized as the enclosing border.

In figure 4.3 on the facing page, the inputs for the left-hand image in Find Enclosed by Image Object are image object domain: image object level, class filter: N2. On the right-hand image, using the same algorithm, enclosed objects are classified with the class enclosed. Note that the objects at the upper image border are classified as enclosed.

4.4.1 Classification Settings

Choose the class that will be used to classify enclosed image objects.

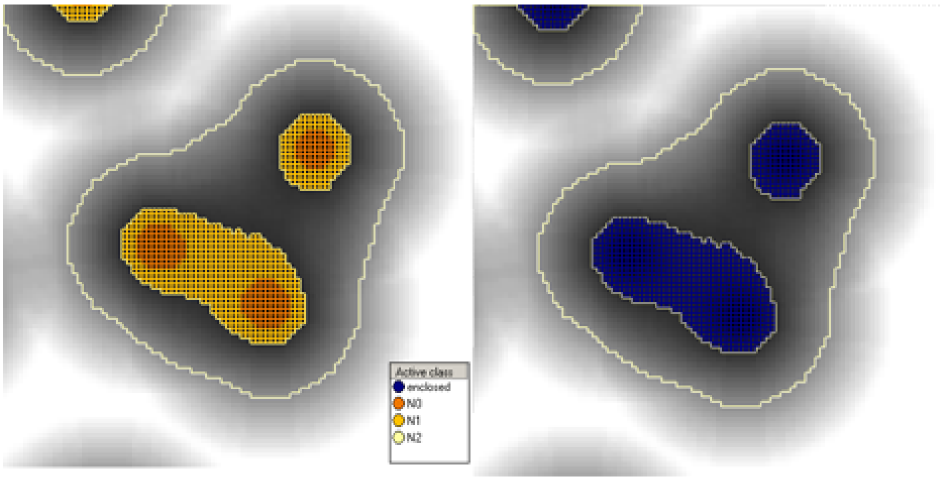


Figure 4.3. Comparison of Find Enclosed by Class

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

4.5 Connector

Classify the image objects that make up the shortest connection between the current image object and another image object that meets the conditions described by the connection settings.

The process starts to search from the current image object along objects that meet the conditions as specified by Connect Via and Super Object Mode Via, until it reaches an image object that meets the conditions specified by Connect To and Super Object Mode To. All image objects that are part of the resulting connection are assigned to the selected class.

You can define the maximum search range under Search Range In Pixels.

4.5.1 Connector Settings

Connector Via

Choose the classes whose image objects you want to be searched when determining the connection. Image objects assigned to other classes will not be taken into account.

Super Object Mode Via

Limit the number of image objects that are taken into account for the connection by specifying a superordinate object. Choose one of the following options:

- Don't Care: Use any image object
- Different Super Object: Use only images with a different superobject than the seed object
- Same Super Object: Use only image objects with the same superobject as the seed object

Connect To

Choose the classes whose image objects you want to be searched when determining a destination for the connection. Image objects assigned to other classes will not be taken into account.

Super Object Mode To

Limit the number of image objects that are taken into account for the destination of the connection by specifying a superordinate object. Choose one of the following options:

- Don't Care use any image object
- Different Super Object use only images with a different superobject than the seed object
- Same Super Object use only image objects with the same superobject as the seed object.

Search Range

Enter the Search Range in pixels that you wish to search.

4.5.2 Classification Settings

Choose the class that should be used to classify the connecting objects.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

4.6 Assign Class By Slice Overlap (Prototype)

4.6.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

4.6.2 Algorithm Parameters

Use Class

Select class for assignment

Candidate Domain

Class filter for objects that are considered.

Create 3D Objects

Create 3D objects

Slices Up

Describes how many slices the object is copied up. Use −1 for no limit.

Slices Down

Describes how many slices the object is copied down. Use −1 for no limit.

4.7 Optimal Box (Prototype)

Generate member functions for classes by looking for the best separating features based upon sample training.

4.7.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

4.7.2 Sample Class Parameters

For Target Samples

Select a class or create a new class that provides samples for the target class (the class to be ‘trained’).

For Rest Samples

Select a class or create a new class that provides samples for the rest of the domain.

4.7.3 Insert Membership Function Parameters

For Target Samples Into

Select a class or create a new class that receives membership functions after optimization for target. If set to unclassified, the target sample class is used.

For Rest Samples Into

Select a class or create a new class that receives inverted similarity membership functions after optimization for target. If set to unclassified, the rest sample class is used.

Clear All Membership Functions

When inserting new membership functions into the active class, choose whether to clear all existing membership functions or clear only those from input feature space:

- No, Only Clear if Associated with Input Feature Space: Clear membership functions only from the input feature space when inserting new membership functions into the active class.
- Yes, always clear all membership functions: Clear all membership functions when inserting new membership functions into the active class.

4.7.4 Feature Optimization Parameters

Input Feature Set

Input set of descriptors from which a subset will be chosen. Click the ellipsis button to open the Select Multiple Features dialog box. The Ensure Selected Features are in Standard Nearest Neighbor Feature Space checkbox is selected by default.

Minimum Number of Features

The minimum number of features descriptors to employ in a class. The default is 1.

Maximum Number of Features

The minimum number of features descriptors to employ in a class.

4.7.5 Optimization Settings Parameters

Weighted Distance Exponent

Enter a number greater than 0 to decrease weighting with increasing distance. Enter 0 to weight all distances equally. The default is 2.

Simplification Factor Mode

The simplification factor is calculated as:

$$e^{\frac{\text{cur dimension}}{\text{total dim}}}$$

The following values are available:

- Do Not Apply: Simplification factor is not applied
- Apply to Weighted Distance: Weighted distance is multiplied by simplification factor trades off simplicity against distance separation for equally good overlap
- Apply to Overlap: Overlap is divided by simplification factor trades off simplicity against false positives.

Simplification Factor Exponent

The exponent where the simplification factor is calculated as:

$$e^{\frac{\text{cur dimension}}{\text{total dim}}}$$

4.7.6 Optimization Output Parameters

False Positives Variable

Set this variable to the number of false positives after execution.

False Negatives Variable

Set this variable to the number of false negatives after execution.

False Positives Object List

Select an existing image object list variable, to which image objects associated with false positive samples are added. If a value is not set, no output occurs. Please note that samples stored in the workspace have no associated objects, so cannot be included in the list.

False Negatives Object List

Select an existing image object list variable, to which image objects associated with false negative samples are added. If a value is not set, no output occurs. Please note that samples stored in the workspace have no associated objects, so cannot be included in the list.

Separated Positives Object List (For Selected Feature Space)

This list contains image objects associated with positive samples, which are separated from other positive samples by false positives, using the generated feature space.

Separated Positives Object List (From ‘For Target Samples Into’ Class)

This list contains image objects associated with positive samples, which are separated from other positive samples by false positives, using a feature space from ‘For Target Samples Into’.

Show Info in Message Console

Show information on feature evaluations in message console.

4.8 Classifier

The Classifier algorithm lets you apply machine-learning functions to your analysis. Using it is a two-step process. Firstly, a classifier is trained using the classified objects of the image object domain as training samples. The trained classifier is stored as a string in the Configuration variable. In the second step, the trained classifier from the first step is applied to the domain, classifying the image objects according to what it learned in the training step.

The Classifier algorithm can be pixel- or object-based.

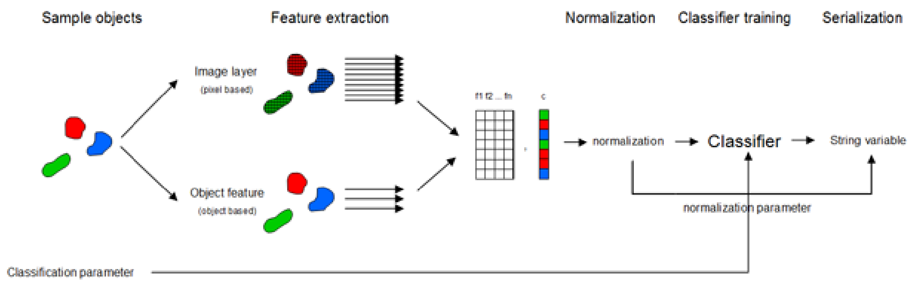


Figure 4.4. The training step

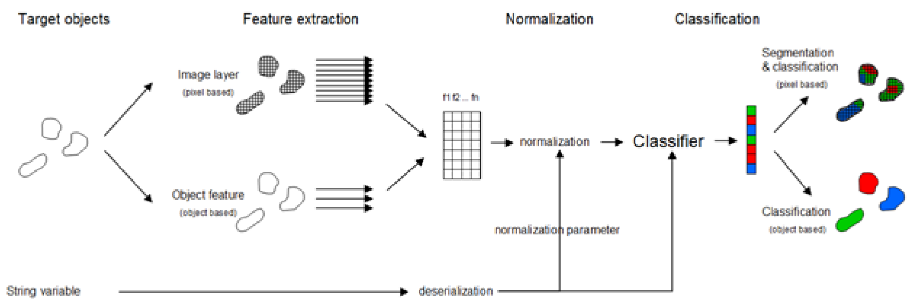


Figure 4.5. The application step

4.8.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

4.8.2 General Parameters

Operation

Choose the operation to perform:

- **Train**: Train a classifier. The domain objects serve as training samples.
- **Apply**: Apply a trained classifier to the domain objects.
- **Query**: Query attributes of a trained classifier.

4.8.3 Train: General Parameters

Configuration

Enter or select the variable to store the configuration to

Use Samples Only

Choose Yes or No depending on whether you want to use sample image objects only

4.8.4 Train: Feature Parameters

Type

Choose from:

- **Object-Based**: Use image objects
- **Pixel-Based**: Use the pixels contained in image objects

Feature

Select features used for object based classification.

Layer array

Select layer array used for pixel based classification.

4.8.5 Train: Classifier Parameters

Select a classifier to train from Decision Tree, Bayes, KNN or SVM. For more information about the supported classifiers, see the *OpenCV C++ Reference.*[1]

4.8.6 Apply: General Parameters

Configuration

Select the variable to load the configuration from.

1. opencv.willowgarage.com/documentation/cpp/index.html

4.8.7 Apply: Feature Parameters

Choose the same settings as in the training step.

Type

Choose from:

- **Object-Based**: Use image objects
- **Pixel-Based**: Use the pixels contained in image objects

Layer Array

Select a layer array to use as pixel-based features.

4.8.8 Query: General Parameters

Configuration

Select the variable to load the configuration from.

4.8.9 Query: Query Information about a trained classifier

Type

Select from:

- **Bayes**
- **KNN**
- **SVM**
- **Decision Tree**. Selecting Decision Tree displays the following options:
  - **Operation**. Select from:
    - **Query Parameter**:
      - **Depth**:
      - **Min Sample Count**:
      - **Depth**:
    - **Query Node**:
      - **Id**: Specify the tree node to query properties from
      - **Id leaf**: Specify the variable to receive a property
      - **Assigned Class**: Specify the class variable to receive
      - **Left Child Node**: Specify the variable to receive a property
      - **Right Child Node**: Specify the variable to receive a property
      - **Split Feature Index**: Specify the variable to receive a property
      - **Split Threshold**: Specify the variable to receive a property
    - **Plot**: Export a diagram of the decision tree as a Windows bitmap
      - **Image Path**: Enter the image path to export the image file

    * **Convert to CNL**: Generate CNL code, that is equivalent to the decision tree
        - **Target Process Path**: Location of the CNL tree
        - **Prefix**: Add a prefix.

For more information about classifier functions, see the *OpenCV C++ Reference*.[2]

2. opencv.willowgarage.com/documentation/cpp/index.html

5 Variables Operation Algorithms

Variable operation algorithms are used to modify the values of variables. They provide different methods for performing computations based on existing variables and image object features, and for storing the results within variables.

5.1 Timer

The Timer algorithm adds to a variable the elapsed time of its sub-processes (with the exception of customized algorithms). It is useful for debugging and improving performance in rule sets.

5.1.1 Supported Domains

Execute

5.1.2 Algorithm Parameters

Timer Variable

Select the variable to which the process execution time and all its subprocesses will be added.

5.2 Update Variable

Perform an arithmetic operation on a variable.

5.2.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List; Array

5.2.2 Algorithm Parameters

Variable Type

Object, scene, feature, class, level, image layer, thematic layer, or map variables must be selected as the vriable type. Select the variable assignment, according to the variable type selected in the Variable Type field. To select a variable assignment, click in the field and do one of the following depending on the variable type:

- For object variables, use the drop-down arrow to open the Select Single Feature dialog box and select a feature or create a new feature variable
- For scene variables, use the drop down arrow to open the Select Single Feature dialog box and select a feature or create a new feature variable
- For feature variables, use the ellipsis button to open the Select Single Feature dialog box and select a feature or create a new feature variable
- For class variables, use the drop-down arrow to select from existing classes or create a new class
- For level variables, use the drop-arrow to select from existing levels
- For image layer variables, select the image layer you want to use for the update operation
- For thematic layer variables, select the thematic layer you want to use for the update operation
- For map variables, select the map name you want to use for the update operation
- For array variables, select the array you want to use for the update operation.

Variable

Select an existing variable or enter a name to create a new one. If you have not already created a variable, the Create Variable dialog box will open. Only scene variables can be used in this field.

Operation

This field displays only for object and scene variables. Select one of the arithmetic operations:

Value	Description
=	Assign a value.
+=	Increase by value.
-=	Decrease by value.
x=	Multiply by value.
/=	Divide by value.

Assignment

If Scene Variable or Object Variable is selected, you can assign either a value or a feature. This setting enables or disables the remaining parameters. If Image Layer Variable or Thematic Layer Variable is selected, you can assign either a layer or index.

Value

This field displays only for Scene and Object variables. If you have selected to assign by value, you may enter a value or a variable. To enter text use quotes. The numeric value of the field or the selected variable will be used for the update operation.

Feature

This field displays only for scene and object variables. If you have chosen to assign by feature you can select a single feature. The feature value of the current image object will be used for the update operation.

Comparison Unit

This field displays only for Scene and Object variables. If you have chosen to assign by feature, and the selected feature has units, then you may select the unit used by the process. If the feature has co-ordinates, select Co-ordinates to provide the position of the object within the original image or select Pixels to provide the position of the object within the currently used scene.

Arithmetic Expression

For all variables, you may assign an arithmetic expression to calculate a value.

Array Item

The Array Item parameter appears when ‘by array item’ is selected in the Assignment parameter. You may create a new array item or assign an Image Object Domain.

5.3 Compute Statistical Value

Perform a statistical operation on the feature distribution within an image object domain and store the result in a scene variable.

5.3.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

5.3.2 Active Classes

Choose the list of active classes for the classification.

5.3.3 Algorithm Parameters

Variable

Select an existing variable or enter a name to create a new one. If you have not already created a variable, the Create Variable dialog box will open. Only scene variables can be used in this field.

Operation

Select one of the statistical operations listed in table 5.0 on the current page, *Available Operations for Compute Statistical Value Algorithm*.

Table 5.0. Available Operations for Compute Statistical Value Algorithm

Value	Description
Number	Count the objects of the currently selected image object domain
Sum	Return the sum of the feature values from all objects of the selected image object domain
Maximum	Return the maximum feature value from all objects of the selected image object domain
Minimum	Return the minimum feature value from all objects of the selected image object domain
Mean	Return the mean feature value of all objects from the selected image object domain
Standard Deviation	Return the standard deviation of the feature value from all objects of the selected image object domain
Median	Return the median feature value from all objects of the selected image object domain
Quantile	Return the feature value, where a specified percentage of objects from the selected image object domain have a smaller feature value

Parameter

If you have selected the quantile operation, specify the percentage threshold.

Feature

Select the feature that is used to perform the statistical operation. (This parameter is not used if you select Number as your operation.)

Unit

- You may select a unit for the operation
- For positional features (such as X Max) you can choose Pixels from the drop-down list to return the relative object position (such as the position within a tile or subset).
- If you select a positional feature you can select Co-ordinates in the drop-down list to use the absolute co-ordinates if desired.

5.4 Compose Text

Assign a string variable text parts. Content of variable will be replaced by the text assigned with parameters text prefix and text suffix.

5.4.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

5.4.2 Algorithm Parameters

Result

Enter the name of the variable that will store the composed text.

Text Prefix

Edit the beginning of the composed text: the text suffix is attached afterwards. You can enter one of the following:

- Text (between quotation marks)
- A number value (without quotation marks)
- A variable (without quotation marks)

Text Suffix

Edit the end of the composed text: it is attached after the text prefix. You can enter one of the following:

- Text (between quotation marks)
- A number value (without quotation marks)
- A variable (without quotation marks)

Examples Some possible combinations are shown below:

Text prefix	Type	Text suffix	Type	Composed Text
“class”	Text	1	Number value	class1
CLASSNAMEID	Variable	ID	Variable	class1

5.5 Update Region

Modify a region variable. You can resize or move the region defined by a region variable, or enter its co-ordinates. Alternatively, you can use the co-ordinates of an image object bounding box or the active pixel for a region update.

5.5.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Array

5.5.2 Algorithm Parameters

Variable

Select an existing region variable or create a new one. To create a new variable, type a name for the new variable and click OK to open the Create Region Variable dialog box for further settings.

Mode

Select the operation for modifying the region defined by the region variable (see table 5.1, *Update Region Algorithm Modes*).

5.6 Update Image Object List

The Update Image Object List algorithm allows you to modify list variables, add objects, remove objects, and clear and compact lists.

5.6.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Superobject; Sub-objects; Linked Objects

Table 5.1. Update Region Algorithm Modes

Value	Description
Set min/max co-ordinates	Set each existing co-ordinate of the region by entering values for minimum and maximum co-ordinates, see below.
Set by origin/extent	Set each existing co-ordinate of the region by entering values for the Origin and the Extent, see below.
Move	Move a region by entering absolute Move in values for each existing co-ordinate, see below.
Resize	Shrinking or grow a region from its center. Enter absolute or percent values for each existing co-ordinate, see Unit and Resize in below. Because growing is measured from the center, the resulting region co-ordinates might be negative. In this case, the region is shifted so that the respective origin co-ordinate is 0. Examples: • If a region (100, 100), [100, 100] is grown by 100% the result is (50, 50), [200, 200]; no shifting is needed. • If a region (0, 0), [100, 100] is grown by 100% the result would be (50, 50), [150, 150]; instead the region is shifted by +50 in x- and y direction to (0, 0), [200, 200].
From object	Use co-ordinates of the bounding box including all image objects in the image object domain.
From array	Select a user-defined array (for more information, please consult the user guide).
Check bounds	A region can be fully or partly outside the scene, for example after initializing a region variable from a main map to use it in a rescaled map. This mode makes sure that region is fitted within the scene specified in the process domain.Examples: • If a region (100, 100), [9999, 9999] should be applied to a scene of (500,500), select Check bounds to truncate the region to (100, 100), [500, 500] • If a region (100, 100),[9999, 9999] should be applied to a scene of (500,500), select Check bounds to truncate the region to (0, 0), [500, 500].
Active Pixel	Define a region based on the active pixel/voxel. The co-ordinates of the active pixel/voxel are used for the origin of a region with the extent of 1 in each dimension.
Assign	Set the region to the same values like the region specified in Assign Region.

5.6.2 Algorithm Parameters

Variable

Select an existing image object list variable or create a new one. The computed value will be stored in the image object list variable.

Mode

Select the operation to use to modify the object list.

- Add Image Objects adds all objects in the current image object domain to the selected image object list. All image object domains are supported
- Remove Image Objects removes all image objects in the current image object domain from the list. All image object domains are supported
- Clear List removes all image objects from the selected image object list. The Execute image object domain is supported.
- Compact List removes all invalid objects from the list. The Execute image object domain is supported.

Feature

Select which feature you wish to use for sorting.

Sorting Order

Select the sorting order for the image objects.

5.7 Update Feature List

Update Feature List lets you to create a list that contains features, which can be later exported (for example as project statistics). The feature list can contain any combination of features, all of which can be added and removed from a list. Features can also be added and removed between lists and entire lists can be deleted.

5.7.1 Supported Domains

Execute

5.8 Automatic Threshold

Calculates a threshold value to be used for multi-threshold segmentation.

```
⊟··· ▪ auto_trsh
      <0.001s   at New Level: delete
    ▪ <0.001s   trsh=auto threshold on Layer 1
      0.047s    creating 'New Level': dark <= trsh < bright on Layer 1
```

Figure 5.1. Typical usage of automatic threshold algorithm in combination with multi-threshold segmentation

Figure 5.2. Result of multi-threshold segmentation with rule set as described above

The threshold can either be determined for an entire scene or for individual image objects. Depending on this setting, it is stored in a scene variable or in an image object variable. You can use the variables as a parameter for multi-threshold segmentation.

```
⊟··· ▪ auto_trsh_dmn
    ▪ dark at  New Level: obj.trsh=auto threshold on layer
      dark at  New Level: dark <= obj.trsh < bright on layer
```

Figure 5.3. Use of variables as parameters in a multi-threshold segmentation

Using the automatic threshold algorithm in combination with multi-threshold segmentation allows you to create fully automated and adaptive image analysis algorithms based on threshold segmentation. A manual definition of fixed thresholds is not necessary.

The algorithm uses a combination of histogram-based methods and the homogeneity measurement of multi-resolution segmentation to calculate a threshold dividing the selected set of pixels into two subsets, so that heterogeneity is increased to a maximum.

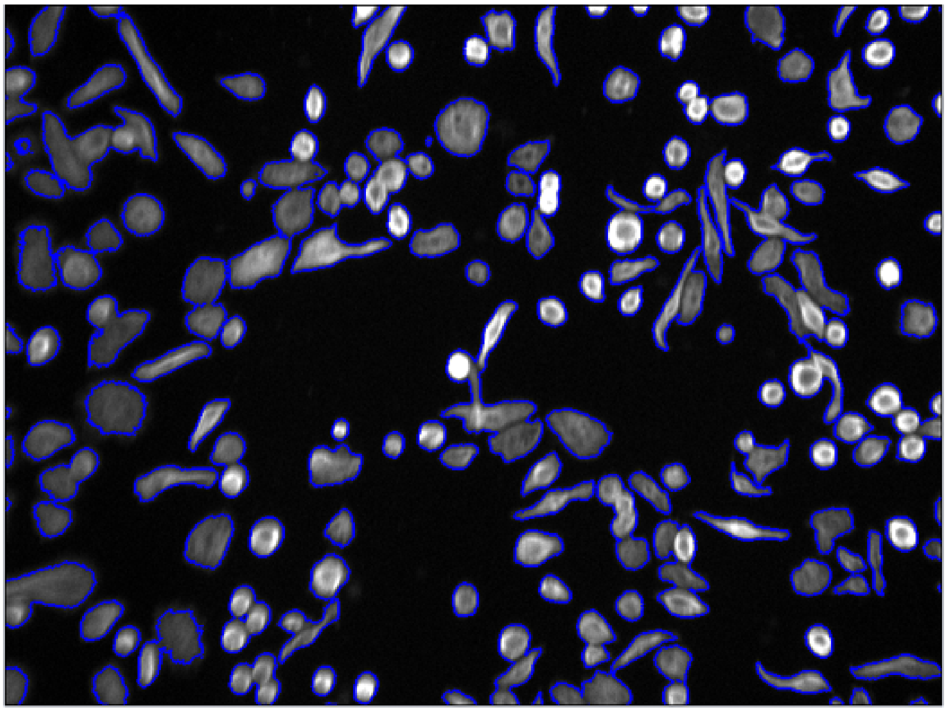


Figure 5.4. Result of automatic threshold calculation for fluorescent microscope image

5.8.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

5.8.2 Algorithm Parameters

Image Layer

Select the image layer you want to be used for automatic threshold determination.

Value Range

Define the value range that is taken into account for automatic threshold determination.

- Entire Value Range the complete threshold value range is taken into account.
- Restricted Value Range the threshold values within the specified interval (Min. Value Max.Value) are taken into account.

Min. Value

If the value range parameter is set to Restricted Value Range, define a minimum value for the automatically determined threshold.

Max. Value

If the value range parameter is set to Restricted Value Range, define a maximum value for the automatically determined threshold.

Threshold

Select the variable where the threshold value is to be saved. You can either select a numeric scene variable or a numeric image object variable (type Double). Saving threshold values in scene variables is most useful in combination with the Pixel Level domain. Saving threshold values in image object variables allows you to differentiate between the image objects of different image object domains.

Quality

Select the variable in which a quality control parameter for the threshold value is to be stored. The quality control parameter reflects the internal criterion according to which the threshold is determined. The higher its value, the more distinct the segmentation into image objects. You can use the quality control parameter to implement fully automated segmentation processes splitting up the image object domain into image objects that meet the predefined requirements.

5.9 Update Array

The Update Array algorithm is used to update user-defined arrays in rule sets. For more information on creating arrays, please consult the Definiens Developer XD 2.0.4 user guide.

5.9.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

5.9.2 Algorithm Parameters

Array

Select an array from the drop-down list.

The array may be number, string, class, image layer, thematic layer, level, feature, region or map name.

Mode

The following options are available to modify the array:

1. **Add Values**: Select a value to add to the array
2. **Add Based From Array Items**: Add a value from a specified array
3. **Add Based From Feature**: Add a value to an array derived from a feature
4. **Update Value**: Enter an index or value pair for the value to be updated
5. **Remove By Value**: Select a value to remove from the array
6. **Remove By Index**: Select the index of the value to be removed
7. **Clear**: Remove *all* values from the array
8. **Sort**: Sort array values in ascending or descending order.

Selecting options 1–3 will display the Add Value parameter. Select ‘only once’ for a single value; selecting ‘always’ allows duplicates.

6 Basic Object Reshaping Algorithms

Basic reshaping algorithms modify the shape of existing image objects. They execute operations such as merging image objects and splitting them into their sub-objects.

6.1 Remove Objects

Merge image objects in the image object domain. Each image object is merged into the neighbor image object with the largest common border. This algorithm is especially helpful for clutter removal.

6.1.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

6.1.2 Algorithm Parameters

Target Class

Click the ellipsis button to open the Edit Classification Filter dialog box and select the target classes to which the image objects in the image object domain will be merged.

Show Advanced Paramaters

Changing this value to Yes activates the following options:

Merge By Merge By allows you merge image objects based on shape or color. Merging by shape merges image objects to their neighbors, based on the longest common border. Merging by color merges neighbors based on the smallest layer value difference.

Merging by shape activates the following parameters:

- If Use Threshold has a value of Yes, the Border Threshold parameter is activated

- Enter a value in Border Threshold – only image objects with a common border length longer than or equal to this threshold will be merged
- If Use Legacy Mode is set to Yes, the algorithm will not look for a common superobject. (If a superobject level exists, objects may not be completely removed.)

Merging by color activates the following parameters:

- If Use Threshold has a value of Yes, the Color Threshold parameter is activated.
  - In Layer Usage, select the layers to be analyzed
  - Enter a value in Color Threshold – only image objects with a color difference smaller than or equal to neigboring image objects will be merged
  - If Use Legacy Mode is set to Yes, the algorithm will not look for a common superobject. (If a superobject level exists, objects may not be completely removed.)

6.2 Merge Region

Merge all image objects in the image object domain.

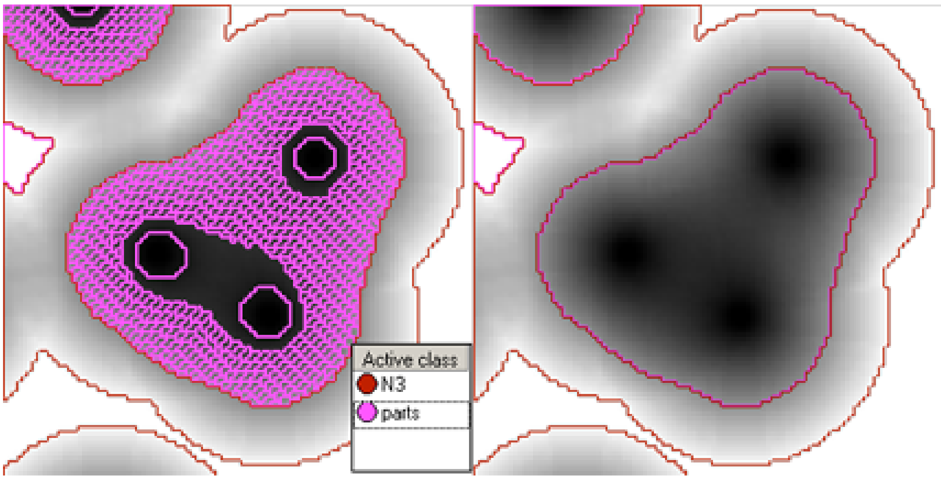


Figure 6.1. Result of merge region algorithm on all image objects classified as parts

6.2.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

6.2.2 Algorithm Parameters

Fusion Super Objects

Enable the fusion of affiliated super objects.

Use Thematic Layers

Enable to keep borders defined by thematic layers that were active during the initial segmentation of this image object level.

6.3 Grow Region

Enlarge image objects defined in the image object domain by merging them with neighboring image objects ("candidates") that match the criteria specified in the parameters.

The grow region algorithm works in sweeps. That means each execution of the algorithm merges all direct neighboring image objects according to the parameters. To grow image objects into a larger space, you may use the Loop While Something Changes checkbox or specify a specific number of cycles.

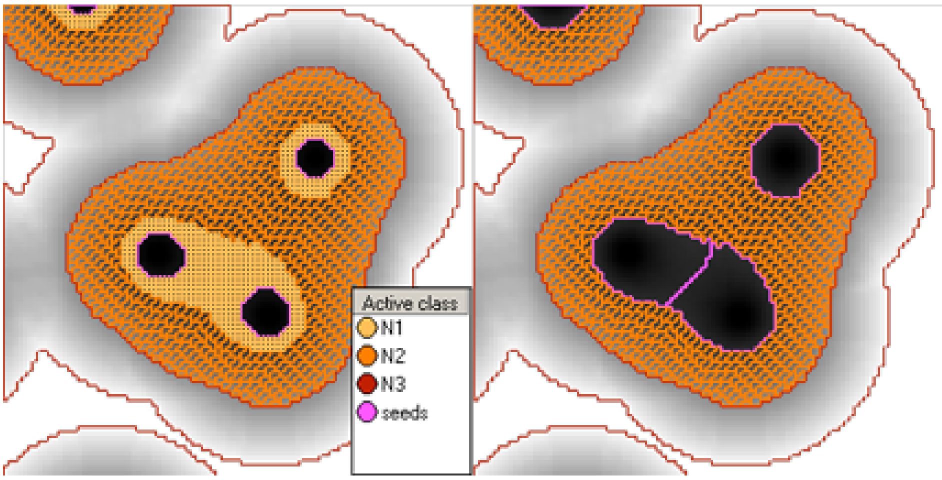


Figure 6.2. Result of looped grow region algorithm on image objects of class seed and candidate class N1. Note that the two seed objects in the image center grow to fill the entire space originally covered by objects of class N1 while still being two separate objects

6.3.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

6.3.2 Algorithm Parameters

Candidate Classes

Choose the classes of image objects that can be candidates for growing the image object.

Fusion Super Objects

Enable the fusion of affiliated super objects.

Candidate Condition

Choose an optional feature to define a condition that neighboring image objects need to fulfill in addition to be merged into the current image object.

Use Thematic Layers

Enable to keep borders defined by thematic layers that were active during the initial segmentation of this image object level.

6.4 Convert to Sub-objects

Split all image objects of the image object domain into their sub-objects.

6.4.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

6.4.2 Algorithm Parameters

None

6.5 Convert Image Objects

Convert image objects to a specified type, based on the differences among image object types with regard to their spatial connectivity and dimensions.

6.5.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

6.5.2 Algorithm Parameters

Image Object Type

Convert all image objects in the image object domain to image objects of the specified type:

Type	Description
Connected 2D	Convert all image objects in the image object domain to connected 2D image objects. The conversion is performed only for image objects that are not of the type connected 2D.
Connected 3D	Convert all image objects in the image object domain to connected 3D image objects. Internally, all image objects in the image object domain are converted to connected 2D image objects. Connected 3D image objects are then created based on the overlaps of connected 2D image objects over the slices. Connected 3D image objects are continuous with respect to slices. They may have several disconnected parts within a single slice. A special situation occurs if you have multiple image object levels and some parts of a connected 3D image object belong to different superobjects. In that case, the superobjects are merged automatically. If merging is not possible, then single disconnected superobjects are generated.
Disconnected	Convert all image objects in the image object domain to disconnected image objects. The algorithm tries to create a single image object per class. If some 3D image objects to be merged belong to different superobjects, the conversion works according to the Fusion of Superobjects settings; see below.

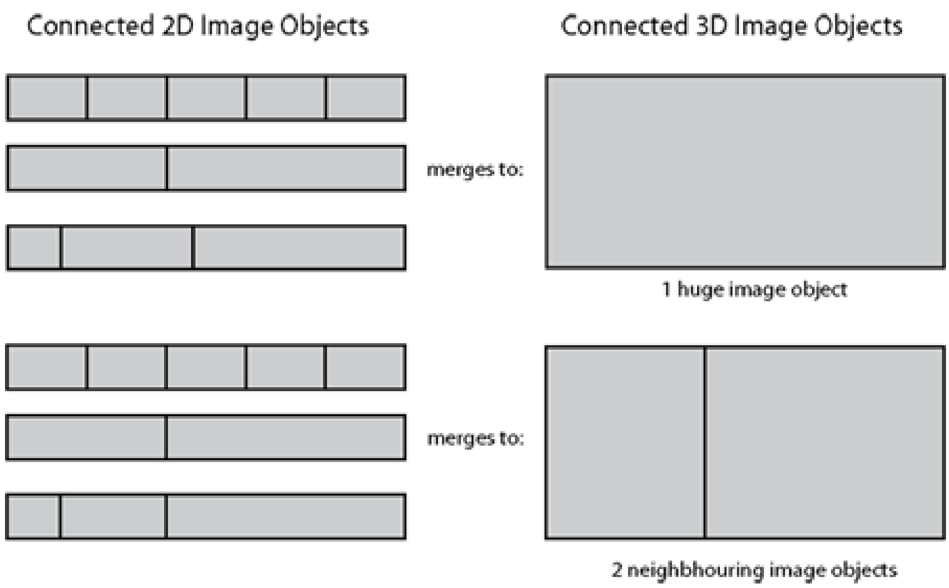


Figure 6.3. Merging scheme of overlapping image objects between slices.

Fusion of Super Objects

This parameter enables you to specify the effect on super objects. If this parameter is set to Yes, super objects will also be fused. This will usually have the effect that affected super objects will be converted to disconnected objects. If this parameter is set to No, the super objects stay untouched. In this case, fusion of the active image objects is restricted by the extent of the super objects.

- If the value is yes, superobjects are merged. If superobjects cannot be merged into a single connected image object they are merged into single disconnected image objects (default)
- If the value is no, superobjects are not merged and only image objects having the same superobject are merged. Consequently, there can be several disconnected image objects per class.

6.6 Cut Objects at Region

Cut all image objects within the image object domain that overlap the border of a given region. Each image object to be cut is split into two image objects, called pieces: one is completely within the region and one is completely outside.

6.6.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

6.6.2 Algorithm Parameters

Region

Select or enter the name of an existing region. Alternatively, you can enter the coordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner, and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: $(x_G, y_G, z_G, t_G), [R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Object Type Inner Pieces

Define the type of all image objects (pieces) located inside the region, no matter if they are cut or not:

Value	Description
Keep current	Keep current type (default)
Connected 2D	Convert resulting image objects to connected 2D image objects.
Connected 3D	Convert resulting image objects to connected 3D image objects (only available with Definiens Developer XD 2.0.4)
Disconnected	Convert resulting image objects to disconnected image objects.

Classify Inner Pieces

Select Yes to classify all image objects (pieces) located inside the region, no matter if they are cut or not.

Object Type Outer Pieces

Define the type of all cut image objects (pieces) located outside the region. The same settings are available for Object Type Inner Pieces.

Classify Outer Pieces

Select Yes to classify all cut image objects (pieces) located outside the region.

Class for Inner Pieces

Select or create a class. To create a new class, type a name for the new class and click OK to open

Class for Outer Pieces

Select or create a class. To create a new class, type a name for the new class and click OK to open.

7 Advanced Object Reshaping Algorithms

Advanced reshaping algorithms include sophisticated algorithms supporting a variety of complex object shape transformations.

7.1 Shape Split (Prototype)

7.1.1 Supported Domains

Execute; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

7.1.2 Algorithm Parameters

Maximum Border Ratio

Measures the relative border length of the cut line for both resulting objects. Cutting is only executed if the measured value is below this threshold for both objects. Use a zero value to disable this feature. This parameter is only used for contraction mode.

Maximum Cut Point Distance

Measures the distance of the cutting points on the object border. Cutting is only performed if this value is below this threshold. Enter a zero value to disable this feature.

Maximum Border Length

Restricts the maximum border length of the smaller objects. Enter a zero value to disable this feature.

7.2 Multiresolution Segmentation Region Grow

Grow segmented image objects according to the multiresolution segmentation criteria. For a detailed description of all parameters, refer to the algorithm *Multiresolution Segmentation* (p 34).

7.2.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

7.2.2 Image Layer Weights

Enter weighting values the higher the weight assigned to an image layer, the more weight will be given to that layer's pixel information during the segmentation process.

You can also use a variable as a layer weight.

7.2.3 Thematic Layer Usage

Thematic layer usage flags

Scale Parameter

Scale parameter

Candidate Classes

Candidate classes filter

7.2.4 Composition of Homogeneity Criteria

Shape

Weight of the shape criterion

Compactness

Weight of the compactness criterion

7.3 Image Object Fusion

Define a variety of growing and merging methods and specify in detail the conditions for merger of the current image object with neighboring objects.

Image object fusion uses the term 'seed' for the current image object. All neighboring image objects of the current image object are potential candidates for a fusion (merging). The image object that would result by merging the seed with a candidate is called the target image object.

A class filter enables users to restrict the potential candidates by their classification. For each candidate, the fitting function will be calculated. Depending on the fitting mode,[1] one or more candidates will be merged with the seed image object. If no candidate meets all fitting criteria no merge will take place.

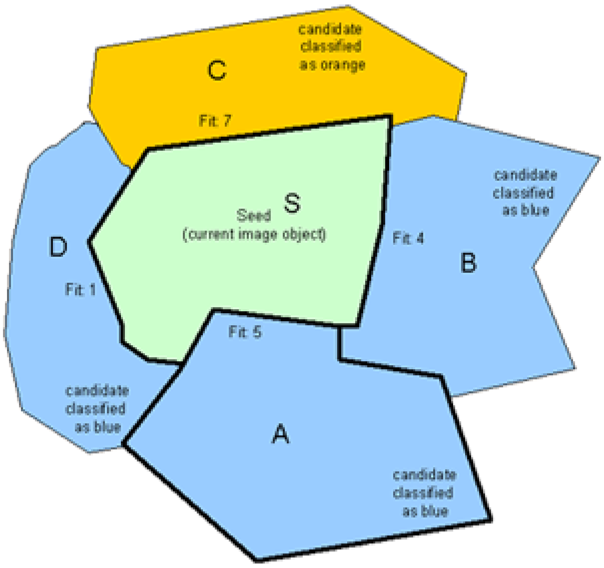


Figure 7.1. Example for image object fusion with seed image object S and neighboring objects A , B , C and D

7.3.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

7.3.2 Candidate Settings

Enable Candidate Classes

Select Yes to activate candidate classes. If the candidate classes are disabled the algorithm will behave like a region merging.

1. If you do not need a fitting function, we recommend that you use the algorithms Merge Region and Grow Region. They require fewer parameters for configuration and provide higher performance.

Candidate Classes

Choose the candidate classes you wish to consider. If the candidate classes are distinct from the classes in the image object domain (representing the seedclasses), the algorithm will behave like a growing region.

7.3.3 Fitting Function

The fusion settings specify the detailed behavior of the Image Object Fusion algorithm.

Fitting Mode

Choose the fitting mode (see table 7.1, *Fitting Mode Options for Image Object Fusion Algorithm*).

Table 7.1. Fitting Mode Options for Image Object Fusion Algorithm

Value	Description
All fitting	Merges all candidates that match the fitting criteria with the seed.
First fitting	Merges the first candidate that matches the fitting criteria with the seed.
Best fitting	Merges the candidate that matches the fitting criteria in the best way with the seed.
All best fitting	Merges all candidates that match the fitting criteria in the best way with the seed.
Best fitting if mutual	Merges the best candidate if it is calculated as the best for both of the two image objects (seed and candidate) of a combination.
Search mutual fitting	Executes a mutual best fitting search starting from the seed. The two image objects fitting best for both will be merged. Note: These image objects that are finally merged may not be the seed and one of the original candidate but other image objects with an even better fitting

Fitting Function Threshold

Select the feature and the condition you want to optimize. The closer a seed candidate pair matches the condition, the better the fitting.

Use Absolute Fitting Value

Enable to ignore the sign of the fitting values. All fitting values are treated as positive numbers independent of their sign.

7.3.4 Weighted Sum

Define the fitting function. The fitting function is computed as the weighted sum of feature values. The feature selected in Fitting Function Threshold will be calculated for the seed, candidate, and the target image object. The total fitting value will be computed by the formula. Fitting Value = (Target * Weight) + (Seed * Weight) + (Candidate * Weight) To disable the feature calculation for any of the three objects, set the according weight to 0.

Target Value Factor

Set the weight applied to the target in the fitting function.

Seed Value Factor

Set the weight applied to the seed in the fitting function.

Candidate Value Factor

Set the weight applied to the candidate in the fitting function:

Typical Settings (TVF, SVF, CVF)	Description
1, 0, 0	Optimize condition on the image object resulting from the merge
0, 1, 0	Optimize condition on the seed image object
0, 0, 1	Optimize condition on the candidate image object
2, −1, −1	Optimize the change of the feature by the merge

7.3.5 Merge Settings

Fusion Super Objects

This parameter defines the behaviour when the seed and candidate objects selected for merging have different super objects. If enabled, the super objects will be merged with the sub objects. If disabled the merge will be skipped.

Thematic Layers

Specify the thematic layers that are to be considered in addition for segmentation. Each thematic layer used for segmentation will lead to additional splitting of image objects while enabling consistent access to its thematic information. You can segment an image using more than one thematic layer. The results are image objects representing proper intersections between the thematic layers.

Compatibility Mode

Select Yes from the Value field to enable compatibility with older software versions (version 3.5 and 4.0). This parameter will be removed with future versions.

7.3.6 Classification Settings

Define a classification to be applied to the merged image objects.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class.

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

7.4 Border Optimization

Change the image object shape by either adding sub-objects from the outer border to the image object or removing sub-objects from the inner border of the image object.

7.4.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

7.4.2 Border Optimization Settings

Candidates

Choose the classes you wish to consider for the sub-objects. Sub-objects need to be classified with one of the selected classes to be considered by the border optimization.

Destination

Choose the classes you wish to consider for the neighboring objects of the current image object. To be considered by the Dilatation, sub-objects need to be part of an image object classified with one of the selected classes. To be considered by the Erosion sub-objects need to be moveable to an image object classified with one of the selected classes. This parameter has no effect for the Extraction.

Operation

Choose from the following values:

Value	Description
Dilatation	Removes all Candidate sub-objects from its Destination superobject inner border and merges them to the neighboring image objects of the current image object.
Erosion	Removes all Candidate objects from its Seed superobject inner border and merges them to the neighboring image objects of Destination domain.
Extraction	Splits an image object by removing all sub-objects of the Candidate domain from the image objects of Seed domain.

7.4.3 Classification Settings

The resulting image objects can be classified.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class If you do not use the class description, we recommend you use the Assign Class algorithm instead.

7.5 Morphology

Smooth the border of image objects by the pixel-based binary morphology operations Opening or Closing. This algorithm refers to image processing techniques based on mathematical morphology.

7.5.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Superobject; Linked Object; Image Object List

7.5.2 Morphology Settings

Operation

Decide between the two basic operations: Opening or Closing. Conceptually, imagine using Opening for sanding image objects and Closing for coating image objects. Both will result in a smoothed border of the image object. Open Image Object removes pixels from an image object. Opening is defined as the area of an image object that can completely contain the mask. The area of an image object that cannot contain the mask completely is separated.

Figure 7.2. Opening operation of the morphology algorithm

Close Image Object adds surrounding pixels to an image object. Closing is defined as the complementary area to the surrounding area of an image object that can completely contain the mask. The area near an image object that cannot contain completely the mask is filled; thus comparable to coating. Smaller holes inside the area are filled.

Mask

Define the shape and size of mask you want. The mask is the structuring element, on which the mathematical morphology operation is based. In the Value field, the chosen Mask pattern will be represented on one line. To define the binary mask, click the ellipsis button. The Edit Mask dialog box opens.

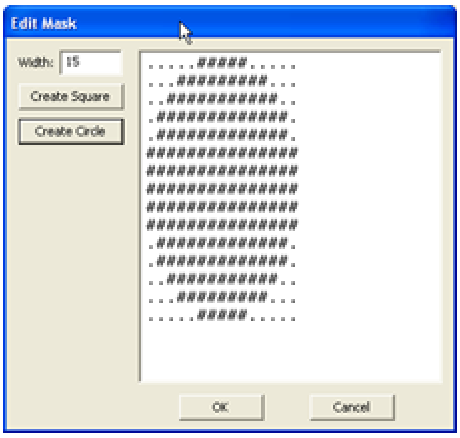


Figure 7.3. Edit Mask dialog box

To modify the binary mask you have the following options:

- Change the Width of the mask by entering new positive number.
- Create Square helps you to create a quadratic mask. Enter the dimensions. Start trying with values similar to the size of areas +1 you want to treat by sanding or to fill by coating. (Square masks perform less-precise operations and produce fewer artifacts than circle masks do.)
- Create Circle helps you to create a circular mask. Enter the side length. Start trying with values similar to the size of areas +1 you want to treat by sanding or to fill by coating. Alternatively, you can directly define a binary mask in the mask text field using a full-stop for FALSE and the hash symbol (#) for TRUE.

3.5 Compatibility Mode

Select Yes from the Value field to enable compatibility with older software versions (version 3.5 and 4.0). This parameter will be removed with future versions.

7.5.3 Classification Settings

When the operation Open Image Object is active, a classification will be applied to all image objects sanded from the current image object. When using the Close Image Object operation, the current image object will be classified if it gets modified by the algorithm.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

7.6 Watershed Transformation

Calculate an inverted distance map based on the inverted distances for each pixel to the image object border. Afterwards, the minima are flooded by increasing the level (inverted distance). Where the individual catchment basins touch each other (watersheds), the image objects are split.

The watershed transformation algorithm is commonly used to separate image objects from others. Image objects to be split must already be identified and classified.

7.6.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Superobject; Linked Object; Image Object List

7.6.2 Watershed Settings

Length Factor

The Length factor is the maximal length of a plateau, which is merged into a catchment basin. Use the toggle arrows in the Value field to change to maximal length. The Length Factor must be greater or equal to zero.

7.6.3 Classification Settings

Define a classification to be applied if an image object is cut by the algorithm.

Active Classes

Choose the list of active classes for the classification.

Erase Old Classification If There Is No New Classification

- If you select Yes and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is deleted.
- If you select No and the membership value of the image object is below the acceptance threshold (see classification settings) for all classes, the current classification of the image object is kept.

Use Class Description

- If Yes is selected, class descriptions are evaluated for all classes. The image object is assigned to the class with the highest membership value.
- If No is selected, class descriptions are ignored. This option delivers valuable results only if Active Classes contains exactly one class

If you do not use the class description, we recommend you use the Assign Class algorithm instead.

8 Pixel-Based Object Reshaping Algorithms

Pixel-based reshaping algorithms modify the shape of image objects by adding or removing pixel/voxels according to given criteria.

8.1 Pixel-Based Object Resizing

Grow or shrink image objects based on pixel criteria. Typically, the Relative Area of Classified Objects feature is used to find suitable seed image objects. These are grown or shrunken to larger or smaller image objects. In addition you can use this algorithm to smooth the surface of image objects by growing or shrinking. You can choose to resize in one, two or three dimensions.

8.1.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

8.1.2 Algorithm Parameters

Resizing modes

Growing Grow each seed image object. The starting extents of the seed image objects are lost.

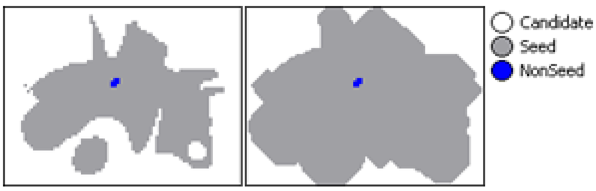


Figure 8.1. Sample starting classification (left) and after 10x growing (right)

Coating Add an new image object around each seed image objects and grows it. The seed image objects continue to exist as separate image objects with unchanged extent.

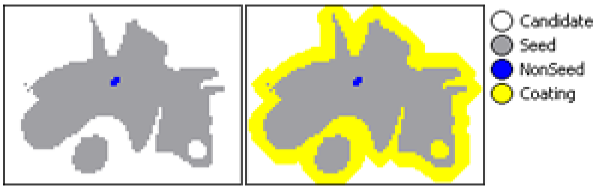


Figure 8.2. Sample starting classification (left) and 10x coating (right). Coating works from outside the Candidate image objects to the center

Shrinking Add an new image object inside each candidate image objects and grows it. A candidate image object shrinks by the extent of the new image object. This mode works similar like coating, but inside of candidate image objects.

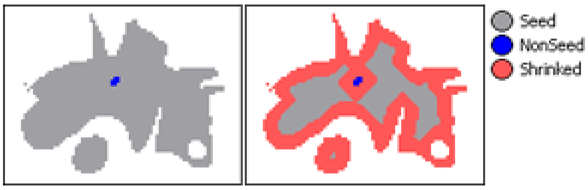


Figure 8.3. Sample starting classification (left) and after 10x shrinking (right). Shrinking works from outside the Candidate image objects to the center

Class for New Image Objects

Select a class to be assigned to the new image objects. This feature is available for Coating or Shrinking modes, but not for Growing.

Preserve Current Object Type

Select the type of image object to determine how image objects of the target class are merged.

Value	Description
Yes	Newly created image objects are 2D-connected. Other image objects maintain their connectivity (default)
No	Newly created image objects are disconnected. Any overwritten objects or split sub-objects are disconnected.

8.1.3 Candidate Object Domain Parameters

Define the set of neighboring image objects whose pixels are considered for growing. A pixel can be only added to an image object that is part of the candidate object domain. This feature is available for Coating or Shrinking modes, but not for Growing.

Class Filter

Select a candidate class. Image objects that are selected in the image object domain are automatically excluded from the candidates.

Threshold Condition

Define an additional threshold condition to define the candidate image object domain. Only pixels that belong to image objects that fulfill the threshold condition are considered for resizing.

8.1.4 Pixel Level Constraint Parameters

Pixel Layer Constraint 1; Pixel Layer Constraint 2

In addition, you can define one or two independent image layer intensity conditions to be fulfilled by candidate pixels on a given image layer. The candidate pixels are only added to the set of active pixels if the conditions defined are fulfilled.

Layer

Select any image layer to be used for the pixel layer constraint.

Operation

Select a comparison operation to be executed. An image layer must be selected.

Value	Description
<	Less than. (Default)
<=	Less than or equal to
=	Equal to
>	Greater than
>=	Greater than or equal to

Reference

Select the type of value used for comparing to the pixel layer intensity value. An image layer must be selected

Value	Description
Absolute value	Compare with an absolute layer intensity value that you can define; see Value below. It also can be represented by a variable.
Value of current pixel	Compare with the layer intensity value of the current pixel. You can define a Tolerance, see below.

Value

Enter a layer intensity value used as threshold for the comparison operation. Alternatively, you can select a feature or a variable. To create a new variable, type a name for the new variable and click OK to open the Create Variable dialog box for further settings. An absolute value has to be selected as a reference option.

Tolerance

Enter a value used as tolerance for the threshold Value of the comparison operation. The value of the current pixel has to be selected as a reference option. Alternatively, you can select a feature or a variable.

Tolerance Mode

Select a calculation mode for the Tolerance value of the threshold value of the comparison operation. The value of the current pixel must be selected as a reference option.

Value	Description
Absolute	The Tolerance value represents a percentage value. For example, 20 means a tolerance of 20 gray values.
Percentage	The Tolerance value represents a percentage value. For example, 20 means a tolerance of 20% of the threshold value.

8.1.5 Candidate Surface Tension Parameters

Use surface tension options to smooth the border of the resizing object.

Additionally, you can edit the settings so that the shapes of image objects are smoothed with no significant growing or shrinking.

Surface tension uses the Relative Area of Classified Objects region feature of pixels of a given class to optimize the image object shape while resizing. Within a cube of given size

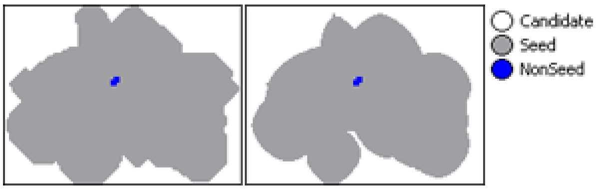


Figure 8.4. Sample classification after 10x growing without (left) and with (right) surface tension

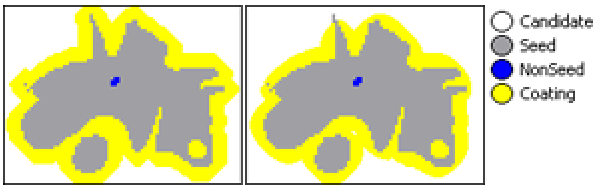


Figure 8.5. Sample classification after 10x coating without (left) and with (right) surface tension

(figure 8.8) around the current candidate pixel, the ratio of relative area of seed pixels to all pixels inside the box is calculated.

If the result is according to a given comparison operation (figure 8.9) with a given value, the current candidate is classified as seed; otherwise it keeps its current classification.

Reference

Choose to use either seed pixels of a current image object or pixels of a given class for surface tension calculation. Your choice influences smoothing while resizing.

None Surface tension is not active.

Object Surface tension is calculated based on image objects: Within the calculation box, only pixels of the candidate image objects are mentioned. This allows you to smooth the border of each seed image object without taking neighbor seed image objects into account.

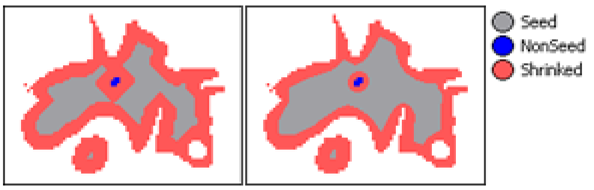


Figure 8.6. Sample classification after 10x shrinking without (left) and with (right) surface tension

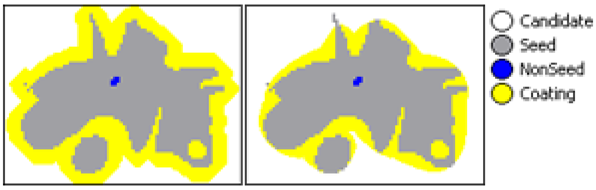


Figure 8.7. Sample classification after 10x coating without (left) and with 10x smoothing coating (right) based on surface tension

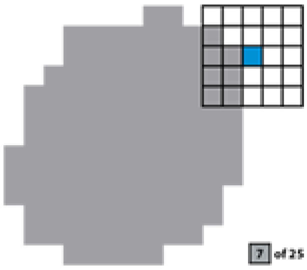


Figure 8.8. Example of calculation of the surface tension of seed objects (gray) is based on pixels within a box around the current candidate pixel (blue)

Class Surface tension is calculated based on a given class (figure 8.10); within the calculation box, all pixels of a given candidate class are taken into account. This enables you to smooth the border around neighboring image objects of the same class.

Class Filter

Select a candidate class used to measure the relative area. A class must be defined as Reference.

Operation

Select a comparison operation to be executed.

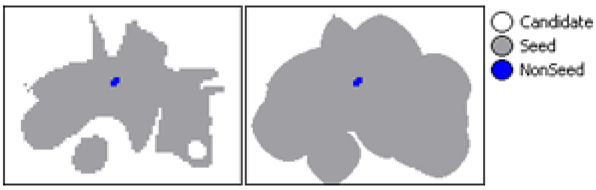


Figure 8.9. Object-based surface tension calculation while growing keeps image objects more separated

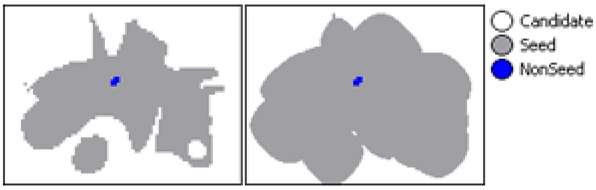


Figure 8.10. Class-based surface tension calculation while growing smooths multiple image objects of a class

Value	Description
>=	Greater than or equal to (default)
<	Less than
<=	Less than or equal to
=	Equal to
>	Greater than

Value

Enter a value for the surface tension calculation. Alternatively, you can select a feature or a variable. To create a new variable, type a name for the new variable and click OK to open the Create Variable dialog box for further settings.

Box Size in X and Y

Enter the pixel size of the square box around the current candidate pixel in x - and y -directions to use for the surface tension calculation. The integer value must be uneven; even integer values are changed to the next higher integer. You can enter a value as a threshold. Alternatively, you can select a feature or a variable. To create a new variable, type a name for the new variable and click OK to open the Create Variable dialog box for further settings. The default value is 5.

Box Size in Z

Similar to Box Size in X and Y. The default value is 1.

8.1.6 Size Limits Parameters

These parameters limit the sizes of growing and shrinking operations.

Min Object Size

The Minimum Object Size parameter stops an object shrinking when it reaches a minimum size. It is very useful for preventing a shrinking object from disappearing (that is attaining a size of zero).

Max Object Size

The Maximum Object Size parameter stops an object from growing when it reaches a maximum size. It is very useful for preventing a growing object from ‘leaking’.

8.2 Pixel-Based Density Filter

Create new classified image objects from pixel/voxels of image objects in the image object domain according to the class density in the surroundings. You can use this algorithm to smooth the shape of image objects by removing protrusions.

The class density calculation is based on the Relative Area of Classified Objects Region feature of pixel/voxels of a given class. For each candidate pixel/voxel of image objects in the image object domain, a cubic box of given size is used to calculate the ratio of relative area of classified pixel/voxels to all pixel/voxels inside the box.

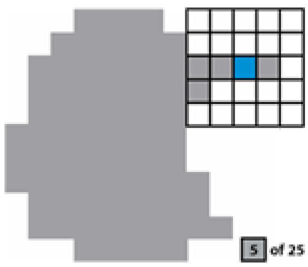


Figure 8.11. Example of calculation of the class (gray) density is based on pixels within a box around a candidate pixel (blue)

If the class density is according to given comparison operations with a given value the current candidate pixel/voxel is classified as target class, otherwise it keeps its current classification. Image objects of the target class are merged with each execution of the algorithm.

8.2.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

8.2.2 Algorithm Parameters

Target Class

Select a class to be assigned to image objects according to *density criteria* on page 108.

Preserve Current Object Type

Select the type of image object to determine how image objects of the target class are merged.

Value	Description
Yes	Newly created image objects are 2D-connected. Other image objects maintain their connectivity (default)
No	Newly created image objects are disconnected. Any overwritten objects or split sub-objects are disconnected.

8.2.3 Growing and Shrinking Directions Parameters

Growing can be enabled or disabled for each dimension. It can also be partially enabled so that growing occurs towards positive or negative co-ordinates only.

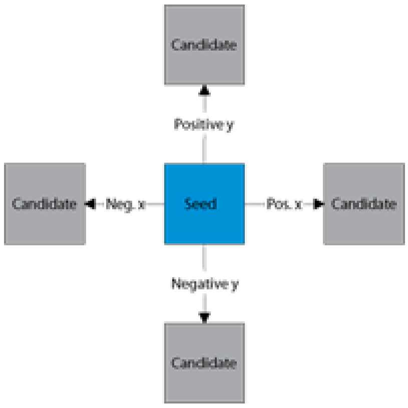


Figure 8.12. Growing directions for two dimensions

X, Y and Z Directions

Define the search order around active pixels. You can do this for each co-ordinate axis separately:

Value	Description
Yes	Enable resizing.(Default for x and y -directions.)
Only Positive	Restrict resizing in direction of positive co-ordinates only.
Only Negative	Restrict resizing in direction of negative co-ordinates only.
No	Disable resizing. (Default for z -direction.)

Balance Growing Directions by Pixel Size

If you work with a three-dimensional scene that has a different voxel size for the z-direction compared to the x - and y -directions, you can choose to take the voxel resolution into account. This ensures that image objects are resized evenly in all directions. Resizing must be enabled for z -direction.

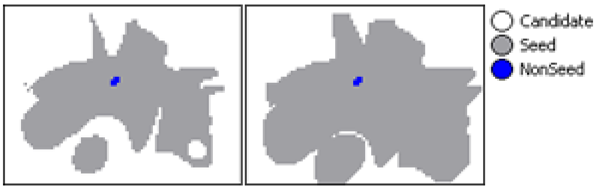


Figure 8.13. Example of 10x growing in x -direction is restricted to the positive direction and in y -direction to the negative direction

Value	Description
No	Growing is performed in all directions with the same preference.
Yes	In this case the growing takes the extension of the voxel into account. For each growing step in the z direction the number of growing steps in the xy direction equals the ratio of the voxel dimensions, for example the xy extend to z extend.

8.2.4 Density Criteria Parameters

You can define one to three independent density criteria that need to be fulfilled by candidate pixel/voxels on a given image layer. Only if all criteria are fulfilled is the candidate pixel/voxel classified.

Reference

To activate a density criterion, select whether pixel/voxels of a current image object or pixel/voxels of a given class are used for class density calculation. That way, you influence the way of smoothing while resizing:

Value	Description
None	Density criterion is not active. (Not available for Density Criterion 1)
Object	Class density is calculated based on the image object that contains the current candidate pixel/voxel. Within the calculation box, only pixel/voxels of this image object, are mentioned. This allows you to smooth the border of the image object without taking neighbor image objects into account.
Class	Class density is calculated based on a given class (see below). Within the calculation box, all pixel/voxels of the given class are taken into account. This allows you to smooth the border around neighboring image objects of the same class.

Class Filter

Select a candidate class to use for calculation of class density. A class must be defined as Reference.

Operation

Select a comparison operation to be executed:

Value	Description
>=	Greater than or equal to. (Default)
<	Less than
<=	Less than or equal to
=	Equal to
>	Greater than

Value

Enter a value, or select a feature, or a variable used a threshold for the class density calculation. To create a new variable, type a name for the new variable and click OK to open the Create Region Variable dialog box for further settings.

Box Size in X and Y

Enter the pixel/voxel size of the square box around the current candidate pixel/voxel in x and y -directions, used for the class density calculation. The integer value must be an odd number; even-numbered integer values are changed to the next higher integer.

You can enter a value, or select a feature, or a variable used as threshold. To create a new variable, type a name for the new variable and click OK to open the Create Region Variable dialog box for further settings. The default value is 1.

Box Size in Z

Similar to Box Size in X and Y. The default value is 1.

8.3 Pixel-Based Shape Processing Filters

Provides a set of different filters for processing image objects with respect to various shape criteria:

Filter	Description
Pixel Count	Detects and modifies low-level pixel patterns.
Corner Pixels	Detects the corner pixel of an image object. In conjunction with the Connect filter, it can be used to cut image objects according to their shape.
Connect	Creates a pixel connection between two linked image objects. Can be used to connect pixels detected by the Corner Pixels or Hole Cutting Pixels filter and thus to cut image objects.
Inner Border	Reclassifies all pixels at the inner border of an image object.
Outer Border	Reclassifies all pixels at the outer border of an image object.
Hole Cutting Pixels	Detects pairs of pixels at the inner and outer borders of image objects. In conjunction with the Connect filter, it can be used to cut holes out of image objects.

8.3.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

8.3.2 Algorithm Parameters

Target Class

Select a class to be assigned to the processed pixels.

Preserve Current Object Type

Select Yes or No:

Value	Description
Yes	Ensures that the image object type of processed image objects does not change due to pixel removal.
No	Changes the type of all modified image objects to disconnected, regardless of their current state. You can use the *Convert Image Objects algorithm* (p 82) to make sure processed image objects are assigned the desired image object type after processing.

Filter Type

Select a filter.

Pixel Count The Pixel Count filter detects basic pixel patterns in the 4-neighborhood of a pixel and reclassifies all pixels in the domain where a pattern occurs. It can be useful to connect border pixels to a 4-connected structure or to detect pixel bridges in image objects. In addition to the basic parameters mentioned above, you can set the following parameters:

Corner Pixels The Corner Pixels filter detects the corners of an image object specified by the image object domain. The detected corners can be concave or convex.

The filter enables you to further connect detected concave corner pixels using image object links. In conjunction with the Connect filter mode, these connections can be used to create lines along which the image objects can be cut according to their shape. In addition to the basic parameters mentioned above, you can set the following parameters:

Connect The Connect filter creates a pixel line between two linked image objects. It checks all image objects specified in the image object domain for image object links to another image object. If an image object link exists, it creates a one pixel line between the two image objects according to the filter parameters and deletes the image object link.

This mode is specifically designed to be used in conjunction with the Corner Pixel filter detection of concave corners.

In addition to the basic parameters mentioned above, you can set the following parameters:

- Candidate Class Filter specifies a class filter for candidate objects. Only pixels that belong to an object of one of the specified classes will be taken into account for the connection
- Exclude Border Pixels[1] specifies whether pixels located at the border of areas defined by the Candidate Class Filter parameter are taken into account for the connection:
  - If the value is Yes, border pixels are not taken into account
  - If the value is No, all pixels of the specified areas are taken into account
- Distance Mode specifies how distances between two pixels are calculated:
  - Spatial Distance determines the shortest connection along the candidate pixels that can be found between two image objects. Uses predefined values 1 for 4-connected pixels and 2 for diagonally connected pixels to calculate pixel distance
  - Color Contrast determines the shortest connection along the candidate pixels that can be found between two image objects. Uses the difference between the pixel intensities of two pixels for an image layer to calculate pixel distance.
- Image Layer specifies the layer used to calculate pixel distances. It is only available if the Distance Mode parameter is set to Color Contrast

Inner Border The Inner Border filter reclassifies all neighboring pixels of an image object that touch an inner border of the object. This filter is useful to detect holes within image objects. There are no additional parameters for this filter.

1. If you enable the Exclude Border Pixels parameter, it might not be possible to establish a valid connection between two image objects. To avoid endless loops or other unwanted effects, make sure that the rule set you use can handle this situation properly.

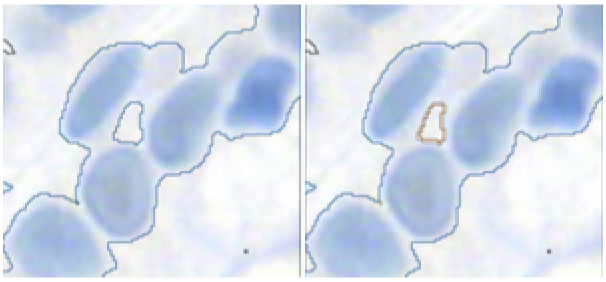


Figure 8.14. Result of Inner Border filter processing

Outer Border The Outer Border filter reclassifies all neighboring pixels of an image object that touch the outer border of the object. There are no additional parameters for this filter.

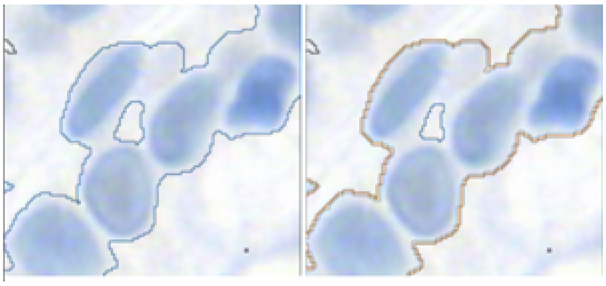


Figure 8.15. Result of Outer Border filter processing

Hole Cutting Pixels The Hole Cutting Pixels filter detects inner border pixels with close proximity to the outer border and related pixels on the outer border.

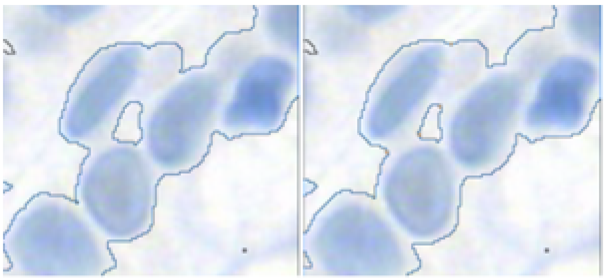


Figure 8.16. Result of Hole Cutting Pixels filter processing

Both pixels are linked so that you can use the Connect filter to cut out so-detected inner holes from the object. In order to avoid structures that are quite similar to a nearly closed hole the filter always detects two pairs of inner and outer pixels that have a maximum distance to each other.

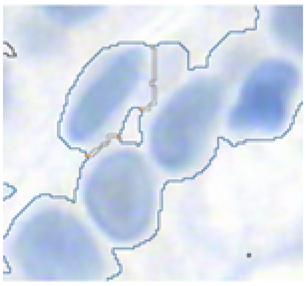


Figure 8.17. Result of subsequent Connect filter processing

Mode Parameters

Corner Type Specify the type of corner pixels to search for:

Value	Description
Concave	Detects and reclassifies concave corner pixels.
Convex	Detects and reclassifies convex corner pixels.
All	Detects and reclassifies both concave and convex corner pixels.

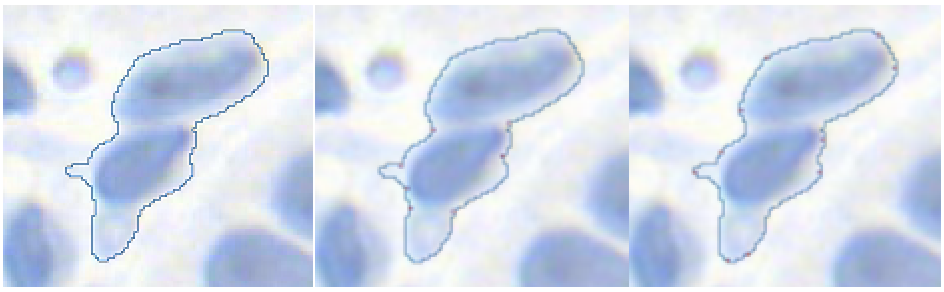


Figure 8.18. Image object (left) and its results after detection of concave (middle) and convex (right) corner pixels

Leg Length For each border pixel, the filter analyzes the angle of the border by creating an angle between the pixel and the adjacent border pixels to the left and to the right. The leg length describes the length of the angle legs in border pixels. Increase the leg length value to apply a smoothening to the real image object border. Typical values lie within the range 2 to 9.

Minimum Angle Specify the minimum angle (based on the leg length) that must be present at a border pixel so that it will be considered as a corner.

If you enter a value of 0°, every border pixel with an angle different to 180° is considered as a potential corner. The larger the value, the more acute the minimum angle that is needed for a corner to be accepted.

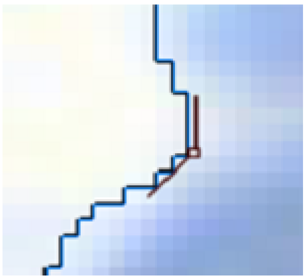


Figure 8.19. Angle measurement at a border pixel with leg length 3 pxls (approx. 45° for a concave corner pixel)

Connect Corners for Cutting This parameter is only available if Corner Type parameter is set to Concave. It lets you specify whether detected corners are to be connected using image object links:

Value	Description
Yes	Does not connect corners using image object links.
No	Detects and connects matching corners using image object links. The connections can be used as input for further processing with the Connect filter. Unconnected corners are not instantiated.

Max. Cut Length This parameter is only available if Corner Type parameter is set to Concave and if Connect Corners for Cutting parameter is set to Yes.

Specify the maximum length of the direct connection between two corner pixels. Corners are only be linked for connection if their distance is below this threshold. Values are measured in pixels. A value of 0 will disable this restriction.

Cut Length/Border Length This parameter is only available if Corner Type parameter is set to Concave and if Connect Corners for Cutting parameter is set to Yes. Specify the maximum ratio of the direct distance between two corners and the distance along the object border. Corners are only linked for connection if the ratio is below this threshold.

Values must lie within the range 0 to 1. A value of 1 will link all corner pairs, a value of 0 will reject all corner pairs.

The example below shows the result of concave corner pixel detection with a leg length of 3 and a minimum angle of 35°. The Connect Corners for Cutting parameter is set to Yes; a cut length/border length ratio of up to 0.3 is allowed.

The selected pixel objects (marked in red and green) are connected using an image object link.

Reference Specify the reference for pixel pattern detection:

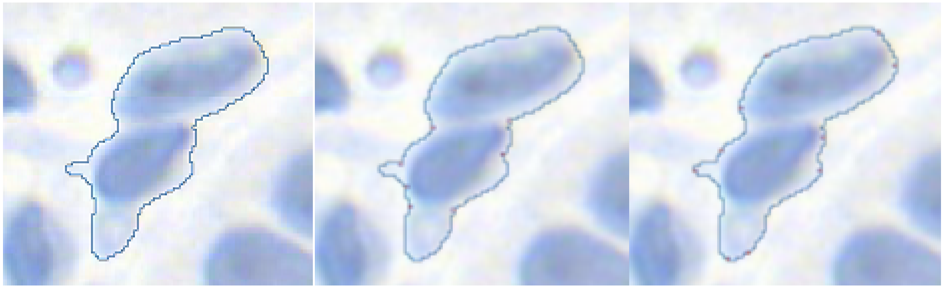


Figure 8.20. Result of Pixel Corner Pixels filter processing with the settings described above

Value	Description
Object	Detects patterns of pixels within the same object.
Class	Detects patterns of pixels assigned to the same class.

Class Filter This parameter is only available if Reference parameter is set to Class. Specify the class filter used for pixel pattern detection in Class reference mode.

Pixel Pattern Specify the pattern of target pixels to search for:

Table 8.-14. Pixel Pattern Settings

Value	Description
0	No neighboring pixels. Reclassifies isolated pixels
1	Exactly one neighboring pixel. Reclassifies end points of lines with a width of one pixel
2	Exactly two neighboring pixels. Combines the reclassification results of 2 (angle) and 2 (line) modes
2 (angle)	Exactly two neighboring pixels forming an angle with the center pixel. Reclassifies pixels that are part of a diagonal pixel bridge. Use this mode to convert 8-connected pixel structures into 4-connected pixel structures
2 (line)	Exactly two neighboring pixels forming a line with the center pixel. Reclassifies pixels (except end points) being part of lines with a width of one pixel
3	Exactly three neighboring pixels
4	Exactly four neighboring pixels. Reclassifies pixels that are completely surrounded by target pixels

The example below shows the result of a pixel count filter processing on the blue area using the 2 (angle) pixel pattern. The orange pixels are used as a reference.

Figure 8.21. Result of Pixel Count filter processing with the settings described in table 8.-14 on the preceding page, *Pixel Pattern Settings*

In the course of pixel processing, the orange pixel structure is converted into a 4-connected structure that can be merged into one connected image object.

Candidate Class Filter Class filter for candidate objects. Only pixels that belong to an object of one of the specified classes will be considered for the shortest path.

Exclude Border Pixels When this option is set to yes, the connection must not go along pixels at the border of the selected class

Search in Z Direction Search also in Z direction (3D images only)

Distance Mode Select the way pixel distances are computed using spatial distance, color contrast or color value.

9 Linking Operation Algorithms

Use linking operations algorithms to work with image object links. Image object links can be used to group multiple image objects in different areas of the image without creating a common superobject.

9.1 Create Links

This algorithm allows you to create and establish object links between image objects.

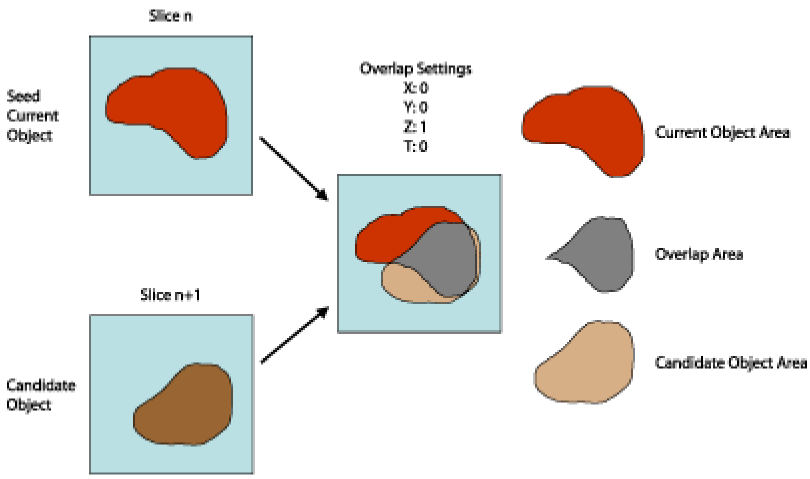


Figure 9.1. Overlap calculations of the Create Links algorithm

For $z = 1$ slices, n and $n+1$ are scanned and some overlap area is determined. Usually parameters x, y and t are zero. It may be useful to have values different from zero if structures such as blood vessels or time series are to be analyzed. For $x, y, t \neq 0$, the overlap region gets calculated with respect to the shifted template.

9.1.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Superobject; Subobjects; Linked Objects; Image Object List

9.1.2 Algorithm Parameters

Link Class

Select a class that will be assigned to the new links. This class represents the group name of the links and is used to compute statistics of the image objects belonging to this link class.

9.1.3 Candidate Object Domain Parameters

Class Filter

Specify for the classes of candidate image objects to be linked with the seed image objects.

Threshold Condition

Specify an additional condition for candidate objects. Only image objects meeting the specified condition will be considered for linking.

Map

You can specify a different map to the map selected in the image object domain. In this case, image object links between different maps will be created. This can be useful to express object relations between maps.

Candidate PPO

Select the PPO level to use for the objects from the next parent process. This parameter is only valid if domain and candidate maps are the same.

9.1.4 Overlap Settings Parameters

These parameters are used to specify the shift in space and time from the seed object to determine the position of objects to that will be linked.

Link To

- Use ‘all’ to link to all objects that match the conditions
- Use ‘best’ to create the link only to the candidate object that has the largest overlap in the set of all possible overlapping objects.

Overlap Calculation

Select one of the following options:

Value	Description
Do not use overlap	Overlap calculation is omitted. Link is created to any object matching the target conditions
Relative to larger object	The ratio of the overlap area to the area of the larger object (between seed and target object) is calculated
Relative to smaller object	The ratio of the overlap area to the area of the smaller object is calculated
Relative to current object	The ratio of the overlap area to the area of the current objects is calculated
Relative to candidate object	The ratio of the overlap area to the area of the candidate objects is calculated
Absolute [in pixels]	The absolute overlap area in pixels is calculated

X-Position Shift

Shift in pixels in the x direction (± integer values).

Y-Position Shift

Shift in pixels in the y direction (± integer values).

Z-Position Shift

Shift in pixels in the z direction (± integer values).

T-Position Shift

Shift in pixels in the t direction (± integer values).

Min. Required Overlap

Lower threshold for image object overlap. A link will only be created if the calculated overlap will exceed the specified threshold. Use 0 to disable this parameter.

- If the overlap calculation is set to Do Not Use Overlap, Min. Required Overlap is not available
- If the overlap calculation is set to Absolute [in pixels], enter any integer
- If the overlap calculation is set to any other option, enter a percentage represented by a float number between 0 and 1.

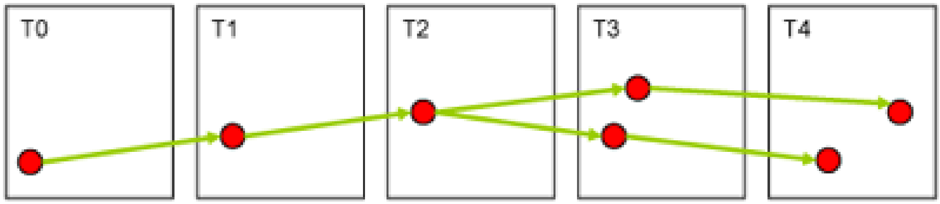


Figure 9.2. Incoming and outgoing links over multiple time frames. The red circles represent objects and the green arrows represent links

Transformation Parameter Set

This parameter is only available if the map specified in the image object domain is different to the map specified in the candidate object domain.

If you don't know the transformation parameters, you can use the Image Registration algorithm to perform an automatic affine registration and store the transformation matrix in a parameter set. Using this transformation information, you can then link objects that relate to each other in the original maps.

9.2 Delete Links

Delete all or selected object links. The parameters of this algorithm can be used to filter out specific links to be deleted. All parameters are cumulative; only links fulfilling all specified criteria will be deleted.

9.2.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

9.2.2 Algorithm Parameters

Link Class Filter

Classification filter for links to delete. Only links grouped under the selected classes will be deleted.

Class Filter

Classification filter for linked image objects. Only links that link to an object of one of the selected classes will be deleted.

10 Level Operation Algorithms

Level operation algorithms enable you to add, remove, or rename entire image object levels within the image object hierarchy.

10.1 Copy Image Object Level

Insert a copy of the selected image objects domain above or below the existing one.

10.1.1 Supported Domains

Image Object Level

10.1.2 Algorithm Parameters

Level Name

Enter the name for the new image object level.

Copy Level

Select whether the copied level is placed above or below the input level specified by the domain.

10.2 Delete Image Object Level

Delete the image object level selected in the image object domain.

10.2.1 Supported Domains

Image Object Level

10.3 Rename Image Object Level

Rename an image object level.

10.3.1 Supported Domains

Execute

10.3.2 Algorithm Parameters

Level to Rename

Select the image object level to be renamed.

New Level Name

Select or edit an image object level to be changed, and select or edit the new name for the level. If the new name is already assigned to an existing level, that level will be deleted. This algorithm does not change names already existing in the process tree.

11 Map Operations Algorithms

Map operations algorithms are used for working with multiple maps and can be used for the following actions:

- Local storage of temporary results
- Multi-scale image analysis and other workspace automation techniques, which enable you to process image data without using the Definiens Server
- Comparison of competing and concurrent analysis strategies on the same image data in parallel. This enables you to select the best results from each analysis and combine them into a final result
- Comparison of results on different sets of image data in parallel. This enhances interactive training approaches and helps to develop robust ruleware.

11.1 Copy Map

Copy a map, or part of a map – such as current image object – to a new map, or overwrite an existing map with the copy. 2D, 3D and 4D maps can be copied; the minimum size of a map to be created is 4×4 pixels.

11.1.1 Supported Domains

Execute; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

11.1.2 Algorithm Parameters

Source Region

Define a region within the source map. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner , and its size $[R_x, R_y, R_z, R_t]$. The input pattern is (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$.

Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Target Map Name

The map to be created by copying. Select a map name from the drop-down list or enter a new name. If you select an existing map, the copy will overwrite it. Alternatively, you can create a map variable.

Use Variable As Scale

Specify the scale for copying a map using a variable, rather than defining a numerical value.

Scale

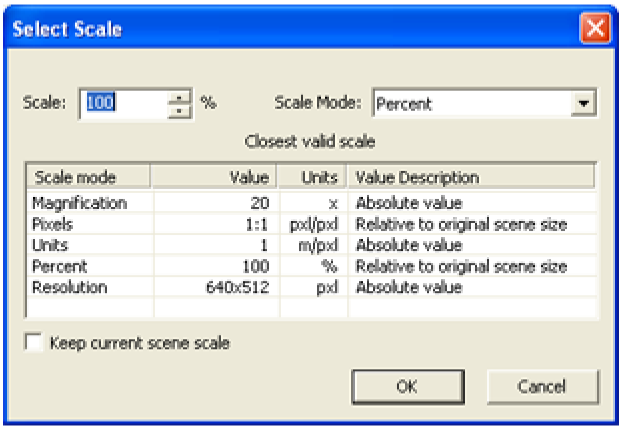


Figure 11.1. Select Scale dialog box

If you choose to resample, the Scale will refer to the original image data. If you choose the default Use Current Scene Scale, the map copy will have the same scale as the map (or part of a map) being copied. For example, if the main map is copied to map2 with the Scale at 50%, and map2 is copied to map3 with the Scale at 50%, map3 will be scaled to 50% of the main map, and not 50% of map2:

1. If you do not want to keep the current scale of the map for the copy, click the ellipsis button to open the Select Scale dialog box.
2. Selecting a scale different to the current scene scale lets you work on the map copy at a different magnification/resolution.
3. If you enter an invalid scale factor, it will be changed to the closest valid scale as displayed in the table below.
4. To change the current scale mode, select from the drop-down list. We recommend that you use the scaling mode consistently within a rule set as the scaling results may differ. Scaling results can differ depending on the scale mode; for example if you enter 40, you work at the following scales, which are calculated differently:

Options dialog box setting	Scale of the scene copy or subset to be created
Units (m/pixel)	40m per pixel
Magnification	40x
Percent	40% of the resolution of the source scene
Pixels	1 px per 40 px of the source scene

Clockwise Rotation Angle

This feature lets you rotate a map that you have copied, by a fixed value or with respect to a variable.

Resampling

Choose whether to apply Gaussian smoothing:

Value	Description
Smooth	Smoothing is applied. If smoothing is applied and downsampling occurs, the algorithm generates smoothed .tif files next to the project file (.dpr)
Fast	Smoothing is not applied. Select for faster processing.

Image Layers

Select an image layer to use in the new map. If no layer is selected, all will be used. Layers are copied if downsampling occurs and Smooth is selected in the Resampling field.

Copy Thematic Layers

Select a thematic layer to use in the new map. If no layer is selected, all will be used. Thematic vector layers are always copied and converted to thematic raster layers. Thematic raster layers are copied if downsampling occurs and Smooth is selected in the Resampling field.

Copying thematic vector layers can be performance-intensive because vector layers are converted to raster layers.

Thematic Layers

You may specify a thematic layer to use; if no layer is selected, all layers will be used.

Copy Image Object Hierarchy

Choose whether the image object hierarchy is copied with the map or not:

- If Yes is selected, the image object hierarchy of the source map will be copied to the new map.
- If No is selected, only selected image and thematic data will be copied to the new map.

Preserve Current Object Type

When this option is set to No, then created objects can be unconnected, so one object can have more than one polygon.

Visiblity Flag

If the value is set to Yes (the default), all maps are available from the map drop-down box. If it is set to No, the created map can be accessed, but cannot be displayed.

Compatibility Mode

Allows compatibility with previous software versions.

Scale Direction

Choose from one of the following:

- Plane only
- Plane and z
- Plane and time
- All directions

11.2 Delete Map

Delete a specified map or maps. There are no algorithm parameters.

11.2.1 Supported Domains

Execute; Maps

11.3 Synchronize Map

Copy image objects from one map to another.

11.3.1 Supported Domains

Image Object Level

11.3.2 Algorithm Parameters

Target Map Name

Required. Use the dropdown list to select an existing map or map variable, or create a new map variable and assign a value to it.

Region

Define a target region to which image objects are copied. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner , and its size $[R_x, R_y, R_z, R_t]$. The input pattern is (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Alternatively, you can create a region by entering numbers with that exact syntax: (origin x , origin y) - (extent x , extent y); for example (10,20) - (100,110).

Level

Required. Select the target image object level in the target map.

Class Filter

Select objects of classes on the target map (these can be overwritten). Click the ellipsis button to open the Edit Classification Filter dialog box. The default is none, which means objects of any class can be overwritten.

Threshold Condition

Select a threshold condition. Image objects matching the threshold will be overwritten. Click the ellipsis button to open the Select Single Feature dialog box. The default is none, which means objects of any class can be overwritten.

Clockwise Rotation Angle

This feature lets you rotate a map that you have synchronized, by a fixed value or with respect to a variable. If you have rotated a copied map using the Copy Map algorithm, you can restore it with the Synchronized Map algorithm by using the negative value of the angle of rotation.

Preserve Current Object Type

- If Yes is selected, the current object type is preserved for all affected objects
- If No is selected, modified objects can become disconnected objects.

Synchronize Complete Hierarchy

When this option is set to yes, then all levels on the target map will be affected.

11.4 3D/4D Settings

Define the map layout settings for multidimensional data sets.

Because Definiens software handles multidimensional data sets internally as a two-dimensional image, you have to define how these two-dimensional maps represent the corresponding data set.

11.4.1 Supported Domains

Execute

11.4.2 Algorithm Parameters

Mode

Select a mode:

Mode	Description
Noninvasive	The map size in x is assumed to be the 2D slice size in x and y. This is used to compute the number of slices. Example: Imagine a map with the size 256×1024. The noninvasive mode uses the map size x=256 to determine the slice size y=256. Thus, the map is handled as a 3D map consisting of 1024/256=4 slices of size 256×256 each.
2D extent	Enter the size of a single slice or a single frame; see below.
4D layout	Depending on the data set, you can enter the number of slices and the number of frames; see below.

Slice/Frame Size X & Y

Depending on the dataset, enter the x and y sizes per single slice or frame. (2D Extent must be selected.)

Number of Slices

Enter the number of slices of the data set. (4D Layout must be selected.)

Number of Frames

Enter the number of frames of the data set. (4D Layout must be selected.)

Distance Between Slices

Enter the distance between slices the distance you enter is relative to the xy resolution. The default is 0.5. For example:

- Slice Distance 1 means $x = y = z = 1$
- Slice Distance 2 means $2x = 2y = 1z$
- Slice Distance 0.5 means $x = y = 0.5z$

Time Between Two Frames

Enter the time elapsing between frames. The default is 0.5.

Co-ordinate of the First Slice

Enter the co-ordinate of the first slice, which determines the co-ordinates of other slices. The default is 0.

Start Time

Enter the time of the first frame, which determines the time of other frames. The default is 0.

11.5 Scene Properties

Select the magnification, pixel size and scene unit for a scene, via a rule set.

11.5.1 Supported Domains

Execute; Maps

11.5.2 Algorithm Parameters

Magnification

If appropriate, enter the magnification settings. This is most often used in microscope images, where a known magnification was used and the information was not embedded in the image format. You can also enter a variable in this field. Magnification can only be set for a scene that was not rescaled.

Scene Unit

Set the default unit for calculating feature values. Choose from pixels, kilometers, hectometers, decameters, meters, decimeters, centimeters, millimeters, micrometers, nanometers, angstroms, inches, feet, yards, statute miles and nautical miles.

Pixel Size

If you wish to link the size of your objects to a known scale (for instance in a geographical image) enter the scene unit that corresponds to one pixel.

12 Image Layer Operation Algorithms

Image layer operation algorithms are used to create or to delete image object layers. In addition, you can use them to apply filters to image layers at the pixel/voxel level.

12.1 Distance Map

The Distance Map algorithm creates a layer that contains the distances of pixels to objects of a selected class, using a flood-filling algorithm.

- In a time series, distance is not calculated between frames; only between slices in the $x-y$ direction
- In scaled maps, the distance corresponds to the map co-ordinates and is scaled with the map.

12.1.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

12.1.2 Algorithm Parameters

Distance To

The Distance To parameter lets you select a class; the distance is then calculated to all objects that border this class. The 8-*Neighborhood* (p 10) calculation is used as it allows diagonal distance measurements. If no class is selected, the distance of each pixel to its next image object border is calculated.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

12.2 Create Temporary Image Layer

Create a temporary image layer with values calculated from a selected feature for the image objects selected in the image object domain.

12.2.1 Supported Domains

Image Object Level

12.2.2 Algorithm Parameters

Layer Name

Select the default name for the temporary image layer or edit it.

Feature

Select a single feature that is used to compute the pixel values filled into the new temporary layer.

Value for Undefined

The value to be set, when feature value is undefined.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

12.3 Delete Layer

Delete a selected image layer or thematic layer. One image must be kept in the map. This algorithm is often used in conjunction with the Create Temporary Image Layer algorithm to remove this image layer after you finished working with it.

12.3.1 Supported Domains

Execute

12.3.2 Algorithm Parameters

Layer Type

Select a type of layer to be deleted – thematic layer or image layer.

Layer

Select a layer to be deleted.

12.4 Convolution Filter

Apply a convolution filter to the image. There are two options; a preset Gaussian smoothing filter (Gauss Blur) and a user-defined kernel. A convolution filter uses a kernel, which is a square matrix of a value that is applied to the image pixels. Each pixel value is replaced by the average of the square area of the matrix centered on the pixel.

12.4.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.4.2 Algorithm Parameters

Type

The Gauss Blur is a convolution operator used to remove noise and detail. The Custom Kernel enables the user to construct a kernel with customized values.

Expression

The formula for Gaussian Blur is:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

(Where σ is the standard deviation of the distribution.)

Advanced Parameter

Displays for Gauss Blur. Enter a value for the reduction factor of the standard deviation. A higher value results in more blur.

Custom Kernel

Displays only when Custom Kernel is selected. Click the ellipsis button on the right to open the Kernel dialog box (figure 12.1) and enter the numbers for the kernel.

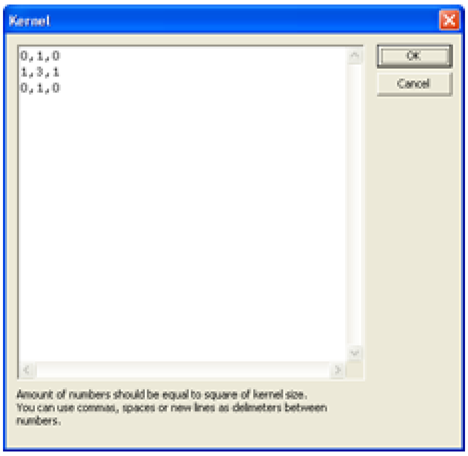


Figure 12.1. Kernel dialog box

The number of entries should equal the square of the kernel size entered in the 2D kernel size field. Use commas, spaces or lines to separate the values.

12.4.3 Kernel Parameters

2D Kernel Slice

Enter an odd number only for the filter kernel size. The default value is 3.

Number of Slices

This is available if Type is set to Gauss Blur. Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in x slices above and x slices below (x being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.4.4 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G), which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: $(x_G, y_G, z_G, t_G), [R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select ‘yes’ or ‘no’ to create a visible or hidden image layer.

Output Layer Type

Select the data type of the output layer. Available options are:

- As input layer
- 8-bit unsigned
- 16-bit unsigned
- 16-bit signed
- 32-bit unsigned
- 32-bit signed
- 32-bit float

12.5 Layer Normalization

The Layer Normalization algorithm offers two options to normalize images. The linear normalization filter stretches pixel values to the entire pixel value range. The histogram normalization changes pixel values based on the accumulated histogram of the image. The effect is illustrated in the following histograms:

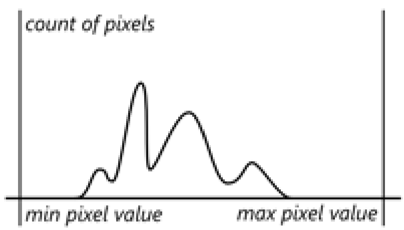


Figure 12.2. Sample histogram of original image

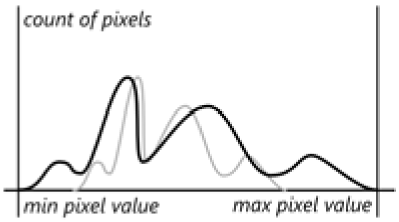


Figure 12.3. Example of historgram after linear normalization

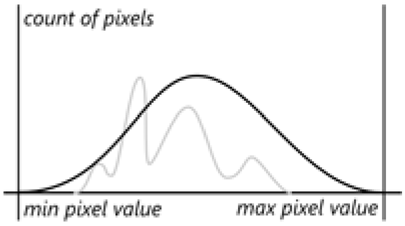


Figure 12.4. Sample of histogram after histogram normalization

12.5.1 Supported Domains

Pixel Level

12.5.2 Algorithm Parameters

Type

- If Linear is selected, a linear stretch is applied to the layer histogram
- If Histogram is selected, a histogram stretch is applied to the layer histogram.

12.5.3 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select ‘yes’ or ‘no’ to create a visible or hidden image layer.

Output Layer Type

Select the data type of the output layer. Available options are:

- As input layer
- 8-bit unsigned
- 16-bit unsigned
- 16-bit signed
- 32-bit unsigned
- 32-bit signed
- 32-bit float

12.6 Median Filter

Use the Median Filter algorithm to replace the pixel value with the median value of neighboring pixels. The median filter may preserve image detail better than a mean filter. Both can be used to reduce noise.

12.6.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.6.2 Kernel Parameters

2D Kernel Size

Enter a number to set the kernel size in one slice. The default value is 3.

Number of Slices

Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in x slices above and x slices below (x being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.6.3 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a name for the output layer or use the drop-down list to select a layer name to be used for output. If left empty, a temporary layer will be created. If a temporary layer is selected it will be deleted and replaced.

Output Layer Type

Select an output layer type from the drop-down list. Select As Input Layer to assign the type of the input layer to the output layer.

12.7 Sobel Operation Filter

Creates a layer by applying Sobel operator.

12.7.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.7.2 Kernel Parameters

2D Kernel Size

Enter a number to set the kernel size in one slice. The default value is 3.

Number of Slices

Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in x slices above and x slices below (x being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.7.3 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

Output Layer Type

Select the data type of the output layer. Available options are:

- As input layer
- 8-bit unsigned
- 16-bit unsigned
- 16-bit signed
- 32-bit unsigned
- 32-bit signed
- 32-bit float

12.8 Pixel Freq. Filter

The Pixel Frequency Filter algorithm scans the input layer and selects the color that is found in the greatest number of pixels. The frequency is checked in the area defined by the size of the kernel.

12.8.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

12.8.2 Kernel Parameters

2D Kernel Size

Enter a number to set the kernel size. The default value is 3.

Number of Slices

Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in x slices above and x slices below (x being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.8.3 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select ‘yes’ or ‘no’ to create a visible or hidden image layer.

Output Layer Type

Select the data type of the output layer. Available options are:

- As input layer
- 8-bit unsigned
- 16-bit unsigned
- 16-bit signed
- 32-bit unsigned
- 32-bit signed
- 32-bit float

12.9 Pixel Min/Max Filter (Prototype)

Creates layer where each pixel has the difference of kernel max or min value to the center value.

12.9.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

12.9.2 Algorithm Parameters

Mode

Choose the min/max filter mode – diff. brightest to center, diff. center to darkest or diff. brightest to darkest.

12.9.3 Kernel Parameters

2D Kernel Size

Enter a number to set the kernel size in one slice. The default value is 3.

Number of Slices

Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in *x* slices above and *x* slices below (*x* being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.9.4 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$.

Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a name for the output layer or use the drop-down list to select a layer name to be used for output. If left empty, a temporary layer will be created. If a temporary layer is selected it will be deleted and replaced.

Output Layer Visible

Select ‘yes’ or ‘no’ to create a visible or hidden image layer.

12.10 Edge Extraction Lee Sigma

The Edge Extraction Lee Sigma algorithm uses a specific edge filter that creates two individual layers from the original image. One layer represents bright edges, the other one dark edges.

To extract two layers – one with bright and one with dark edges – this algorithm must be applied twice with the appropriate settings changed. If two edge layers are created, it is important to give them two individual image layer aliases. Otherwise, the first existing layer would be overwritten by the second generated layer.

12.10.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.10.2 Algorithm Parameters

Sigma

Set the Sigma value. The Sigma value describes how far away a data point is from its mean, in standard deviations. A higher Sigma value results in a stronger edge detection; the default value is 5. The sigma value for a given window is:

$$\sum = \sqrt{\frac{\sigma^2}{(\bar{x})^2}}$$

If the number of pixels P within the moving window that satisfy the criteria in the formula below is sufficiently large (where W is the width, a user-defined constant), the average of these pixels is output. Otherwise, the average of the entire window is produced.

$$\left(1 - w\sum\right)P_{\text{Center}} \leq P \leq \left(1 + W\sum\right)P_{\text{Center}}$$

Edge Extraction Mode

- If Dark is selected, edges of darker objects are extracted
- If Bright is selected, edges of brighter objects are extracted

Input Layer

Use the drop-down list to select the input layer.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$.

Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a name for the output layer or use the drop-down list to select a layer. If a temporary layer is selected it will be deleted and replaced.

12.11 Edge Extraction Canny

Enhance or extract feature boundaries using Canny's algorithm. Edge extraction filters may be used to enhance or extract feature boundaries. The resulting layer typically shows high pixel values where there is a distinctive change of pixel values in the original image layer.

12.11.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.11.2 Algorithm Parameters

Algorithm

The Canny algorithm is provided.

Lower Threshold

Lower threshold is applied after higher threshold. During the first step, edges are detected and pixels with values lower than Higher Threshold are removed from detected edges. During the final step, non-edge pixels (those previously removed because values were less than higher threshold) with values higher than lower threshold are marked as edge nodes again. After applying the algorithm the first time, you can check results (edge pixel values) and the value for the threshold. Usually values for this field are from 0–5 – the default is 0.

Higher Threshold

After edges are detected, pixels with values lower than this threshold will not be marked as edge pixels. This allows removal of low intensity gradient edges from results. After applying the algorithm once, users can check the results (values of edge pixels) and find the correct value for the threshold. Usually values for this field are from 0–5 – the default is 0.

Gauss Convolution FWHM

Enter the width of the Gaussian filter in relation to full width at half maximum of the Gaussian filter. This field determines the level of details covered by Gaussian filter. A higher value will produce a wider Gaussian filter and less detail will remain for edge detection. Therefore, only high intensity gradient edges will be detected by Canny's algorithm. The range of the field is 0.0001–15. The default value is 1.

Input Layer

Use the drop-down list to select a layer to use for input.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is:

(x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Use the drop-down list to select a layer to use for output or enter a new name. Output is 32-bit float. If the name of an existing 32-bit float temporary layer is entered or selected, it will be used. If a temporary layer is selected it will be deleted and replaced.

Sample Results The table below shows results of typical settings:

Original Layer	Lower Threshold: 0 Higher Threshold: 0 Gauss Convolution FWHM: 0.2	Lower Threshold: 0.3 Higher Threshold: 0.6 Gauss Convolution FWHM: 0.2	Lower Threshold: 0.3 Higher Threshold: 0.69 Gauss Convolution FWHM: 0.2
			

12.12 Edge 3D Filter

The Edge 3D Filter algorithm creates a layer with 3D edge information.

12.12.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Object; Linked Objects

12.12.2 Algorithm Parameters

Smoothing Scale Factor

This parameter defines the sharpness of the detected edges. The value range is 0–1 – the larger the value, the more sharp an edge is displayed in the resulting image. Smaller values will progressively blur the detected edges until they are unrecognizable.

Return Option

- Edge 3D returns all edge signals
- Mean + Edge 3D displays bright edges only (mean plus pixel value)
- Mean - Edge 3D displays bright edges only (mean minus pixel value)

Edge Finding Method

- Abs. Deviation takes all pixels in the kernel
- Abs. Deviation of Pixels Along Color Edges only considers pixels on edges

12.12.3 Kernel Parameters

2D Kernel Size

Enter a number to set the kernel size in one slice. The default value is 3.

Number of Slices

Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in x slices above and x slices below (x being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.12.4 Layer Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G), which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G), $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a name for the output layer or use the drop-down list to select a layer name to be used for output. If left empty, a temporary layer will be created. If a temporary layer is selected it will be deleted and replaced.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

12.13 Surface Calculation

Use the surface calculation algorithm to derive the slope for each pixel of a digital elevation model (DEM). This can be used to determine whether an area within a landscape is flat or steep and is independent from the absolute height values. There is also an option to calculate aspect using Horn's method.

12.13.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.13.2 Algorithm Parameters

Layer

Select the layer to which the filter will be applied. Gradient Unit and Unit of Pixel parameters apply to slope calculations only.

Algorithm

- Slope uses the Zevenbergen-Thorne method [1]
- Aspect uses Horn's method. [2]

Gradient Unit

Available for slope. Select Percent or Degree from the drop-down list for the gradient unit.

Unit of Pixel Values

Enter the ratio of the pixel height to pixel size.

Input Layer

Use the drop-down list to select a layer for input.

1. Zevenbergen LW, Thorne CR (1987). Quantitative Analysis of Land Surface Topography. *Earth Surface Processes and Landforms*, 12(1):47–56
2. Horn BKP (1981). Hill Shading and the Reflectance Map. *Proceedings of the IEEE*, 69(1):14–47

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G) , which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: (x_G, y_G, z_G, t_G) , $[R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a name for the output layer or use the drop-down list to select a layer name to be used for output. If left empty, a temporary layer will be created. If a temporary layer is selected it will be deleted and replaced.

Output Layer Type

Select an output layer type from the drop-down list. Select As Input Layer to assign the type of the input layer to the output layer.

12.14 Layer Arithmetics

The layer arithmetic algorithm uses a pixel-based operation that enables the merger of up to four layers by mathematical operations $(+, -, \times, \div)$. The layer created displays the result of this mathematical operation. This operation is performed on the pixel level, which means that all pixels of the image layers are used. For example, Layer 2 can be subtracted from Layer 1. This would mean that whenever the same pixel value in both layers exists, the result would be 0.

Before or after the operation, the layers can be normalized. Furthermore, weights can be used for each individual layer to influence the result.

12.14.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.14.2 Algorithm Parameters

Input Layer

Select a layer to be used as the input for the filter.

Minimum Input Value

Enter the lowest value of the value range that will be replaced by the output value. The default is 0.

Maximum Input Value

Enter the highest value of the value range that will be replaced by the output value. The default is 255.

Output Value

The value that will be written in the raster layer. May be a number or an expression. For example, to add Layer 1 and Layer 2, enter Layer 1 + Layer 2. The following operations can be used in the expression:

- Four basic operations: $(+, -, \times, \div)$
- One power operation: ^
- Three Boolean operations (result of operation: 1 for true or 0 for false): < = >
- & → logical AND example: “Layer 1 & Layer 2” = smallest value is taken
- | → logical OR example: “Layer 1 | Layer 2” = largest value is taken and written into new image layer

Formula Examples

- `Layer 1+(Layer 2*10)`
- `(Layer 1>128)*255`
- `(Layer 1<0.5)*Layer 2+(Layer 1=0.5)*Layer 3+(Layer 1>0.5)*Layer 3`
- `Layer 1^0.5-Layer 2+Layer 3/8`

Note the spacing for operators and layer names. Use an expression (in quotation marks) to avoid creating variables, such as a variable with the name Layer 1. If a local layer needs to be referenced, it is necessary to manually enter the full expression (e.g. CA.Layer 1) in the field.

If variables have previously been created (for example, a user accidentally enters Layer 1 without quotation marks), the value of the variable will be used even if the same name is set in quotation marks subsequently. You will need to delete the variable or edit it in order to use the layer in the Output Value field.

Output Layer

Enter a name for the output layer or use the drop-down list to select a layer name to be used for output. If left empty, a temporary layer will be created. If a temporary layer is selected it will be deleted and replaced.

Output Region

Region within the output layer.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

Output Layer Type

This field is available if the output layer does not yet exist. Select a data type for the raster channel if it must be created:

- 8-bit unsigned
- 16-bit unsigned
- 16-bit signed
- 32-bit unsigned
- 32-bit signed
- 32-bit float

12.15 Line Extraction

The line extraction algorithm creates a layer and classifies the pixels of the input layer according to their line filter signal strength.

12.15.1 Supported Domains

Pixel Level

12.15.2 Algorithm Parameters

Line Direction

Enter the direction of the extracted line in degrees, between 0 and 179. The default value is 0.

Line Length

Enter the length of the extracted line. The default value is 12.

Line Width

Enter the length of the extracted line. The default value is 4.

Border Width

Enter the width of the homogeneous border at the side of the extracted line. The default value is 4.

Max Similarity of Line to Border

Enter a value to specify the similarity of lines to borders. The default value is 0.9.

Min Pixel Variance

Enter a value to specify the similarity of lines to borders. Use -1 to use the variance of the input layer. The default value is 0.

Min Mean Difference

Enter a value for the minimum mean difference of the line pixels to the border pixels. If positive, bright lines are detected. Use 0 to detect bright and dark lines.

Input Layer

Use the drop-down list to select the layer where lines are to be extracted.

Output Layer

Enter or select a layer where the maximal line signal strength will be written. If a temporary layer is selected it will be deleted and replaced.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

12.16 Pixel Filters Sliding Window

Creates a filtered image layer using various sliding window filter methods along orthogonal axes x, y or z .

12.16.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.16.2 Algorithm Parameters

Algorithm

Choose the algorithm used in filtering.

Input Image Layer

Select the image layer used for filtering. For some object-based filters, no input layer is needed.

Filtered Layer

Float-valued image layer which stores the filter results.

12.16.3 Filter Kernel Sizes Selection Parameters

Kernel Size in x

Kernel size used for sliding window filter, which should always be an odd number. If the kernel size is 1, this filter axis is not used.

Kernel Size in y

Kernel size used for sliding window filter, which should always be an odd number. If the kernel size is 1, this filter axis is not used.

Kernel Size in z

Kernel size used for sliding window filter, which should always be an odd number. If the kernel size is 1, this filter axis is not used.

12.17 Abs. Mean Deviation Filter (Prototype)

The Abs. Mean Deviation Filter algorithm creates an image layer, where each pixel has the absolute deviation from mean with Gaussian weight.

12.17.1 Supported Domains

Pixel Level; Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

12.17.2 Algorithm Parameters

Gauss Reduction

Reduction factor of standard deviation along the (unrotated) x axis

Return Option

Select the filter response option:

- Circular
- Elliptic:
  - Gauss reduction (y axis) is the reduction factor of the standard deviation along the (unrotated) y axis
  - Rotation angle (xy axis) is the y axis rotation angle in the xy plane
- Elliptic difference:
  - Gauss reduction (y axis) is the reduction factor of the standard deviation along the (unrotated) y axis
  - Rotation angle (xy axis) is the y axis rotation angle in the xy plane

12.17.3 Kernel Parameters

2D Kernel Slice

Enter an odd number only for the filter kernel size. The default value is 3.

Number of Slices

Enter the number of slices to be considered as part of the kernel. If a region is specified in the image object domain, the algorithm will use the region values in x slices above and x slices below (x being the number of slices entered). If there is no region, the entire area of the slices above and below will be considered part of the kernel. If there are insufficient slices or regions, only those available will be considered.

12.17.4 Layers Parameters

Input Layer

Select a layer to be used as the input for the filter.

Input Region

Define a region within the input image layer. Select or enter the name of an existing region. Alternatively, you can enter the co-ordinates of a region specified by its origin (x_G, y_G), which is the lower left corner and its size $[R_x, R_y, R_z, R_t]$. The input pattern is: $(x_G, y_G, z_G, t_G), [R_x, R_y, R_z, R_t]$. Alternatively, you can select a variable. To create a new variable, type a name for the new variable and click OK or press Enter to open the Create Variable dialog box for further settings.

Output Layer

Enter a layer name to be used for output. A temporary layer will be created if there is no entry in the field or if the entry does not exist. If an existing layer is selected it will be deleted and replaced.

Output Layer Visible

Select 'yes' or 'no' to create a visible or hidden image layer.

Output Layer Type

Select the data type of the output layer. Available options are:

- As input layer
- 8-bit unsigned
- 16-bit unsigned
- 16-bit signed
- 32-bit unsigned
- 32-bit signed
- 32-bit float

12.18 Contrast Filter (Prototype)

12.18.1 Supported Domains

Pixel Level

12.18.2 Algorithm Parameters

Input Layer

Input Layer

Temp Channel Alias

Name of generated temporary channel.

Calc Mode

Calculation Mode.

Channels Are Temp

New channels may be temporary or added to the project.

12.19 Fill Pixel Values

The Fill Pixel Values algorithm lets you fill in raster regions based on the image object domain, by interpolation from neighbor object pixels. It is generally suitable for interpolating non-data areas in continuously varying rasters (such as elevation models).

To use this function, it is necessary to create and classify objects that define the pixels to be interpolated (such as those with a zero value). This function works only on temporary image layers – these can be created or copied using the *Layer Arithmetics* on page 148 algorithm).

12.19.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Objects; Sub Objects; Linked Objects

12.19.2 Algorithm Parameters

Input Layer

Select the temporary image layer to be modified.

Class Filter

Object classes with known pixel values.

Calculation Mode

The following modes are available:

- Inverse Distance Weighting
- Bi-linear Interpolation (a legacy function)

IDW Distance Weight A general form of finding an interpolated value u at a given point x based on samples $u_i = u(x_i)$ for $i = 0, 1, \ldots, N$ using IDW is an interpolating function:

$$\phi(x, u) = \left(\sum_{i=0}^{N} \frac{(u-u_i)^2}{d(x,x_i)^p}\right)^{\frac{1}{p}}$$

derived from the minimizing condition

$$\frac{\partial \phi(x,u)}{\partial u} = 0$$

x denotes an interpolated (arbitrary) point, x_i is an interpolating (known) point, d is a given distance metric operator) from the known point x_i to the unknown point x . N is the total number of known points used in interpolation and p is a positive real number, called the power parameter.

IDW Distance Weight

This value corresponds to the exponent *p* in the formula for IDW distance weight.

13 Thematic Layer Operation Algorithms

Thematic layer operation algorithms are used to transfer data from thematic layers to image objects and vice-versa.

13.1 Assign Class by Thematic Layer

Assign the classes stored in a thematic layer to image objects. This algorithm quickly assigns all classes in thematic layers to an image object level. If an image object contains pixels with different classification within the thematic layer, the classification of the majority of the pixels is assigned to the image object.

Classes are matched by name between the thematic layer and the class hierarchy. The algorithm assumes that classes with the same names must be present in the class hierarchy. The class mode parameter sets the behavior if no matching class can be found.

13.1.1 Supported Domains

Image Object Level; Image Object List

13.1.2 Algorithm Parameters

Thematic Layer

Specify the thematic layer where classification values will be read.

Thematic Layer Attribute

Specify the thematic layer attribute column that contains classification values.

Class Mode

Specify the behavior when the class specified in the thematic layer is not present in the rule set.

- Skip if class does not exist – class assignment is skipped if the class specified in the thematic layer is not present in the rule set
- Use default class – use a default class if the class specified in the thematic layer is not present in the rule set
- Create new class – a random, unique color will be assigned to the new class in the Class Hierarchy window.

Default Class

Only available if class mode is Use Default Class. Specify the default class for this mode.

13.2 Synchronize Image Object Hierarchy

Change an image object level to exactly represent the thematic layer. Image objects smaller than the overlapping thematic object will be merged; image objects intersecting with several thematic objects will be cut.

13.2.1 Supported Domains

Image Object Level; Image Object List

13.2.2 Algorithm Parameters

Thematic Layer

Select a thematic layer to be represented.

13.3 Read Thematic Attribute

Create and assign local image object variables according to a thematic layer attribute table. A variable with the same name as the thematic attribute will be created, attached to each image object in the domain and filled with the value given by the attribute table.

13.3.1 Supported Domains

Image Object Level; Image Object List

13.3.2 Algorithm Parameters

Thematic Layer

Select the thematic layer for the algorithm.

Thematic Layer Attribute

Choose attributes from the thematic layer for the algorithm. You can select any numeric attribute from the attribute table of the selected thematic layer.

Variable

Select an existing variable or enter a new name to add a new one. If you have not already created a variable, the Create Variable dialog box will open.

13.4 Write Thematic Attributes

Generate a attribute column entry from an image object feature. The updated attribute table can be saved to a .shp file.

13.4.1 Supported Domains

Image Object Level

13.4.2 Algorithm Parameters

Thematic Layer

Select the thematic layers for the algorithm.

Feature

Select the feature for the algorithm.

Save Changes to File

If the thematic layer is linked with a shapefile, the changes can be updated to the file.

14 Workspace Automation Algorithms

Workspace automation algorithms are used for working with subroutines of rule sets. These algorithms enable you to automate and accelerate the processing of workspaces with particularly large images. Using workspace automation algorithms you can create multi-scale workflows, which integrate analysis of images at different scales, magnifications, or resolutions.

14.1 Create Scene Copy

Create a scene copy that is a duplicate of a project with image layers and thematic layers, but without any results such as image objects, classes, or variables. This algorithm enables you to use subroutines.

14.1.1 Supported Domains

Execute; Maps

14.1.2 Algorithm Parameters

Scene Name

Edit the name of the scene copy to be created.

Scale

See *Scale* (p 124) for an explanation of the Select Scale dialog box.

Additional Thematic Layers

Edit the thematic layers you wish to load to a scene copy. This option is used to load intermediate result information that has been generated within a previous subroutine and exported to a geocoded thematic layer. Use semicolons to separate multiple thematic layers, for example, `ThematicLayer1.tif;ThematicLayer2.tif`.

14.2 Create Scene Subset

Copy a portion (subset) of a scene as a project with a subset of image layers and thematic layers. Image objects, classes and variables are not copied. The algorithm uses the given co-ordinates (geocoding or pixel co-ordinates) of the source scene. You can create subset copies of an existing subset.

14.2.1 Supported Domains

Execute; Image Object Level

14.2.2 Algorithm Parameters

Scene Name

Edit the name of the scene copy to be created.

Use Variable as Scale

Specify the scale for copying a map using a variable, rather than defining a numerical value.

Scale

See *Scale* (p 124) for an explanation of the Select Scale dialog box.

Border Size

Extends the bounding box of the image object by the entered border size in pixels. (Only available when image object level is selected as the image object domain.)

Include Maps With Objects Linked Via

Class filter for object link class that links objects on different maps. the subset will contain all maps where the current object has linked objects on. (Only available when image object level is selected as the image object domain.)

Exclude Other Image Objects

Define a no-data area outside of the current image object to exclude other image objects from further processing. (Only available when image object level is selected as the image object domain.) Selecting 'yes' brings up two further parameters:

- Customize Path – if set to 'yes', the Export Path parameter is activated

- In Export Path, enter the file attributes and filepath in the form `{:Workspc.Output.Root}\:Scene.Name}_{:Variable:MyVariable}.ext` or simply `C:\MyFile.tif`

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace.

Minimum and Maximum X, Y, Z & T Co-ordinates

Edit the co-ordinates of the subset. For the default co-ordinates orientation (0,0) in the bottom left-hand corner, the different co-ordinates are defined as follows:

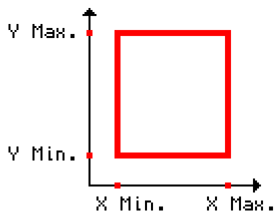


Figure 14.1. Co-ordinates of a subset

- The minimum X co-ordinates describe the left border
- The maximum X co-ordinates describe the right border
- The minimum Y co-ordinates describe the lower border
- The maximum Y co-ordinates describe the upper border.

Alternatively, click the drop-down arrow button to select available variables. Entering a letter will open the Create Variable dialog box.

Co-ordinates Orientation

You can change the corner of the subset that is used as the calculation base for the co-ordinates. The default is (0,0) in the lower-left corner.

Additional Thematic Layers

Edit the thematic layers you wish to load to a scene copy. This option is used to load intermediate result information that has been generated within a previous subroutine and exported to a geocoded thematic layer. Use semicolons to separate multiple thematic layers, for example, ThematicLayer1.tif;ThematicLayer2.tif.

14.3 Create Scene Tiles

Create a tiled copy of a scene. Each tile is a separate project with its own image layers and thematic layers. Together the tile projects represent the complete scene as it was before creating the tiled copy. The given co-ordinates (geocoding or pixel co-ordinates) of the source scene of the rule set are used. Results are not included before the tiles are processed.

After processing, you can stitch the tile results together and add them to the complete scene within the dimensions as it was before creating the tiled copy. You can tile scenes and subsets several times.

14.3.1 Supported Domains

Execute

14.3.2 Tile Size Parameters

Tile Height

Edit the height of the tiles to be created. Minimum height is 100 pixels.

Tile Width

Edit the width of the tiles to be created. Minimum with is 100 pixels.

14.4 Submit Scenes for Analysis

This algorithm enables you to connect subroutines with any process of the main process tree or other subroutines. You also can also choose whether to stitch the results of the analysis of subset copies.

A rule set that contains the submit scenes for analysis algorithm can only be executed if you are connected to Definiens Server. Rule sets that include subroutines cannot be processed on a local machine.

14.4.1 Supported Domains

Execute

14.4.2 Algorithm Parameters

Type of Scenes

Select the type of scene to submit to analysis – the top level scenes, tiles, or subsets and copies.

If you select top-level scenes, they can only be submitted on the client, and not on a server. This option is designed for use with actions that are used in the Analysis Builder window and that will not be analyzed on the server.

State Filter

This field only displays for top-level scenes. The default is any state. use the drop-down list to select a processing state: created, edited, processed, failed, canceled or rejected.

Submit Recursively

Submits scenes recursively.

Scene Name Prefix

Enter the prefix of the names of scene copies to be selected for submitting. A prefix is defined as the complete scene name or the beginning of it. Enter the unique part of the name to select only that scene, or the beginning of the name to select a group with similar or sequential names. For example, if you have scene names 7a, 7b and 7c, you can select them all by entering a 7, or select one by entering 7a, 7b or 7c.

Process Name

Address a subroutine or a process in the process tree of a subroutine for execution by using a forward slash before hierarchy steps, for example, subroutine/process name. You can use the context menu in the Process Tree window to copy a process and paste it into this field.

Parameter Set for Process

Select a parameter set to transfer variables to the following subroutines. Click the ellipsis button to open the Select Parameter Set dialog box.

Percent of Tiles to Submit

If you do not want to submit all tiles for processing but only a certain percentage, you can edit the percentage of tiles to be processed. If you change the default value of 100, the tiles are selected randomly. If the calculated number of tiles to be submitted is not an integer it is rounded up to the next integer.

If the value entered is less than or equal to 0, 1.0 will be used. If the value entered is greater than 100, 100 will be used. Tiles that are not selected are automatically assigned the status “skipped”.

14.4.3 Stitching Parameters

Stitch Subscenes

Select Yes to stitch the results of subscenes together and add them to the complete scene within its original dimensions. Only the main map of a tile projects can be stitched.

Overlap Handling

If Subsets and Copies are stitched, the overlapping must be managed. You can opt to create Intersection image objects (default) or select Union to merge the overlapping image objects.

Class for Overlap Conflict

Overlapping image objects may have different classifications. In that case, you can define a class to be assigned to the image objects resulting from overlap handling.

14.4.4 Post-Processing Parameters

Request Post-Processing

Select Yes to execute another process after subscenes are processed.

Post-Process Name

Address a subroutine or a process in the process tree of a subroutine for execution by using a forward slash before hierarchy steps, for example, subroutine/process name. You can use the context menu in the Process Tree window to copy a process and paste it into this field.

Parameter Set for Post-Process

Select a parameter set to transfer variables to the following subroutines. Click the ellipsis button to open the Select Parameter Set dialog box.

14.5 Delete Scenes

Delete the scenes you no longer want to use or store.

14.5.1 Supported Domains

Execute

14.5.2 Algorithm Parameters

Type of Subscenes

Select the type of scene copy to be deleted: tiles or subsets and copies.

Scene Name Prefix

Enter the prefix of the names of scene copies to be selected for deleting. A prefix is defined as the complete scene name or the beginning of it. Enter the unique part of the name to select only that scene, or the beginning of the name to select a group with similar or sequential names. For example, if you have scene names 7a, 7b and 7c, you can select them all by entering a 7, or select one by entering 7a, 7b or 7c.

Delete Results

If No is selected, the subset or tile is removed from the Workspace, along with the .dpr file and the shapefile. If Yes is selected (the default), all exported results will additionally be deleted.

2.0.3 Compatibility Mode

Selecting Yes activates backward compatibility for Developer XD 2.0.3

14.6 Read Subscene Statistics

Read exported result statistics and perform a defined mathematical summary operation. The resulting value is stored as a variable that can be used for further calculations or export operations concerning the main scene. This algorithm summarizes all values in the selected column of a selected export item, using the selected summary type.

When the analysis of subscenes results in exporting statistics for each scene, the algorithm allows you to collect and merge the statistical results of multiple files. The advantage is that you do not need to stitch the subscenes results for result operations concerning the main scene. The following preconditions apply:

- For each subscene analysis, a project or domain statistic has already been exported
- All preceding subscene analysis including export has been processed completely before the read subscene statistics algorithm starts any result summary calculations. To ensure this, result calculations are done within a separate subroutine.

14.6.1 Supported Domains

Execute

14.6.2 Algorithm Parameters

Use Variable as Scene Type

This function lets you use the text of a variable containing a scene type, rather than the scene type itself. Selecting 'yes' activates the Scene Type Variable parameter.

Scene Type Variable

Create or select a scene type variable.

Type of Scene

Select the type of scene to be submitted (this parameter only appears if Use Variable as Scene Type is set to 'no'):

- Current scene
- Subsets and copies
- Subset and copies recursively
- Tiles
- Tiles recursively
- Tiles from subset.

Scene Name Prefix

Enter the prefix of the names of the scenes to be selected for reading. A prefix is defined as the complete or the beginning of the scene name. Enter the unique part of the name to select only that scene, or the beginning of the name to select a group with similar or sequential names. For example, if you have scene names 7a, 7b and 7c, you can select them all by entering a 7, or select one individually by entering 7a, 7b or 7c.

Export Item

Enter the name of the export item statistics to be summarized (for example `ProjectStatistics` or `ObjectStatistics`).

Output Type

This value specifies the type of statistical output:

- Single Column reads single column values and writes output to the specified variable
- All Columns reads all available columns, creates a variable for each one, then puts these variables into a specified feature list

Column

If ‘single column’ is selected as the Output Type, specify which column should be used by the algorithm for the statistical summary operation.

Feature List

The Feature List parameter appears when ‘all columns’ is selected as the output type. Enter the feature list variable to receive the columns.

Add Summary Prefix

The Add Summary Prefix option appears when the Output Type parameter is set to ‘all columns’. It adds the summary type as a prefix to each variable’s name.

14.6.3 Mathematical Parameters

The following summary operations are available:

Value	Description
Count	Calculate the number of items
Sum	Sum all values of appropriate statistics table columns
Mean	Calculate the average of all values
Std. Dev.	Calculate the standard deviation of all values
Min	Calculate the minimal value of all values
Max	Calculate the maximal value of all values

The following parameters are common to all the mathematical parameters:

Use Variable for Usage Flag

If ‘yes’ is selected, a variable value is used to calculate a summary; if set to ‘no’, a fixed value is used

Calculate Summary

Select ‘yes’ to calculate the summary.

Result Variable

Enter a variable to store the resulting value. (This parameter is only visible if Single Column is selected and Calculate Summary is set to ‘yes’.)

15 Interactive Operation Algorithms

Interactive operation algorithms are used to provide user interaction with the user of actions in Architect XD.

15.1 Show User Warning

Edit and display a user warning.

15.1.1 Supported Domains

Execute

15.1.2 Algorithm Parameters

Message

Edit the text of the user warning.

15.2 Delete a File

Delete any file or folder in a Workspace directory.

15.2.1 Supported Domains

Execute

15.2.2 Algorithm Parameters

File Name

Define the filename and the path of the parameter set files to be loaded.

You can use the suggested file name pattern `{:Workspc.OutputRoot\}parameter_sets\paramset.psf`. It defines the folder `parameter_sets` located in the workspace output root folder as displayed in the Workspace Properties dialog box. The parameter set files are named `paramset`, followed by the filename ending as described below and the file extension `.psf`.

Click the drop-down arrow to select text elements for editing the filename pattern.

File Name Ending

Define the filename ending to add to the File Name field. Click the drop-down arrow to select an available variable. Alternatively, you can insert text between the quotation marks.

15.3 Set Active Pixel

Sets the active pixel to the given co-ordinate.

15.3.1 Supported Domains

Execute

15.3.2 Algorithm Parameters

X Co-ordinate/Y Co-ordinate

There are two ways to set the value:

1. Assign the value by entering a numeric value, or enter a non-numeric value to create a variable
2. Click the drop-down list to select a feature.

15.4 Create/Modify Project

Create a new project or modify an existing one.

15.4.1 Supported Domains

Execute

15.4.2 Image Layer Parameters

Image File

Browse for an image file containing the image layers. Alternatively you can edit the path.

Image Layer ID

Change the image layer ID within the file. Note, that the ID is zero-based.

Image Layer Alias

Edit the image layer alias.

15.4.3 Thematic Layer Parameters

Thematic File

Browse for a thematic file containing the thematic layers. Alternatively, you can edit the path.

Attribute Table File

Browse for an attribute file containing thematic layer attributes. Alternatively, you can edit the path.

Attribute ID Column Name

Edit the name of the column of the attribute table containing the thematic layer attributes of interest.

Thematic Layer Alias

Edit the thematic layer analysis.

15.4.4 General Settings Parameters

Show Subset Selection

Opens the Subset Selection dialog box when executed interactively.

Enable Geocoding

Activate to select the bounding co-ordinates based on the respective geographical co-ordinate system.

15.5 Manual Classification

This algorithm allows the user of an action to classify image objects of the selected class manually by clicking on them.

15.5.1 Supported Domains

Execute

15.5.2 Algorithm Parameters

Class

Select a class that can be assigned manually.

Use Brush

The Use Brush parameter, when assigned the value Yes, activates a brush tool. Holding down the left mouse button allows a user to manually classify objects by dragging the mouse over them. If this action is performed while pressing the Shift key, objects are unclassified.

Brush Size

Define the brush size used for classification.

15.6 Configure Object Table

Display a list of all image objects together with selected feature values in the Image Object Table window.

15.6.1 Supported Domains

Execute

15.6.2 Algorithm Parameters

Classes

Select classes to list all of its image objects.

Features

Select the features to display the feature values of the image objects.

15.7 Select Input Mode

Set the mode for user input via graphical user interface.

15.7.1 Supported Domains

Execute

15.7.2 Algorithm Parameters

Input Mode

Select an input mode:

- If Normal is selected, normal input mode is returned to (for example, selection of image objects by clicking them)
- Manual Object Cut activates the Cut Objects Manually function.

15.8 Start Thematic Edit Mode

The Start Thematic Edit Mode algorithm activates thematic editing, creation of thematic layers and drawing functions. This mode can call on *any* map, not just the active one. The following actions are available after execution:

- Click and hold the left mouse button as you drag the cursor across the image to create a path with points in the image
- To create points at closer intervals, drag the cursor more slowly or hold the Ctrl key while dragging.
- Release the mouse button to automatically close the polygon
- Click along a path in the image to create points at each click. To close the polygon, double-click or select Close Polygon in the context menu
- To delete the last point before the polygon is complete, select Delete Last Point in the context menu.

15.8.1 Supported Domains

Execute

15.8.2 Algorithm Parameters

Mode

Select Full Editing or Classify Only. The Classify Only option enables the editing of class and layer name parameters only.

Class

Selct an annotation class.

Use Magnetic Snap

When this function is activated and the user draws a line around an image object with the mouse, the line will automatically snap to the area of highest contrast.

Magnetic Snap Radius

Insert a value to specify a detection radius; the magnetic snap detects edges only within the specified distance. A value of zero means no magnetic snap.

Thematic Layer Name

The thematic layer name for manual annotation.

15.9 Select Thematic Objects

Use the Select Thematic Objects algorithm to enable selection of thematic objects in the user interface. The algorithm activates thematic editing and enables cursor selection mode. It is designed to be used with actions.

15.9.1 Algorithm Parameters

Layer Name

Enter the name of the layer where thematic objects are to be selected.

Selection Mode

Choose the type of selection:

- Single: enables selection of single polygons.
- Polygon: enables selection of all shapes within a user-drawn polygon.
- Line: enables selection of all shapes crossed by a user-drawn line.
- Rectangle: enables selection of all shapes within a user-drawn rectangle.

15.10 Finish Thematic Edit Mode

Use the Finish Thematic Edit Mode algorithm to switch back from thematic editing to image object editing and save the shapefile. It is designed to be used with actions and can be used on multi-scene projects.

15.10.1 Supported Domains

Execute

15.10.2 Algorithm Parameters

Save Changes

Selct Yes to save manual editing changes into the file.

File Name

Define the file name of the shapes file.

15.11 Select Image Object

Select an image object designated in the image object domain in the active view. The process has the same effect as if the user manually selects the image object with the mouse. You can use this algorithm in Architect solutions if you want to highlight an image object or automatically display its properties in the Image Object Information window. Only the first image object in the selected domain will be selected.

15.11.1 Supported Domains

Image Object Level; Current Image Object; Super Object; Sub Objects; Linked Objects

15.11.2 Algorithm Parameters

None

15.12 Polygon Cut

The Polygon Cut algorithm allows users to manually cut image objects. When activated, the client will enter polygon input mode, which allows line or polygon editing as specified by the rule set writer. Line and polygon mode are identical except in polygon mode, the user can close a shape by double-clicking or releasing the mouse button near the start point. New objects can be classified according to classification parameters and there is an option to store them in an object list variable.

When creating a polygon, displaying the context menu by right-clicking offers the following options for a open polygon:

- Cancel Split deletes the entire polygon
- Close and Split joins the first and last points together
- Delete Last Point removes the last point created

The following restrictions apply:

- In polygon mode, if the area of the bounding box exceeds 2,048 x 2,048, a warning will appear that the process will take some time to excecute and prompts users if they want to continue
- In polygon mode, if the area of the bounding box exceeds 10,000 x 10,000, a warning is shown that this value cannot be exceeded and the cutting action is not executed
- In 3D and 4D projects, the cutting is always performed within a single *xy* slice at a single time point.

15.12.1 Supported Domain

Execute

15.12.2 Algorithm Parameters

Mode

- Polygon Cut – in polygon mode, a closed polygon is drawn and the area inside it is cut
- Line Cut – in line mode, a line is drawn and rasterized by creating pixel width image objects along the line

Class

Specify the class of objects to be annotated.

Use Magnetic Snap

Select ‘yes’ or ‘no’. If ‘yes’ is selected, the drawn line will automatically snap to high-contrast areas, when the user drags the mouse while holding down the left mouse button

Magnetic Snap Radius

Insert a value to specify a detection radius; the magnetic snap detects edges only within the specified distance. A value of zero means no magnetic snap. The default value is 30.

Class Filter

Specify the class of objects to be cut.

Image Object List

If selected, the image object list variable that will receive the cut image objects.

Callback Process

The path to the process invoked after each cut operation.

Preserve Object Type for Cut Objects

Select ‘yes’ or ‘no’. If ‘yes’ is selected, the image object type will be preserved for cut image objects. If ‘no’ is chosen, cut image objects will be marked as disconnected (this option is less processor intensive).

Ensure Connected for Objects Created by the Cut

Select ‘yes’ or ‘no’. If ‘yes’ is selected, the resulting image objects will be converted to connected 2D. If ‘no’ is chosen, resulting image objects will be marked as disconnected.

15.13 Save/Restore View Settings

Save and restore view settings of the active view. Use this algorithm to configure the current view to a predefined setting used in Architect XD solutions. This is achieved in two steps: first the current view settings are stored using the Save View Settings mode. Then the active view can be restored to the saved view settings by invoking it in the Restore View Settings mode.

15.13.1 Supported Domain

Execute

15.13.2 Algorithm Parameters

Mode

- Save view settings – in this mode the view settings can be saved. They will be stored together with the algorithm. Use the View Settings parameter to save the current view.
- Restore view settings – in this mode the view settings are restored to the state represented by another process using this algorithm in the Save View Settings mode. Use the Process Path parameter to specify the process that stores the view settings.

View Settings

This parameter is only available in the Save View Settings mode. Click the ellipsis in the Value column (which displays Click to Capture View Settings) to store the current view settings. In addition to main maps, the view settings for maps and window splits are also saved.

Process Path

This parameter is only available in the Restore View Settings mode. Refer to another process using this algorithm in Save View Settings mode to restore the view settings that are stored in this algorithm. To refer to a process, right-click on it and select Go To to open the Go To Process dialog box.

15.14 Display Map

This algorithm displays a specific map in the currently active view. It can be used to configure the correct view for the user in interactive Architect solutions that use multiple maps.

15.14.1 Supported Domains

Execute

15.14.2 Algorithm Parameters

Map Name

Select the name of the map to display in the current active view.

15.15 Define View Layout

In Architect XD you can define a particular window layout for end users. The Define View Layout algorithm allows you to display windows split into one, two or four with a different map displayed in each pane or a 3D projection.

The following view properties apply:

- If a map is not available, the main map is displayed
- When changing layouts using the same map, the zoom properties of the active window are used
- When changing maps, the image is adjusted to the size of the window
- Any layout settings, such as numbers of layers or outlines, are reset when changing layouts.

15.15.1 Supported Domains

Execute

15.15.2 Algorithm Parameters

Split Type

- No Split
- Split Vertically
- Split Horizontally
- 4 Panes
- Comparison View for a 2D project, two vertical panes; for a 3D project, comparisons of XY, XZ or YZ views

View Type

- Independent View
- Side by Side View
- Swipe View

Synchronize Views

Define how image views are laid out and what content is displayed, for each image view. Select yes to synchronize all the map views with each other.

15.15.3 Pane Parameters

Map Name

Select the name of the map to display in the current active view.

Projection

XY, XZ, YZ or 3D

15.16 Set Custom View Settings

Apply custom view settings to the active view.

15.16.1 Supported Domains

Execute

15.16.2 Algorithm Parameters

Classification

Select which classification display settings should be applied. Values are On or Off.

Outlines

Select which outline display settings should be applied; available values are None, Opaque and Transparent.

Thematic Objects

Display vector layer polygons in image view, without entering particular editing mode. Values are On or Off.

15.17 Change Visible Layers

Many algorithms use temporary layers for computation purposes; however, this may be confusing for people who are running solutions in Architect XD and Tissue Studio™. The Change Visible Layers algorithm allows rule-set developers to specify the visibility of new image layers. Hidden layers are hidden by default in Definiens Developer XD 2.0.4, although it is possible to turn this option on and off in Tools > Options (for more details consult the user guide).

15.17.1 Supported Domains

Execute

15.17.2 Algorithm Parameters

Apply to

Select whether to apply the action to image layers or thematic layers.

Mode

The following options are available:

- Show single layer
- Hide single layer
- Show all layers
- Hide all layers

Image Layer/Thematic Layer

Select the image layer or thematic layer (depending on your selection in the Apply To field) to show or hide.

15.18 Change Visible Map

The Change Visible Map algorithm lets you show or hide maps in image view. You can define which maps can be displayed.

15.18.1 Supported Domains

Execute

15.18.2 Algorithm Parameters

Show Map

Choose from the following values:

- Main
- From Parent
- Active Map
- <Create New Variable>

Hide Map

Values for the Hide Map parameter are the same as those for Show Map.

Allow Empty Image View

When Allow Empty Image View has a value of "yes":

- If Show Map has no value and a map is selected in the Hide Map field, this map is removed and a black image view is displayed
- If a map is selected in the Show Map field but Hide Map has no value, all black image views will display the selected map

15.19 Show Scene

The Show Scene algorithm allows the user to navigate through the slide hierarchy. This facility only functions in the context of workspaces.

15.19.1 Algorithm Parameters

Show Scene

Select from the following:

- First child
- First slide
- Next sibling
- Next slide
- Previous sibling
- Previous slide
- Parent

15.20 Ask Question

15.20.1 Supported Domains

Execute

15.20.2 Algorithm Parameters

Message

Amend this value to change the title of the dialog box.

Result

Insert a value that will store the result. If the user selects ‘yes’, the value is 1; if ‘no’ is selected, the value is 0.

15.21 Set Project State

The standard configuration of Definiens Developer XD 2.0.4 is to prompt the user to save a project upon closing, even if the user has not made any changes to the file. The Set Project State algorithm allows rule-set writers to disable this prompt.

15.21.1 Supported Domains

Execute

15.21.2 Algorithm Parameters

Mark Project As

Set this value to modified or unmodified. If modified is selected, the user will be prompted to save; if unmodified is selected, this option will not appear.

15.22 Save/Restore Project State

This algorithm lets you save or restore the state of all maps, levels and image projects as part of a rule set.

15.22.1 Supported Domains

Execute

15.22.2 Algorithm Parameters

Mode

Choose either ‘save state’ or ‘restore’:

- In save mode, the algorithm saves all maps to a temporary project file
- in restore mode, the algorithm restores the previously saved state from the project file.

Name of the State to be Saved or Restored

Enter a name in quotes, or use the drop-down list to select a variable, feature or array item.

15.23 Show HTML Help

The Show HTML Help algorithm allows developers to incorporate an HTML file into a rule set. This can be displayed as a pop-up window that launches in response to a user action. This is useful for adding additional help information to a function.

The help dialog contains a checkbox with the label ‘do not show again’, which disables the function for a period of two weeks.

15.23.1 Supported Domains

Execute

15.23.2 Algorithm Parameters

Help File Name

Define the name of the help file. The HTML file must be created independently and conform to the defined address.

Shapes File

Enter the name of the shape file

15.24 Configure Manual Image Equalization

The Configure Manual Image Equalization algorithm sets the manual image layer equalization display for active or all image views.

15.24.1 Supported Domains

Execute

15.24.2 Algorithm Parameters

Apply to all Image Views

Select 'yes' to apply settings to all image views. If 'no' is selected, settings will be applied to the active image view only.

Use Variable for Equalization

If 'yes' is selected, the software will expect a scene variable of the string type as an equalization parameter. If 'no' is selected, equalization must be specified explicitly.

Equalization

Select one of the following equalization modes:

- Linear
- Linear – Inverse
- Gamma Correction (Positive)
- Gamma Correction (Negative)
- Gamma Correction (Positive) – Inverse
- Gamma Correction (Negative) – Inverse

Range Type

There are two ways to define the range – Min/Max or Center/Width. The value range will not exceed the limitations of the channel; for example, the value for an 8-bit channel will always be between 0 and 255.

$$\min = \text{center} - \frac{\text{width}}{2}, \max = \text{center} + \frac{\text{width}}{2}$$

Min/Max Specify the minimum and maximum value for the layer equalization range.

Center/Width Specify the center and width values for the layer equalization range.

Apply to all Visible Layers

Select 'yes' to apply manual equalization to all visible image layers.

Interactive Range Editing Step

The value of the Interactive Range Editing Step represents the step-size in pixels, for the interactive mouse action.

15.25 Edit Aperio Annotation Links

This algorithm launches the Aperio annotation Link Manager dialog, where users can define annotation mapping. Two options are available in the Link Manager:

- Link by name: Aperio annotations are linked by name (defined layer name) with Definiens ROI.
- Link by position: Aperio annotations are linked by position (1st Aperio–last Aperio layer position) with Definiens ROI.

Pressing OK executes the child process Apply Aperio Annotation Links.

This function requires a connection to a Aperio Spectrum database. For more information on configuration options and the Link Manager, please consult the Aperio chapter of the *Tissue Studio User Guide*.

15.25.1 Supported Domains

Execute

15.25.2 General Parameters

Threshold Condition

Define an additional threshold condition to define the candidate image object domain.

Map

Choose one of the following values:

- Main
- From Parent
- Active Map
- <Create New Variable>

15.25.3 Algorithm Parameters

Variable Holding Links

AperioLinkStr

Definiens ROIs

Select the available Developer ROI classes to be linked to Aperio annotation layers.

15.26 Apply Aperio Annotation Links

This algorithm creates thematic layer, applies the mapping defined and imports any existing Aperio annotation links (from an Aperio Spectrum database) defined in the parent process.

This function requires a database connection for more information on configuration options, please consult the Aperio chapter of the Tissue Studio user guide.

15.26.1 Supported Domains

Execute

15.26.2 General Parameters

Threshold Condition

Add an additional threshold condition to define the candidate image object domain.

Map

Choose one of the following values:

- Main
- From Parent
- Active Map
- <Create New Variable>

15.26.3 Algorithm Parameters

Variable Containing Links

AperioLinkStr

Thematic Layer Name

Enter a name for the thematic layer to be used for manual annotation.

16 Parameter Set Operations Algorithms

Parameter set operations algorithms enable the automated exchange of parameter sets among actions. Parameter sets are especially important in rule sets using workspace automation and rule sets created for actions.

16.1 Apply Parameter Set

Writes the values stored inside a parameter set into the related variables. For each parameter in the parameter set, the algorithm scans for a variable with the same name. If this variable exists, then the value of the variable is updated by the value specified in the parameter set.

16.1.1 Supported Domains

Execute

16.1.2 Parameter Set Name

Select the name of a parameter set.

16.2 Update Parameter Set

Writes the values of variables into a parameter set. For each parameter in the parameter set, the algorithm scans for a variable with the same name. If this variable exists, then the value of the variable is written to the parameter set.

16.2.1 Supported Domains

Execute

16.2.2 Parameter Set Name

Select the name of a parameter set.

> **TIP**: Parameters are created with the Manage Parameter Sets dialog box, which is available on the menu bar under Process or on the toolbar.

16.3 Load Parameter Set

Load a parameter set from file. Use this algorithm to exchange parameter sets among actions.

16.3.1 Supported Domains

Execute

16.3.2 Algorithm Parameters

File Name

Define the file name and the path of the parameter set files to be loaded. You may use the same settings as used for a related Save Parameter Set action.

You can use the suggested file name pattern `{:Workspc.OutputRoot\}parameter_sets\paramset.psf`. It defines the folder `parameter_sets` located in the workspace output root folder as displayed in the Workspace Properties dialog box. The parameter set files are named `paramset`, followed by the file name ending as described below and the file extension `.psf`.

Click the drop-down arrow to select text elements for editing the file name pattern.

File Name Ending

Define the file name ending to add to the File Name field. You may use the same settings as used for a related Save Parameter Set action. Click the drop-down arrow to select an available variable. Alternatively, you can insert text between the quotation marks.

16.4 Save Parameter Set

Save a parameter set as a file. Use this algorithm to exchange parameter sets among actions.

16.4.1 Supported Domains

Execute

16.4.2 Algorithm Parameters

Parameter Set Name

Click the drop-down arrow to select a parameter set to save to file. Alternatively, you can enter a given parameter set name.

File Name

See the explanation contained in *Load Parameter Set* on the facing page.

File Name Ending

See the explanation contained in *Load Parameter Set* on the preceding page.

16.5 Update Action from Parameter Set

Synchronize the values of an action according to the values of a parameter set.

16.5.1 Supported Domain

Execute

16.5.2 Algorithm Parameters

Action Name

Type the name of an action.

Parameter Set Name

Select the name of a parameter set.

16.6 Update Parameter Set from Action

Synchronize the values of a parameter set according to the values of an action.

16.6.1 Supported Domains

Execute

16.6.2 Algorithm Parameters

Action Name

Type the name of an action.

Parameter Set Name

Select the name of a parameter set.

16.7 Apply Active Action to Variables

Apply Active Action to Variables updates variables that have been set by action widgets.

16.7.1 Supported Domains

Execute

16.7.2 Algorithm Parameters

None

17 Sample Operation Algorithms

Use sample operation algorithms to handle samples for Nearest Neighbor classification and to configure the Nearest Neighbor settings.

17.1 Classified Image Objects to Samples

Create a sample for each classified image object in the image object domain.

17.1.1 Supported Domains

Image Object Settings; Image Object List

17.1.2 Algorithm Parameters

None

17.2 Cleanup Redundant Samples

Remove all samples with membership values higher than the membership threshold.[1]

17.2.1 Supported Domains

Image Object Level

17.2.2 Algorithm Parameters

Membership Threshold

You can modify the default value of 0.9.

1. This algorithm might produce different results each time it is executed, because the order of sample deletion is random.

17.3 Nearest Neighbor Configuration

Select classes, features and function slopes to use for nearest neighbor classification.

17.3.1 Supported Domains

Execute

17.3.2 Algorithm Parameters

Active Classes Choose the classes you wish to use for nearest neighbor classification.

NN Feature Space

Select as many features as you like for the nearest neighbor feature space.

Function Slope

Enter the function slope for the nearest neighbor.

17.4 Delete All Samples

Delete all samples. This algorithm has no parameters.

17.4.1 Supported Domains

Execute

17.5 Delete Samples of Classes

Delete all samples of certain classes.

17.5.1 Supported Domains

Execute

17.5.2 Algorithm Parameters

Class List

Select the classes for which samples are to be deleted.

17.6 Disconnect All Samples

Disconnect samples from image objects to enable creation of samples that are not lost when image objects are deleted. Samples are stored in the solution file. This algorithm has no parameters.

17.6.1 Supported Domains

Execute

17.7 Sample Selection

Use the sample selection algorithm to switch the cursor to sample selection mode using the selected class.

17.7.1 Supported Domains

Execute

17.7.2 Algorithm Parameters

Class

Choose a class to use in selecting samples.

Use Brush

The Use Brush parameter, when assigned the value Yes, activates a brush tool. Holding down the left mouse button allows a user to manually classify objects by dragging the mouse over them. If this action is performed while pressing the Shift key, objects are unclassified.

18 Export Algorithms

Export algorithms are used to export table data, vector data and images derived from the image analysis results.[1]

18.1 Export Classification View

18.1.1 Supported Domains

Image Object Level

18.1.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Export Unclassified as Transparent

Select Yes to export unclassified image objects as transparent pixels.

1. In Windows there is a 260-character limit on filenames and filepaths (http://msdn.microsoft.com/en-us/library/windows/desktop/aa365247%28v=vs.85%29.aspx). Definiens software does not have this restriction and can export paths and create workspaces beyond this limitation. For examples of this feature, refer to the FAQs in the Windows installation guide.

Enable Geo Information

Activate to add geographic information.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will be stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace.

18.2 Export Current View

Export the current map view to a raster file.

18.2.1 Supported Domains

Execute; Maps

18.2.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Enable Geo Information

Activate to add geo information.

Save Current View Settings

Click the ellipsis button to capture current view settings. Transparency settings may affect the appearance of the exported view.[2]

Scale

See *Scale* (p 124) for an explanation of the Select Scale dialog box.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will be stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

Export Multiple Slices As

Specify how the slices of a three-dimensional scene are exported.[3] You can export the slices into a single montage image, into multiple files or into a multi-page .tif file. To export only a selection of slices, you can additionally change the settings under Slices and Frames:

- Select Current to export the current slice or slices of the current time frame.
- Select Single to export a certain slice or slices of a certain time frame. Indicate the slice or time frame using the slice and frame index respectively.
- Select Range to export a range of slices or slices of several time frames. Indicate the range using the slice and time frame index respectively.
  The settings can be made independently. By default, both values are set to all.

2. Projects created with versions of Definiens Developer XD prior to 1.1 will display with the current transparency settings. If you want to use the Export Current View algorithm and preserve the current transparency settings, access the Algorithm parameters. Then select Click to Capture Current View Settings in the Save Current View settings field. (If you want to preserve the original transparency settings, do not select Click to Capture Current View Settings.)
3. The Export Current View algorithm only exports the slices that lie within the selected range or time frame. That is, if you select a range or time frame that does not contain any slices for the current map, no image file is written.

18.2.3 Slices Parameters

Slice Selection

Defines which Z slices are exported.

Slice Index

Selected slice index.

First Slice Index

First slice index.

Last Slice Index

Last slice index.

18.2.4 Frames Parameters

Frame Selection

Defines which time frames are exported.

Frame Index

Selected time frame index.

First Frame Index

First time frame index.

Last Frame Index

Last time frame index.

18.3 Export Thematic Raster Files

Export thematic raster files.

18.3.1 Supported Domains

Image Object Level

18.3.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Export Type

Select the type of export:

- Select Image Objects to export feature values
- Select Classification to export classification by unique numbers associated with classes.

Features

Select one or multiple features to export their values.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will be stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

Geo-Coding Shift X

Shift the geocoding lower-left corner in the X direction. Use the drop-down list to shift half a pixel to the left or right.

Geo-Coding Shift Y

Shift the geocoding lower-left corner in the Y direction. Use the drop-down list to shift half a pixel up or down.

18.4 Export Domain Statistics

Select an image object domain and export statistics regarding selected features to a file.

18.4.1 Supported Domains

Image Object Level; Image Object List

18.4.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Domain Features

Select the features whose statistical calculations you want to be exported.

Project Features

Select any features not subject to statistical calculations that you wish to export.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace.

File Name Suffix

File Name Suffix allows users to select features or variables or enter a string. The value of this parameter is then added to the name of the exported file.

When you define a suffix, be aware that certain characters in Windows are invalid in filenames – invalid filename characters will result in Windows error code 123.

Feature Header Suffix

Use Feature Header Suffix to insert text and variables (using square brackets) into the column name of an exported table. Double and string values can be used.

18.4.3 Statistical Operations

Select statistical operators with a Yes or No from the drop-down arrow. The following six operators are available: Number, Sum, Mean, Std. Dev, Min and Max. Users can also select – in addition to Yes or No – a scene variable:

- 0 = No
- <>0 = Yes

18.5 Export Project Statistics

Export values of selected project features to a file.

18.5.1 Supported Domains

Execute

18.5.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)

- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Features

Select one or multiple features for exporting their values.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

18.6 Export Object Statistics

Export statistics of selected features per image object. Per project, one file is generated. In spreadsheet files the following table columns are included:

- inner_x: x co-ordinate of the inner pixel[4]
- inner_y: y co-ordinate of the inner pixel
- level_name: The image object level of the image object
- class_name: The class of the image object
- An additional column per selected feature.

18.6.1 Supported Domains

Image Object Level; Image Object List

4. The inner pixel is a pixel located within the image object serving as image object reference. The calculation of inner pixel co-ordinates has changed with Definiens Developer EII version 8 and co-ordinate values of inner pixel exported with the latest version are different to those exported with earlier versions.

18.6.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Customize Path

Selecting Yes allows you to customized the default export path.

Export Series

If set to Yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Features

Select one or multiple features for export.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

File Name Suffix

File Name Suffix allows users to select features or variables or enter a string. The value of this parameter is then added to the name of the exported file.

18.6.3 Report Parameters

Create Summary Report

Select Yes to create a summary report after processing is complete.

Creation Mode

- Select Recreate to generate a new file; this report is based on the current set of source files. The source files are sorted before the report is created and will always be in the same order
- Selecting Append Only New Data will simply update the file, adding results to the end of the report. Processing the same project several times will add data to the report without deleting existing data.

Remove Auxillary Columns

Choose whether reports should contain auxillary columns.

Split File By

This feature allows you to create a separate export file based on one or more features.

18.7 Export Object Statistics for Report

Export image object statistics to a file. This generates one file per workspace.

18.7.1 Supported Domains

Image Object Level; Image Object List

18.7.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Features

Select one or multiple features for exporting their values.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

18.7.3 Report Parameters

Create Summary Report

Select Yes to create a summary report after processing is complete.

Creation Mode

- Select Recreate to generate a new file; this report is based on the current set of source files. The source files are sorted before the report is created and will always be in the same order
- Selecting Append Only New Data will simply update the file, adding results to the end of the report. Processing the same project several times will add data to the report without deleting existing data.

Remove Auxillary Columns

Choose whether reports should contain auxillary columns.

Split File By

This feature allows you to create a separate export file based on one or more features.

18.8 Export Vector Layers

Export vector layers to file. If you use Definiens Data Management in combination with the Definiens Server this algorithm provides additional parameters for connecting to the ArcGIS Server.

18.8.1 Supported Domains

Image Object Level

18.8.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

18.8.3 Export Data Parameters

Attribute Table

Attribute table.

Shape Type

Select a type of shape for export – polygons, lines or points.

Export Type

Select a type of export for the type of shape selected:

- Polygons (raster or smoothed)
- Lines (main line or skeleton)
- Points (center of main line or center of gravity)

Dimension

Dimension of vertex co-ordinates in the shape file. If 2D dimension for a 3D scene is selected, then a vertical series of slices is exported.

Use Geocoded Co-ordinates

Choose between geo co-ordinates or pixels for exported vertices in the shapefile.

Co-ordinates Orientation

Specifies how the given co-ordinates should be interpreted.

18.8.4 Export Format Parameters

Name of Feature Class to Export

Click to open the Select Feature Class to Export dialog box and select a feature class. This field displays if you use Definiens Data Management in combination with the Definiens Server.

Export Format

Click the drop-down arrow to select Shapefile.

Desktop Export Folder

Use the drop-down list to browse to a storage location. Displays only if Use Export Item is selected in the Export Mode field.

18.9 Export Image Object View

The Export Image Object View algorithm lets you export an image file for each image object.

18.9.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects; Image Object List

18.9.2 Output Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Customize Path

If Static Export Item or Dynamic Export Item are selected and this parameter is set to 'yes', you may enter an file export location in the Export Path field.

Export Series

If set to 'yes', then multiple files per series will be exported, or additional columns will be created for table exports.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

Image File Name

Enter the file name for the exported image. If multiple files are exported, this name is used as a prefix, to which numbers are added.

18.9.3 Settings

View Settings Source

Select from 'local' or 'from process':

- If you select ‘local’, you can use the local settings derived from the *Save Current View Settings* below parameter
- Selecting ‘from process’ displays the View Settings Process Path parameter. You can use this parameter in conjunction with the View Settings parameter of the *Save/Restore View Settings* (p 179) algorithm.

Save Current View Settings

Click the ellipsis button to capture current view settings.

Border Size Around Object

Enter a value for the thickness of the pixels around the bounding box of the exported image object (the default is 0).

Use Fixed Image Size

Select ‘yes’ to export to a predetermined image size.

Size X

Enter a value to fix the width (x -direction) of the exported image.

Size Y

Enter a value to fix the height (y -direction) of the exported image.

Draw Object Outline

If ‘yes’ is selected, a red outline is drawn around the image object.

18.10 Export Mask Image

Export the data of specific regions of interest to a file. A background fill color is used for all areas of the scene copy that are not part of the selected regions of interest. For three-dimensional maps, an individual file is exported per image object slice of the original map.

18.10.1 Supported Domains

Image Object Level

18.10.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Export Series

If set to yes, then multiple files per series will be exported, or additional columns will be created for table exports.

Scale

See *Scale* (p 124) for an explanation of the Select Scale dialog box.

Default File Format

Select the export file type used for desktop processing. If the algorithm is run in desktop mode, files will stored in this format. In server processing mode, the file format is defined in the export settings specified in the workspace.

Background Fill Color

Select the channel value of the required background fill color. You can enter any integer value that represents an actual gray value. If you enter a value that is invalid for the current image file, it will automatically be changed to the closest valid one.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace. This feature is only available if Use Export Item is selected in the Export Mode field.

18.11 Export Image

Export an image using predefined settings.

18.11.1 Supported Domains

None

18.11.2 Algorithm Parameters

Layer

Select the layer for export.

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`

Default File Format

Select from the following formats if exporting in desktop mode (if you are using server mode, then the export settings defined in the workspace will define the file format):

- Tagged Image Files (Geocoded) (*.tif)
- Tagged Image Files (*.tif)
- Erdas Imagine Images (*.img)
- JPEG JFIF (*.jpg)
- JPEG2000 (*.jp2)
- Portable Network Graphics (*.png)
- Windows or OS/2 Bitmap (*.bmp)
- National Imagery Transmission (NITF) (*.ntf)
- PCIDSK (*.pix)
- 8-bit png (*.png)
- 16-bit png (*.png)

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace.

Geo-Coding Shift X

Shift the geocoding lower-left corner in the X direction. Use the drop-down list to shift half a pixel to the left or right.

Geo-Coding Shift Y

Shift the geocoding lower-left corner in the Y direction. Use the drop-down list to shift half a pixel to the left or right.

18.12 Export Result Preview

The Export Result Preview algorithm exports results as overlay images, which allows faster previews of results.

18.12.1 Supported Domains

Pixel Level; Image Object Level

18.12.2 Algorithm Parameters

Export Mode

- Static Export Item writes the export item to the workspace
  - Enter a name in Export Item Name, or use the default name
- Dynamic Export Item lets you enter or select a variable as an export item
  - Select a variable from the drop-down box or enter a name in the Export Item Variable Name field (entering a name launches the Create Variable dialog box, where you can enter a value and variable type)
- Use Explicit Path exports to the location defined in the Export Path field. The default export path is `{:Workspc. OutputRoot} \results\{:Item. Name} \{:Project. Name}.v {:Project.Ver}.{:Ext}`.

Customize Path

Select 'yes' to customize the default export path.

Mode

Select the algorithm mode:

- Create creates a new result preview layer from the domain level
- Create from Subscenes creates a new result preview layer from specified subscenes
- Append appends or creates a domain level into a result preview layer
- Generate Overview generates an overview of an existing result preview layer.

Target Map

Enter the target map for the result preview export.

Generate Overview

Select ‘yes’ to generate an overview.

Desktop Export Folder

Specify the file export folder used for desktop processing. If the algorithm is run in desktop mode, files will be stored at this location. In server processing mode, the file location is defined in the export settings specified in the workspace.

19 Image Registration Algorithms

Image registration algorithms are used to transform two or more sets of image data into one system of co-ordinates.

19.1 Image Registration

Perform the registration of the scenes stored in two different maps.

The resulting registered scene is stored in a new map. Landmark-based image registration requires that you have created landmarks before. This can be done either manually using a Definiens client or automatically by rule sets using the Set Landmarks algorithm.

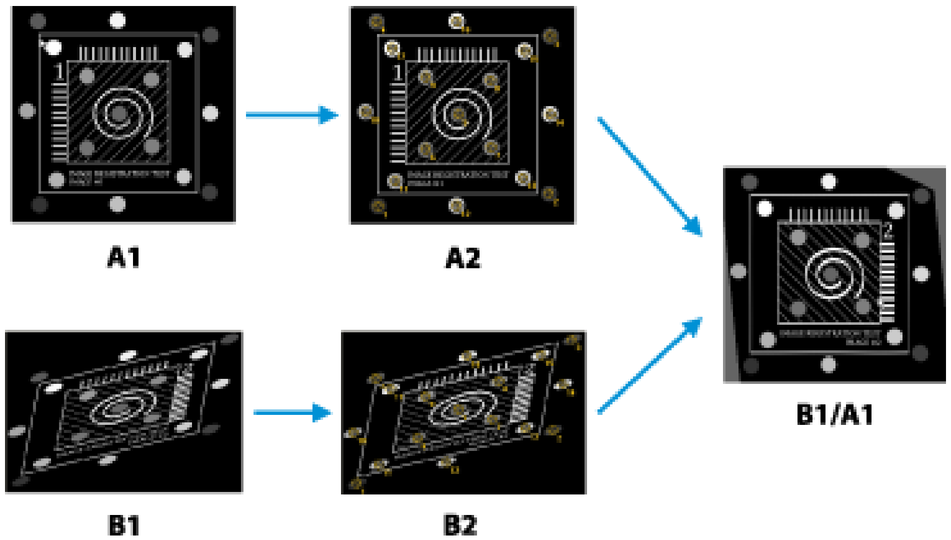


Figure 19.1. Workflow for registering two 2D images

Landmarks may be created manually or automatically with results similar to A2 and B2. B1/A1 illustrates the result of the registration of B1 with respect to A1.

19.1.1 Supported Domains

Execute

19.1.2 Algorithm Parameters

Target Map Name

Name of the target map that provides the matched data.

Maps to Register

List of maps that should be registered to the map selected in the domain.

Transformation Type

- Elastic (Spline) – use an elastic transformation algorithm based on splines
- Affine – use an affine transformation.

Registration Mode

This option is only active for affine transformation types.

- Landmarks – use landmarks for registration
- Automatic by pixel brightness – use automatic registration by optimizing the matching pixel brightness.

Fixed Layer

This parameter is only available for affine transformations using the Automatic by pixel brightness mode. Select the reference layer for pixel brightness optimization.

Iterations

This parameter is only available for affine transformations. Number of iterations when optimizing the affine transformation parameters.

Save Transformation In Parameter Set

Name of the parameter set to save the affine transformation matrix in case of a single map registration.

Resample Type

Resampling method for the transformed image layer.

Value	Description
Linear	Use linear resampling method. This methods offers a good trade-off between quality and performance.
B-spline	Use B-spline resampling method. This method can produce better resampling results than linear but requires more computing time.
Nearest Neighbor	This resampling method is the fastest. It may produce results with less quality than linear or B-spline.

Default Pixel Value

Pixel value to use for empty areas in the transformed image.

19.2 Delete Landmarks

Delete all landmarks in the specified map.

19.2.1 Supported Domains

Execute

19.3 Set Landmark

Set a landmark at the center of each object in the selected domain.

19.3.1 Supported Domains

Image Object Level; Current Image Object; Neighbor Image Object; Super Object; Sub Objects; Linked Objects

20 About Features

20.1 About Features as a Source of Information

Image objects have spectral, shape, and hierarchical characteristics. These characteristic attributes are called features in Definiens software. Features are used as source of information to define the inclusion-or-exclusion parameters used to classify image objects.

There are two major types of features:

- Object features are attributes of image objects, for example area
- Global features are not connected to an individual image object, for example the number of image objects of a certain class.

20.1.1 Conversions of Feature Values

The conversion of feature values is handled differently, depending on the value type:

- Values identifying a position (position values)
- Values identifying certain distance measurements such as length or are (unit values).

Conversion of Position Values

Position values can be converted from one co-ordinate system to another. The following position conversions are available:

- If the unit is a pixel, a position within the pixel co-ordinate system is identified
- If the unit is a co-ordinate, a position within the user co-ordinate system is identified

The position conversion is applied for image object features such as Y center, Y max and X center.

Conversion of Unit Values

Distance values such as length and area are initially calculated in pixels. They can be converted to a distance unit. To convert a pixel value to a unit, the following information is needed:

- Pixel size in meters
- Value dimension, for example 1 for length and 2 for area
- Unit factor, relative to the meter, for example 1 for meter, 100 for centimeter and 0.001 for kilometer

The following formula is valid for converting value from pixel to a unit:

$\mathrm{val}_{\mathrm{unit}} = \mathrm{val}_{\mathrm{pixel}} \times u^{\mathrm{dim}} \times F$

(Where u is pixel size in units, F is the unit factor and dim the dimension.)

20.2 Object Features

Object features are calculated by evaluating image objects themselves as well as their embedding in the image object hierarchy. Object Features are grouped as follows:

- Customized object features are user-defined and reference existing object features
- Type features refer to the spatial connectivity of an image object
- Layer value features evaluate the first, second, and third statistical moment (mean, standard deviation, and skewness) of an image object's pixel value and the object's relations to other image object's pixel values. Use these to describe image objects with information derived from their spectral properties
- Geometry features evaluate the image object's shape. The basic geometry features are calculated based on the pixels forming an image object. If image objects of a certain class stand out because of their shape, you are likely to find a geometry feature that describes them
- Position features refer to the position of an image object relative to the entire scene. These features are of special interest when working with geographically referenced data, as an image object can be described by its geographic position
- Texture features are used to evaluate the texture of image objects. They include texture features based on an analysis of sub-objects helpful for evaluating highly textured data. In addition, features based upon the gray level co-occurrence matrix after Haralick are available
- Object Variables are local variables. In contrast to scene variables they store values for each individual image object. (Think of it as each image object having its own version of the variable.) There is one instance per image object in a project
- Hierarchy features provide information about the embedding of an image object within the image object hierarchy.

Thematic attribute features are used to describe an image object using information provided by thematic layers. If your scene contains a thematic layer, its thematic object's properties, can be used to create thematic attribute features, which can be used for developing ruleware. Depending on the attributes of the thematic layer, a large range of different features becomes available.

21 Object Features: Customized

Object Features > Customized

Customized features are Definiens features that you can create and adapt to your needs. They can be arithmetic or relational features that depend on other existing features. All customized features are based on the features shipped with Definiens Developer XD 2.0.4, as well as newly created customized features.

- Arithmetic features are composed of existing features, variables, and constants, which are combined via arithmetic operations. Arithmetic features can be composed of multiple features but apply only to a single object.
- Relational features are used to compare a particular feature of one object to those of related objects of a specific class within a specified distance. Related objects are surrounding objects such as neighbors, sub-objects, superobjects, sub-objects of a superobject or a complete image object level. Relational features are composed of only a single feature but refer to a group of related objects.

21.1 Create Customized Features

The Manage Customized Features dialog box allows you to add, edit, copy, and delete customized features. It enables you to create new arithmetic and relational features based on the existing ones.

1. To open the Manage Customized Features dialog box, do one of the following:
    - On the menu bar click on Tools and then select Manage Customized Features.
    - On the Tools toolbar click on the Manage Customized Features icon.
2. Click Add to create a new customized feature. The Customized Features dialog box will open, providing you with tools for the creation of arithmetic and relational features.
3. To edit a feature, select it and click Edit to open the Customized Features dialog box
4. To copy or delete a feature, select it and click Copy or Delete.

21.2 Arithmetic Customized Features

Object Features > Customized > Create New Arithmetic Feature

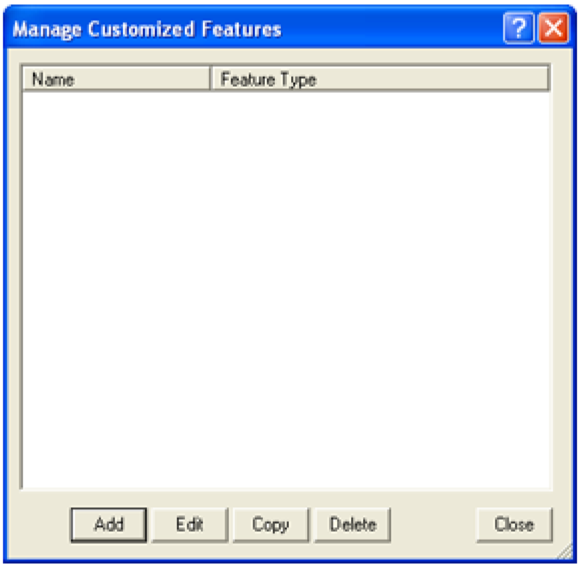


Figure 21.1. Manage Customized Features dialog box

To create an arithmetic customized feature:

1. In the Feature View window, double-click on Create New Arithmetic Feature
2. Insert a name for the customized feature to be created
3. Use the calculator to create the arithmetic expression. You can:
   - Type in new constants
   - Select features or variables in the feature tree on the right
   - Choose arithmetic operations or mathematical functions
4. The expression you create is displayed in the text area above the calculator
5. To calculate or delete an arithmetic expression, highlight the expression with the cursor and then click either Calculate or Del as appropriate
6. You can switch between degrees (Deg) or radians (Rad) measurements and invert the expression
7. To create the new feature click Apply, to create the feature without leaving the dialog box, or OK to create the feature and close the dialog box
8. After creation, the new arithmetic feature can be found in either one of the following locations:
   - In the Image Object Information window
   - In the Feature View window under Object features > Customized

NOTE: The calculator buttons are arranged in a standard layout. In addition:

- ^ signifies an exponent (x^2 for x^2) or a square root (x^0.5 for $\sqrt{x}$)
- Use abs for an absolute value
- Use floor to round down to the next lowest integer (whole value). You can use floor (0.5+x) to round up to the next integer value.

(Note that e is the exponential function and PI (P) is π .)

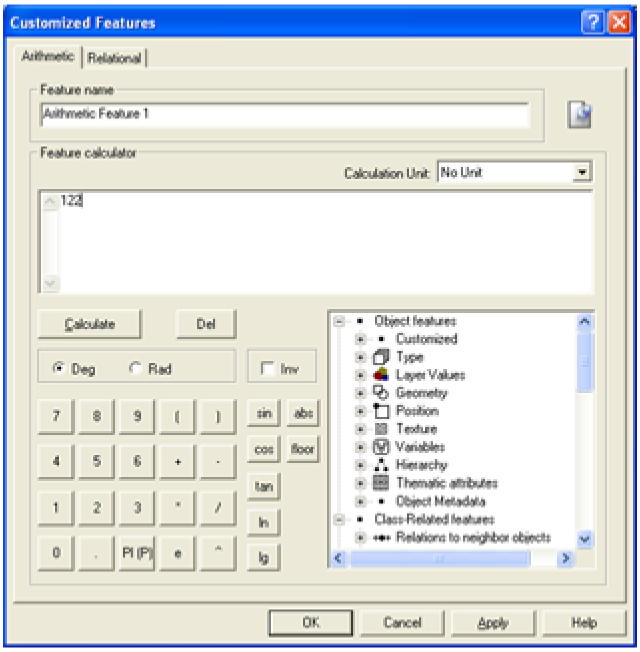


Figure 21.2. Creating an arithmetic feature in the Customized Features dialog box

21.3 Relational Customized Features

Object Features > Customized > Create New 'Relational Feature'

To create a relational customized feature:

1. In the Feature View window, double-click on Create New 'Relational Feature'
2. Insert a name for the relational feature to be created.
3. In the Relational Function area, select the relation existing between the image objects
4. In the drop-down list, choose the relational function to be applied.
5. Define the distance of the related image objects. Depending on the related image objects, the distance can be either horizontal (units, for example, pixels) or vertical (image object levels)
6. In the Feature Selection pane, select the feature for which to compute the relation.
7. In Class Selection, select a class, group or no class to apply the relation.
8. To create the new feature click Apply to create the feature without leaving the dialog box or OK to create the feature and close the dialog box
9. After creation, the new relational feature can be found in the Feature View window under Class-Related Features > Customized.

> **NOTE:** As with class-related features, the relations refer to the groups hierarchy. This means if a relation refers to one class, it automatically refers to all subclasses of this class in the groups hierarchy.

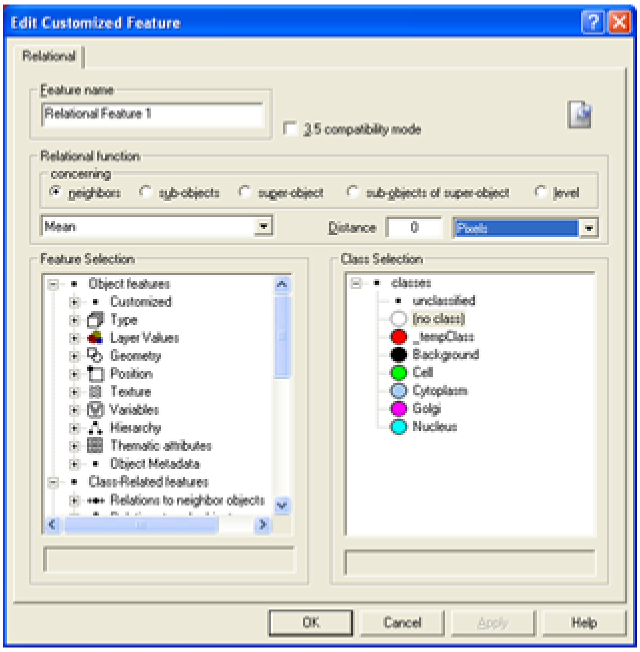


Figure 21.3. Creating a relational feature at the Customized Features dialog box

21.3.1 Relations Between Surrounding Objects

Relations between surrounding objects can exist either on the same level or on a level lower or higher in the image object hierarchy (table 21.1, *Relations Between Surrounding Objects*).

Table 21.1. Relations Between Surrounding Objects

Relation	Description
Neighbors	Related image objects on the same level. If the distance of the image objects is set to 0 then only the direct neighbors are considered. When the distance is greater than 0 then the relation of the objects is computed using their centers of gravity. Only those neighbors whose center of gravity is closer than the distance specified from the starting image object are considered. The distance is calculated either in metric units or pixels. For example, a direct neighbor might be ignored if its center of gravity is further away from the specified distance.
Sub-objects	Image objects that exist under other image objects (superobjects) whose position in the hierarchy is higher. The distance is calculated in levels.
Superobject	Contains other image objects (sub-objects) on lower levels in the hierarchy. The distance is calculated in levels.
Sub-objects of superobject	Only the image objects that exist under a specific superobject are considered in this case. The distance is calculated in levels.

Continues...

Relation	Description
Level	Specifies the level on which an image object will be compared to all other image objects existing at this level. The distance is calculated in levels.

21.3.2 Relational Functions

An overview of all functions existing in the drop-down list under the Relational Function section is shown in table 21.2 on the current page, *Relational Function Options*.

Table 21.2. Relational Function Options

Relational function	Description
Mean	Calculates the mean value of selected features of an image object and its neighbors. You can select a class to apply this feature or no class if you want to apply it to all image objects. Note that for averaging, the feature values are weighted with the area of the image objects.
Standard deviation	Calculates the standard deviation of selected features of an image object and its neighbors. You can select a class to apply this feature or no class if you want to apply it to all image objects.
Mean difference	Calculates the mean difference between the feature value of an image object and its neighbors of a selected class. Note that the feature values are weighted by either by the border length (distance = 0) or by the area (distance > 0) of the respective image objects.
Mean absolute difference	Calculates the mean absolute difference between the feature value of an image object and its neighbors of a selected class. Note that the feature values are weighted by either by the border length (distance = 0) or by the area (distance > 0)of the respective image objects.
Ratio	Calculates the proportion between the feature value of an image object and the mean feature value of its neighbors of a selected class. Note that for averaging the feature values are weighted with the area of the corresponding image objects.
Sum	Calculates the sum of the feature values of the neighbors of a selected class.
Number	Calculates the number of neighbors of a selected class. You must select a feature in order for this feature to apply, but it does not matter which feature you pick.
Min	Returns the minimum value of the feature values of an image object and its neighbors of a selected class.
Max	Returns the minimum value of the feature values of an image object and its neighbors of a selected class.

Continues...

Relational function	Description
Mean difference to higher values	Calculates the mean difference between the feature value of an image object and the feature values of its neighbors of a selected class, which have higher values than the image object itself. Note that the feature values are weighted by either by the border length (distance = 0) or by the area (distance > 0) of the respective image objects.
Mean difference to lower values	Calculates the mean difference between the feature value of an image object and the feature values of its neighbors of a selected class, which have lower values than the object itself. Note that the feature values are weighted by either by the border length (distance = 0) or by the area (distance > 0) of the respective image objects.
Portion of higher value area	Calculates the portion of the area of the neighbors of a selected class, which have higher values for the specified feature than the object itself to the area of all neighbors of the selected class.
Portion of lower value area	Calculates the portion of the area of the neighbors of a selected class, which have lower values for the specified feature than the object itself to the area of all neighbors of the selected class.
Portion of higher values	Calculates the feature value difference between an image object and its neighbors of a selected class with higher feature values than the object itself divided by the difference of the image object and all its neighbors of the selected class. Note that the features are weighted with the area of the corresponding image objects.
Portion of lower values	Calculates the feature value difference between an image object and its neighbors of a selected class with lower feature values than the object itself divided by the difference of the image object and all its neighbors of the selected class. Note that the features are weighted with the area of the corresponding image object.
Mean absolute difference to neighbors	Available only if sub-objects is selected for Relational function concerning. Calculates the mean absolute difference between the feature value of sub-objects of an object and the feature values of a selected class. Note that the feature values are weighted by either by the border length (distance = 0) or by the area (distance > 0) of the respective image objects.

21.4 Finding Customized Features

You can find customized features at different places in the feature tree, depending on the features to which they refer. For example, a customized feature depending on an object feature is sorted below the group Object Features > Customized.

If a customized feature refers to different feature types, they are sorted in the feature tree according to the interdependencies of the features used. For example, a customized feature with an object feature and a class-related feature displays below Class-Related Features.

22 Object Features: Type

Object Features > Type

Type features are based on whether image objects are physically connected within a scene.

22.1 Is 3D

Object Features > Type > Is 3D

The Is 3D feature checks if an image object has three dimensions. If the image object type is 3D, the feature value is 1 (true), if not it is 0 (false).

22.2 Is Connected

Object Features > Type > Is Connected

The Is Connected feature checks if the image object is connected. If the image object type is connected in two-dimensional space, the feature value is 1 (true), otherwise it is 0 (false).

23 Object Features: Layer Values

Object Features > Layer Values

Layer value features evaluate the first (mean), second (standard deviation), and third (skewness) statistical moments of an image object's pixel value and the object's relations to other image objects' pixel values. Use these to describe image objects with information derived from their spectral properties.

23.1 Mean

Object Features > Layer Values > Mean

Features in this group refer to the mean layer intensity value of an image object.

23.1.1 Brightness

Object Features > Layer Values > Mean > Brightness

Editable Parameter

To set which image layers providing the spectral information are used for calculation, select Classification > Advanced Settings > Select Image Layers for Brightness from the main menu. The Define Brightness dialog box opens.

Parameters

- w_k^B is the brightness weight of image layer k with $w_k^B = \{^0_1$
- K is the number of image layers k used for calculation
- w^B is the sum of brightness weights of all image layers k used for calculation with $w^B = \sum_{k=1}^K w_k^B$
- $\bar{c}_k(v)$ is mean intensity of image layer k of image object v
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\min}$ is the brightest possible intensity value of image layer k.

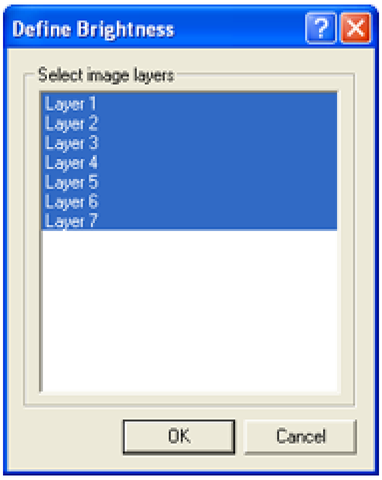


Figure 23.1. Define Brightness dialog box

Expression

$$\bar{c}(v) = \frac{1}{w^B} \sum_{k=1}^{K} w_k^B \bar{c}_k(v)$$

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

Conditions

- The scene includes more than one image layer
- Because combined negative and positive data values would create an erroneous feature value for brightness, it is only calculated using image layers with positive values.

23.1.2 Layer 1/2/3

Object Features > Layer Values > Mean > Layer

The mean intensity of all pixel/voxels forming an image object.

Parameters

- P_v is the set of pixels/voxels of an image object v with $P_v = \{(x,y,z,t) : (x,y,z,t) \in v\}$
- $\#P_v$ is the total number of pixel/voxels contained in P_v
- $c_k(x,y,z,t)$ is the image layer intensity value at pixel/voxel (x,y,z,t)
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- $\bar{c}_k$ is the mean intensity of image layer k.

Expression

$$\bar{c}_k(v) = \bar{c}_k(P_v) = \frac{1}{\#P_v} \sum_{(x,y,z,t)\in P_v} c_k(x,y,z,t)$$

Feature Value

$[c_k^{\min}, c_k^{\max}]$

23.1.3 Max. Diff.

Object Features > Layer Values > Mean > Max. Diff.

Parameters

- i, j are image layers
- $\bar{c}(v)$ is the brightness of image object v
- $\bar{c}_i(v)$ is the mean intensity of image layer i of image object v
- $\bar{c}_j(v)$ is the mean intensity of image layer j of image object v
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- K_B are image layers of positive brightness weight with $K_B = \{k \in K : w_k = 1\}$, where w_k is the image layer weight.

Expression

$$\frac{\max\limits_{i,j \varepsilon K_B} \left|\bar{c}_i(v) - \bar{c}_j(v)\right|}{\bar{c}(v)}$$

Feature Value Range

$$\left[0, \frac{1}{K_B} c_k^{\max}\right]$$

Typically, the feature values are between 0 and 1.

Conditions

- The scene includes more than one image layer
- If $\bar{c}(v) = 0$ the expression is undefined.

23.2 Standard Deviation

Object Features > Layer Values > Standard Deviation

The standard deviation is calculated from the image layer intensity values of all pixel/voxels forming an image object.

23.2.1 Layer 1/2/3

Object Features > Layer Values > Standard Deviation > Layer…

For each image layer, a separate standard deviation feature is listed in Feature View.

Parameters

- $\sigma_k(v)$ is the standard deviation of intensity values of image layer k of all pixel/voxels forming an image object v
- P_v is the set of pixel/voxels of an image object v
- $\#P_v$ is the total number of pixel/voxels contained in P_v
- (x,y,z,t) are the pixel/voxel co-ordinates
- $c_k(x,y,z,t)$ is the image layer intensity value at pixel/voxel (x,y,z,t)
- c_k^{range} is the data range of image layer k with $c_k^{\text{range}} = c_k^{\max} - c_k^{\min}$.

Expression

$$\sigma_k(v) = \sigma_k(P_v) =$$

$$\sqrt{\frac{1}{\#P_v}\left(\sum_{(x,y,z,t)\in P_v} c_k^2(x,y,z,t) - \frac{1}{\#P_v}\left(\sum_{(x,y,z,t)\in P_v} c_k(x,y,z,t)\right)^2\right)}$$

Feature Value Range

$$\left[0, \frac{1}{2}c_k^{\text{range}}\right]$$

23.3 Skewness

Object Features > Layer Values > Skewness

The Skewness feature describes the distribution of all the image layer intensity values of all pixel/voxels that form an image object; this distribution is typically Gaussian. The value is calculated by the asymmetry of the distribution of image layer intensity values in an image object.

A normal distribution has a skewness of zero. A negative skewness value indicates that an image object has more pixel/voxels with an image layer intensity value smaller than the mean; a positive value indicates a value larger than the mean.

23.3.1 Layer Values

For each image layer, a separate skewness feature is listed in Feature View.

Parameters

- $\gamma_k(v)$ is the skewness of intensity values of image layer k of an image object v
- P_v is the set of pixel/voxels of image object v
- $\#P_v$ is the total number of pixel/voxels contained in P_v
- (x,y,z,t) are the pixel/voxel co-ordinates
- $c_k(x,y,z,t)$ is the image layer value at pixel/voxel (x,y,z,t)
- $c_k(v)$ is the mean intensity of image layer k of an image object v
- $c_k^{\max}$ is the brightest image layer intensity value of image layer k.

Expression

$$\gamma_k(v) = \gamma_k(P_v) = \sqrt{\#P_v} = \frac{\sum\limits_{(x,y,z,t)\in P_v} \Big(c_k(x,y,z,t) - \bar{c}_k(v)\Big)^3}{\left(\sum\limits_{(x,y,z,t)\in P_v} \Big(c_k(x,y,z,t) - \bar{c}_k(v)\Big)^2\right)^{\frac{3}{2}}}$$

Feature Value Range

$[-(c_k^{\max})^3, (c_k^{\max})^3]$

23.4 Pixel Based

Object Features > Layer Values > Pixel Based

Features in this group refer to the values of specific pixel/voxels in an image object.

23.4.1 Ratio

Object Features > Layer Values > Pixel Based > Ratio

The amount that a given image layer contributes to the total brightness.

Editable Parameters

Image Layer

Parameters

- w_k^B is the brightness weight of image layer k
- $\bar{c}_k(v)$ is the mean intensity of image layer k of an image object v
- $\bar{c}(v)$ is brightness.

Expression

If $w_k^B = 1$ and $c(v) \neq 0$ then:

$$\frac{\bar{c}_k(v)}{\sum_{k=1}^{n} 1\bar{c}_k(v)}$$

If $w_k^B = 0$ or $c(v) = 0$ then the ratio is equal to 0.

Feature Value Range

$[0,1]$

Conditions

- The scene includes more than one image layer
- Only image layers containing spectral information can be used to achieve reasonable results
- Because combined negative and positive data values would create an erroneous feature value for brightness, it is only calculated with image layers of positive values.

23.4.2 Min. Pixel Value

Object Features > Layer Values > Pixel Based > Min. Pixel Value

The value of the pixel/voxel with the minimum layer intensity value in the image object.

Editable Parameters

Image Layer

Parameters

- (x,y) are pixel/voxel co-ordinates
- $c_k(x,y)$ is the image layer intensity value at pixel/voxel (x,y,z,t)
- $c_k^{\min}$ darkest possible intensity value of image layer k
- $c_k^{\max}$ brightest possible intensity value of image layer k
- P_v is the set of pixel/voxels of an image object v.

Expression

$$\min_{(x,y)\in P_v} c_k(x,y)$$

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

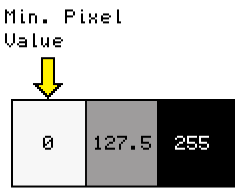


Figure 23.2. Minimum pixel value of an image object consisting of three pixels

Conditions

None

23.4.3 Max. Pixel Value

Object Features > Layer Values > Pixel Based > Max. Pixel Value

The value of the pixel/voxel with the maximum layer intensity value of the image object.

Editable Parameters

Image Layer

Parameters

- (x,y,z,t) are pixel/voxel co-ordinates
- $c_k(x,y,z,t)$ are image layer intensity value at pixel/voxel (x,y,z,t)
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- P_v is the set of pixel/voxels of image object v.

Expression

$$\max_{(x,y)\in P_v} c_k(x,y)$$

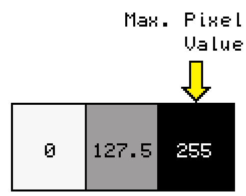


Figure 23.3. Maximum pixel value of an image object consisting of three pixels

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

23.4.4 Mean of Inner Border

Object Features > Layer Values > Pixel Based > Mean of Inner Border

The mean layer intensity value of the pixel/voxels belonging to an image object, which shares its border with other image objects, thereby forming an inner border.

Editable Parameters

Image Layer

Parameters

- $\bar{c}_k$ is the mean intensity of image layer k
- P_v^{Inner} is the set of inner border pixel/voxels of image object v with $P_v^{\text{Inner}} = \{(x,y,z) \in P_v : \exists (x',y',z') \in N_6(x,y,z) : (x',y',z') \notin P_v\}$
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k.

Expression

$\bar{c}_k(P_v^{\text{Inner}})$

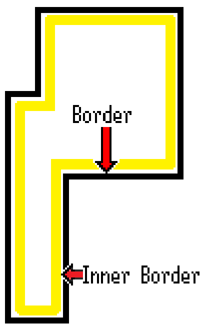


Figure 23.4. Inner border of a image object

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

23.4.5 Mean of Outer Border

Object Features > Layer Values > Pixel Based > Mean of Outer Border

The mean layer intensity value of pixel/voxels not belonging to an image object of interest, which shares its border, thereby forming the outer border of the image object.

Editable Parameter

Image Layer

Parameters

- $\bar{c}_k$ is the mean intensity of image layer k
- P_v^{Outer} is the set of outer border pixel/voxels of image object v with $P_v^{\text{Outer}} = \{(x,y,z) \in P_v : \exists (x',y',z') \in N_6(x,y,z) : (x',y',z') \notin P_v\}$
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k.

Expression

$\bar{c}_k(P_v^{\text{Outer}})$

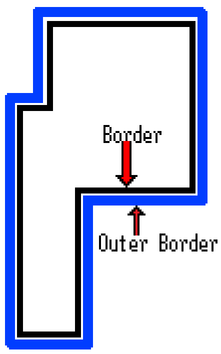


Figure 23.5. Outer borders of a image object

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

23.4.6 Border Contrast

Object Features > Layer Values > Pixel Based > Border Contrast

The Border Contrast algorithm looks at each edge pixel p , bordering an image object in x , y and z directions and calculates how much it contrasts with the neighboring object at the border.

Editable Parameter

Image Layer.

Parameters

- Let v be an image object,
- Let $E(v) = \{$set of all pixel pairs (p,q) : p is in $P_{v_{\text{outer}}}$ and q is in $P_{v_{\text{inner}}}$ and q is in $N_{6(p)}$ $\}$ be the set of all pixel edges of v
- For e in $E(v)$ the pixel edge contrast in layer k is contrast(e) : $c_{k(q)} - c_{k(p)}$

The border contrast is defined as the mean value of the pixel edge contrasts for all pixel edges in $E(v)$:

$$bc = \frac{1}{n} \times \sum_{[e=(p,q) \text{ in } E(v)]} c_{k(q)} - c_{k(p)}$$

with $n = \#E(v)$

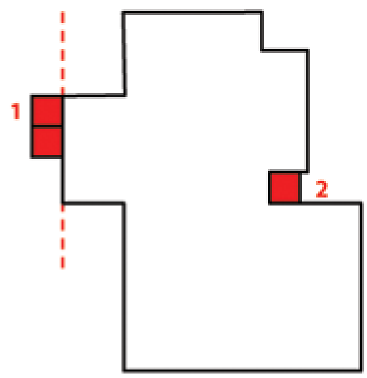


Figure 23.6. Border Contrast algorithm: Where a pixel has a single border with an image object (1), this value is taken; where it has multiple borders (2), the different is subtracted

23.4.7 Contrast to Neighbor Pixels

Object Features > Layer Values > Pixel Based > Contrast to Neighbor Pixels

The mean difference in contrast compared to a surrounding volume of a given size. This feature is used to find borders and gradations in the scene.

Editable Parameters

Image Layer; Distance (edit the size of the surrounding volume)

Parameters

- $B_v(d)$ is the extended bounding box of an image object v with distance d with $B_v(d)$ equal to $\{(x,y,z) : x_{\min}(v) - d \leq x \leq x_{\max}(v) + d \ , \ y_{\min}(v) - d \leq y \leq y_{\max}(v) + d \ , \ z_{\min}(v) - d \leq z \leq z_{\max}(v) + d\}$
- P_v is the set of pixel/voxels of an image object v
- $\bar{c}_k$ is the mean intensity of image layer k.

Expression

$$1000 \times \left(1 - \frac{\bar{c}_k\big(B_v(d) - P_v\big)}{1 + \bar{c}_k(P_v)}\right)$$

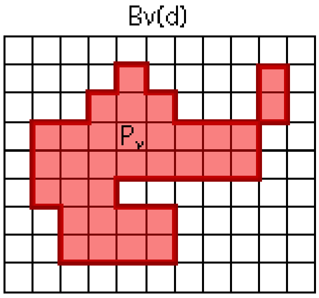


Figure 23.7. The surrounding area of and image object v defined by its bounding box with a distance of one pixel

Feature Value Range

$[-1000, 1000]$

Conditions

- The distance d should always be greater than 0
- If $d = 0$, then $B_v(d) = B_v$; if $Bv = Pv$, the formula is invalid
- If unsigned data exists then maybe $\bar{c}_k(P_v) = -1$ $\therefore$ the formula is invalid
- If $\bar{c}_k(P_v) = 0$, the values are meaningless.

23.4.8 Edge Contrast of Neighbor Pixels

Object Features > Layer Values > Pixel Based > Edge Contrast of Neighbor Pixels

The Edge Contrast of Neighbor Pixels feature describes the edge contrast of an image object to the surrounding volume of a given size. It is used to find edges in a scene. For calculation, the pixel/voxels in the surrounding volume of a given size are compared to the mean image layer intensity of the image object, subdivided into groups of brighter and darker pixel/voxels. The value is calculated by the mean of the image layer intensity value of the brighter pixel/voxels subtracted by the mean value of the darker pixel/voxels.

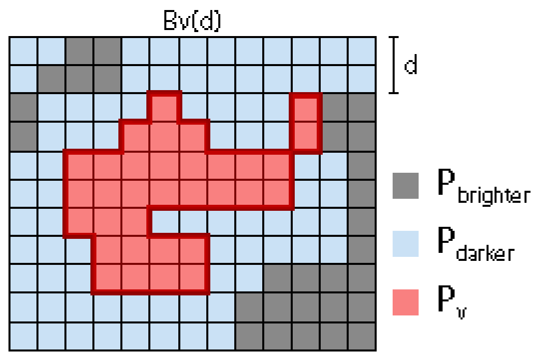


Figure 23.8. Pixels of the surrounding area are compared to the mean image layer intensity of the image object v

Editable Parameters

Image Layer; Distance (the size of the surrounding area)

Parameters

- $P_{\text{brighter}} : \{P : c_k(x,y,z) > \bar{c}_k(v)\}$ within the surrounding volume of image object v defined by the bounding box $(B_v(d))$
- $P_{\text{darker}} : (B_v(d)) = \{P : c_k(x,y,z) < \bar{c}_k(v)\}$ within the surrounding volume of image object v defined by the bounding box $(B_v(d))$
- $(B_v(d))$ is the extended bounding box of an image object v with distance d with $\{(x,y,z) : x_{\min}(v) - d \leq x \leq x_{\max}(v) + d \text{ , } y_{\min}(v) - d \leq y \leq y_{\max}(v) + d \text{ , } z_{\min}(v) - d \leq z \leq z_{\max}(v) + d\}$
- $c_k(x,y,z)$ is the image layer intensity value at pixel/voxel (x,y,z)
- $\bar{c}_k(v)$ is the mean intensity of image layer k of all pixel/voxels forming an image object v
- P_v is the set of pixel/voxels of an image object v.

Expression

$\bar{c}_k(P_{\text{brighter}}) - \bar{c}_k(P_{\text{darker}})$

Feature Value Range

$[0, 255]$

Conditions

If $P_{\text{brighter}}(B_v(d)) = \varnothing$ or $P_{\text{darker}}(B_v(d)) = \varnothing$ then $\bar{c}_k(v)$ is returned.

23.4.9 Std Dev. to Neighbor Pixels

Object Features > Layer Values > Pixel Based > Std Dev. to Neighbor Pixels

The standard deviation of layer intensity values of pixel/voxels within the surrounding volume of a given size. The surrounding volume consists of the pixel/voxels located within the extended bounding box of an image object, but do no belong this image object.

Editable Parameters

Image Layer; Distance (edit the size of the surrounding volume)

Parameters

- P_v is the set of pixel/voxels of an image object v
- $(B_v(d))$ is the extended bounding box of an image object v with distance d with $\{(x,y,z) : x_{\min}(v) - d \leq x \leq x_{\max}(v) + d\ ,\ y_{\min}(v) - d \leq y \leq y_{\max}(v) + d\ ,\ z_{\min}(v) - d \leq z \leq z_{\max}(v) + d\}$.

Expression

$\sigma_k(B_v(d) - P_v)$

Feature Value Range

$$\left[0, \frac{c_k^{\max}}{2}\right]$$

Conditions

If $d = 0$, then $B_v(d) = B_v$ and if $B_v = P_v \therefore$ the formula is invalid.

23.4.10 Circular Mean

Object Features > Layer Values > Pixel Based > Circular Mean

Calculates the mean for all pixels within a ring around the center of an image object. The thickness of the ring is defined by two radius values R_1 and R_2 .

Editable Parameters

- Layer
- Radius Mode
  - Number of Pixels – radius calculated from the number of pixels in the object $R = \sqrt{\frac{\#\mathrm{Pxl}}{\pi}}$
  - Length – radius will equal half the length
  - Width – radius will equal half the width

    - User – radius will be specified in next parameter
- Delta/User – defines delta in pixels for radius of number of pixels, length, width and user modes. It can be positive, negative or zero
- Second Radius Mode
    - Same ($= R_1$), border – second radius will be equal to first radius. Pixels will be used that form a thin border along the first radius
    - None (=0), entire circle – second radius is set to zero. Entire circle of first radius will be processed
    - Relative to R_1 – second radius is set to R_1 . Users can define an offset (positive, negative number or zero) to where the second radius will be extended (positive number) or shortened (negative number)
    - User – radius will be specified in next parameter.

Parameters

- R_1 is the inner radius of the ring
- R_2 is the outer radius of the ring
- c is the center of the object
- $d(u,v)$ is the distance between objects.

Expression

$(R_1 - 0.5) \leq d(u,v) \leq (R_2 + 0.5)$

Feature Value Range

$[R_1, R_2]$

Conditions

None

23.4.11 Circular StdDev

Object Features > Layer Values > Pixel Based > Circular StdDev

Calculates the standard deviation for all pixels within a ring around the center of an image object. The thickness of the ring is defined by two radius values, R_1 and R_2. Parameters and expressions are identical to the Circular Mean feature.

23.4.12 Circular Std Dev/Mean

Object Features > Layer Values > Pixel Based > Circular Std Dev/Mean

Calculates the ratio of the mean to the standard deviation for all pixels within a ring around the center of an image object. The thickness of the ring is defined by two radius values, R_1 and R_2. Parameters and expressions are identical to the Circular Mean feature.

23.4.13 Quantile

Object Features > Layer Values > Pixel Based > Quantile

Select the quantile of a layer.

Editable Parameters

- Select a layer from the Layer drop-down box, from which to extract the quantile
- Insert a percentage value into the Quantile field (or get the value from a feature or array item).

Parameters

The quantile is given by the element of the list Y with index equal to $int(r)$

- Y is the list of pixel intensities (sorted in increasing order)
- N is the number of pixels
- p is a number between 0 and 100 (i.e. a percentage)
- The rank, $r = \frac{p}{100} \times N$

Feature Value Range

$[\text{min layer value}, \text{ max layer value}]$

23.5 To Neighbors

Object Features > Layer Values > To Neighbors

23.5.1 Mean Diff. to Neighbors

Object Features > Layer Values > To Neighbors > Mean Diff. to Neighbors

The Mean Diff. to Neighbors feature describes the difference between an image object and its neighbor image objects, in terms of mean layer intensity values. For each neighboring image object of the image object of interest, the mean layer difference is computed and weighted as follows:

- If the neighboring image objects are direct neighbors (feature distance = 0), the length of the border between the image objects is used for weighting
- If the neighborhood of image objects is defined within a certain perimeter around the image object of interest (feature distance > 0), the area covered by the neighbor image objects is used for weighting.

Editable Parameters

- Image Layer
- Feature Distance: Radius of the perimeter in pixel/voxel. Direct neighbors have a value of 0.

Parameters

- u, v are image objects
- $b(v,u)$ is the length of the common border between v and u
- $\bar{c}_k$ is the mean intensity of image layer k
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- $\#P_u$ is the total number of pixels/voxels contained in P_v
- d is the distance between neighbors
- w is the image layer weight with $w = \sum_{u \in N_v(d)} w_u$
- w_u is the weight of image object u with $w_u = \{^{b(v,u), d=0)}_{\#P_u, d>0}$
- N_v is the direct neighbor to image object v with $N_v : \{u \in V_i : \exists(x,y) \in P_v \exists(x',y') \in P_u : (x'y') \in N_4(x,y)\}$
- $N_v(d)$ is a neighbor to v at a distance d with $N_v(d) = \{u \in V_i : d(v,u) \leq d\}$.

Expression

$$\bar{\Delta}_k(v) = \frac{1}{w} \sum_{u \in N_v(d)} w_u \big(\bar{c}_k(v) - \bar{c}_k(u)\big)$$

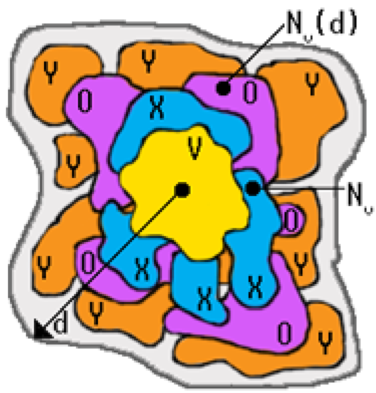


Figure 23.9. Direct neighbors (x) and the neighborhood within perimeter d of image object v

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

Conditions

If $w = 0$ the feature value is 0; therefore the formula is invalid.

23.5.2 Mean Diff. to Neighbors (Abs)

Object Features > Layer Values > To Neighbors > Mean Diff. to Neighbors (Abs.)

The Mean Diff. to Neighbors (Abs) feature describes the difference between an image object and its neighbor image objects, in terms of their mean layer intensity values. It is calculated in a similar way to the Mean Diff. to Neighbors feature; the only difference is that absolute values of differences are averaged.

Editable Parameters

- Image Layer
- Feature Distance – the radius of the perimeter in pixels. Direct neighbors have a value of 0.

Parameters

- u, v are image objects
- $b(v,u)$ is the length of the common border between v and u
- $\bar{c}_k$ is the mean intensity of image layer k
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- c_k^{range} is the data range of image layer k
- d is the distance between neighbors
- $\#P_u$ is the total number of pixels contained in P_u
- w is the image layer weight with $w = \sum_{u \in N_v(d)} w_u$
- w_u is the weight of image object u with $w_u = \{^{b(v,u), d=0)}_{\#P_u, d>0}$
- N_v is the direct neighbor to image object v with $N_v : \{u \in V_i : \exists(x,y) \in P_v \exists(x',y') \in P_u : (x'y') \in N_4(x,y)\}$
- $N_v(d)$ is a neighbor to v at a distance d with $N_v(d) = \{u \in V_i : d(v,u) \leq d\}$.

Expression

$$\bar{\Delta}_k(v) = \frac{1}{w} \sum_{u \in N_v(d)} w_u \big(|\bar{c}_k(v) - \bar{c}_k(u)| \big)$$

Feature Value Range

$[0, c_k^{\text{range}}]$

Conditions

If $w = 0$ the feature value is 0; therefore the formula is invalid.

23.5.3 Mean Diff. to Darker Neighbors

Object Features > Layer Values > To Neighbors > Mean Diff. to Darker Neighbors

The Mean Diff. to Darker Neighbors feature describes the difference between an image object and its darker neighbor image objects, in terms of their mean layer intensity values. In contrast to the Mean Diff. to Neighbors feature, only direct neighbor image objects are counted that have a lower mean layer value than the image object of interest.

Editable Parameters

Image Layer

Parameters

- u, v are image objects
- $b(v,u)$ is the length of the common border between v and u
- $\bar{c}_k$ is the mean intensity of image layer k
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- c_k^{range} is the data range of image layer k , with $c_k^{\text{range}} = c_k^{\max} - c_k^{\min}$
- w is the image layer weight with $w = \sum_{u \in N_v(d)} w_u$, where w_u is the weight of image object u with $w_u = \{ {}^{b(v,u), d=0}_{\#P_u, d>0}$
- N_v is the direct neighbor to image object v with $N_v : \{u \in V_i : \exists(x,y) \in P_v \exists(x',y') \in P_u : (x'y') \in N_4(x,y)\}$
- N_v^D is the darker direct neighbor to v , with $N_v^D \{u \in N_v : \bar{c}_k(u) < \bar{c}_k(v)\}$.

Expression

$$\bar{\Delta}_k^D(v) = \frac{1}{w} \sum_{u \in N_v^D} w_u \big(\bar{c}_k(v) - \bar{c}_k(u)\big)$$

Feature Value Range

$[-c_k^{\text{range}}, c_k^{\text{range}}]$

Conditions

If $w = 0$ the feature value is 0; therefore the formula is invalid. If $N_v^D = 0$ the formula is invalid.

23.5.4 Mean Diff. to Brighter Neighbors

Object Features > Layer Values > To Neighbors > Mean Diff. to Brighter Neighbors

The Mean Diff. to Brighter Neighbors feature describes the difference between an image object and its darker neighbor image objects, in terms of their mean layer intensity values.

In contrast to the Mean Diff. to Neighbors feature, only direct neighbor image objects are counted that have a mean layer value less than that of the image object of interest.

Editable Parameters

- Image Layer

Parameters

- u, v are image objects
- $b(v,u)$ is the length of the common border between v and u
- $\bar{c}_k$ is the mean intensity of image layer k
- $c_k^{\min}$ is the darkest possible intensity value of image layer k
- $c_k^{\max}$ is the brightest possible intensity value of image layer k
- c_k^{range} is the data range of image layer k , with $c_k^{\text{range}} = c_k^{\max} - c_k^{\min}$
- w is the image layer weight with $w = \sum_{u \in N_v(d)} w_u$, where w_u is the weight of image object u with $w_u = \{^{b(v,u),d=0}_{\#P_u,d>0}$
- N_v is the direct neighbor to image object v with $N_v : \{u \in V_i : \exists(x,y) \in P_v \exists(x',y') \in P_u : (x'y') \in N_4(x,y)\}$
- N_v^B is the brighter direct neighbor to v , with $N_v^B \{u \in N_v : \bar{c}_k(u) < \bar{c}_k(v)\}$.

Expression

$$\bar{\Delta}_k^B(v) = \frac{1}{w} \sum_{u \varepsilon N_v^B} w_u \big(\bar{c}_k(v) - \bar{c}_k(u)\big)$$

Feature Value Range

$[-c_k^{\text{range}}, c_k^{\text{range}}]$

23.5.5 Number of Brighter Objects

Object Features > Layer Values > To Neighbors > Number of Brighter Objects

The Number of Brighter Objects feature counts the neighboring objects – those objects with a common border – with a higher intensity. Neighboring objects with the same mean value are not counted.

23.5.6 Number of Darker Objects

Object Features > Layer Values > To Neighbors > Number of Darker Objects

The Number of Darker Objects feature counts the neighboring objects – those objects with a common border – with a lower intensity. Neighboring objects with the same mean value are not counted.

23.5.7 Rel. Border to Brighter Neighbors

Object Features > Layer Values > To Neighbors > Rel. Border to Brighter Neighbors

The Rel. Border to Brighter Neighbors feature describes the extent to which an image object is surrounded by brighter or darker direct neighbor image objects. In a given image layer, it is the ratio of the shared image border of an image object to the total border. A value of 1 that the image object is surrounded completely by brighter neighbors; a value of 0 means it has no brighter direct neighbors.

Editable Parameters

Image Layer

Parameters

- N_v^B is the darker direct neighbor to v , with $N_v^B\{u \in N_v : \bar{c}_k(u) < \bar{c}_k(v)\}$
- b_v is the image object border length
- $b(v,u)$ is the length of common border between v and u.

Expression

$$\sum_{u \in N_v^B} \frac{b(v,u)}{b_v}$$

Feature Value Range

$[0,1]$

23.6 To Superobject

Object Features > Layer Values > To Superobject

23.6.1 Mean Diff. to Superobject

Object Features > Layer Values > To Superobject > Mean Diff. to Superobject

The difference between the mean layer intensity value of an image object and the mean layer intensity value of its superobject.

Editable Parameters

- Image Layer
- Image Object Level Distance – the upward distance of image object levels in the image object hierarchy between the image object and the superobject.

Parameters

- $\bar{c}_k$ is the mean intensity of image layer k
- c_k^{range} is the data range of image layer k , with $c_k^{\text{range}} = c_k^{\max} - c_k^{\min}$
- $S_v(d)$ is the sub-object of image object v with level distance d
- $U_v(d)$ is the superobject of image object v with level distance d.

Expression

$\bar{c}_k(v) - \bar{c}_k\big(U_v(d)\big)$

Feature Value Range

$[-c_k^{\text{range}}, c_k^{\text{range}}]$

23.6.2 Ratio to Superobject

Object Features > Layer Values > To Superobject > Ratio to Superobject

The ratio of the mean layer intensity value of an image object and the mean layer intensity value of its superobject.

Editable Parameters

- Image Layer
- Image Object Level Distance – the upward distance of image object levels in the image object hierarchy between the image object and the superobject.

Parameters

- $\bar{c}_k$ is the mean intensity of image layer k
- $U_v(d)$ is the superobject of image object v with level distance d

Expression

$$\frac{\bar{c}_k(v)}{\bar{c}_k\big(U_v(d)\big)}$$

Feature Value Range

$[0, \infty]$

Conditions

If $U_v(d) = \varnothing$ or $U_v(d) = 0 \rightarrow \infty$ the expression is undefined.

23.6.3 Std. Dev. Diff. to Superobject

Object Features > Layer Values > To Superobject > Std. Dev. Diff to Superobject

Editable Parameters

- Image Layer
- Image Object Level Distance – the upward distance of image object levels in the image object hierarchy between the image object and the superobject.

Parameters

- $\sigma_k(v)$ is the standard deviation of intensity values of image layer k of all pixel/voxels forming an image object v
- $U_v(d)$ is the superobject of image object v with level distance d
- c_k^{range} is the data range of image layer k with $c_k^{\text{range}} = c_k^{\max} - c_k^{\min}$.

Expression

$\sigma_k(v) - \sigma_k\big(U_v(d)\big)$

Feature Value Range

$[-\frac{1}{2}c_k^{\text{range}}, \frac{1}{2}c_k^{\text{range}}]$

Condition

If $U_v(d) = \varnothing$ the expression is undefined.

23.6.4 Std. Dev. Ratio to Superobject

Object Features > Layer Values > To Superobject > Std. Dev Ratio to Superobject

The ratio of the standard deviation of the layer intensity of an image object to the standard deviation of the layer intensity of its superobject.

Editable Parameters

- Image Layer
- Image Object Level Distance – the upward distance of image object levels in the image object hierarchy between the image object and the superobject.

Parameters

- $\sigma_k(v)$ is the standard deviation of intensity values of image layer k of all pixel/voxels forming an image object v
- $U_v(d)$ is the superobject of image object v with level distance d

Expression

$$\frac{\sigma_k(v)}{\sigma_k(U_v(d))}$$

Feature Value Range

$[0, \infty]$

Conditions

- If $U_v(d) = \varnothing$ or $U_v(d) = 0 \to \infty$ the expression is undefined.
- If $\sigma_v(U_v(d)) = 0 \Rightarrow$ the standard deviation ratio to $(U_v(d)) = 1$.

23.7 To Scene

Object Features > Layer Values > To Scene

23.7.1 Mean Diff. to Scene

Object Features > Layer Values > To Scene > Mean Diff. to Scene

Editable Parameters

- Image Layer

Parameters

- $\bar{c}_k(v)$ is the mean intensity of image layer k of all pixels forming an image object v
- $\bar{c}_k$ is the mean intensity of image layer k
- c_k^{range} is the data range of image layer k , with $c_k^{\text{range}} = c_k^{\text{max}} - c_k^{\text{min}}$

Expression

$\bar{c}_k(v) - \bar{c}_k$

Feature Value Range

$[-c_k^{\text{range}}, c_k^{\text{range}}]$

23.7.2 Ratio To Scene

Object Features > Layer Values > To Scene > Ratio to Scene

Editable Parameter

- Image Layer

Parameters

- $\bar{c}_k(v)$ is the mean intensity of image layer k of all pixels forming an image object v
- $\bar{c}_k$ is the mean intensity of image layer k.

Expression

$$\frac{\bar{c}_k(v)}{\bar{c}_k}$$

Feature Value Range

$[-\infty, \infty]$

Condition

If $\bar{c}_k = 0 \Leftrightarrow$ the expression is undefined.

23.8 Hue, Saturation, Intensity

Object Features > Layer Values > Hue, Saturation, Intensity

Hue, Saturation, and Intensity features convert RGB color space values to HSI values.

23.8.1 HSI Transformation

Editable Parameters

- Layer red, layer green, layer blue – for each parameter, assign a corresponding image layer from the drop-down list. By default these are the first three image layers of the scene.
- Output: Select the type of HSI transformation feature to be created: hue (color), saturation or intensity (brightness).

Parameters

- RGB values are expressed as numbers from 0 to 1
- max is the greatest of the RGB values
- min is the smallest of the RGB values.

Expression

$$H = \begin{cases} \text{undefined} & \text{if max} = \text{min} \\ \frac{60^\circ \times \frac{G-B}{\text{max}-\text{min}}}{360^\circ} & \text{if max} = \text{R} \\ \frac{60^\circ \times \frac{B-R}{\text{max}-\text{min}} + 120^\circ}{360^\circ} & \text{if max} = \text{G} \\ \frac{60^\circ \times \frac{R-G}{\text{max}-\text{min}} + 240^\circ}{360^\circ} & \text{if max} = \text{B} \end{cases}$$

Feature Value Range

$[0,1]$

24 Object Features: Geometry

Object Features > Geometry

Geometry features are based on an image object's shape, calculated from the pixels that form it. Because images are raster-based, geometry features may be rotation variant: after image objects are rotated, different feature values may arise.

24.1 Extent

Object Features > Geometry > Extent

24.1.1 Area

Object Features > Geometry > Extent > Area

The number of pixels forming an image object. If unit information is available, the number of pixels can be converted into a measurement. In scenes that provide no unit information, the area of a single pixel is 1 and the area is simply the number of pixels that form it. If the image data provides unit information, the area can be multiplied using the appropriate factor.

Parameters

- A_v is the area of image object v
- $\#P_v$ is the total number of pixels contained in P_v
- u is the pixel size in co-ordinate system units. If the unit is a pixel, then $u = 1$.

Expression

$A_v = \#P_v \times u^2$

Feature Value Range

$[0, \text{scene size}]$

24.1.2 Border Length [for 2D Image Objects]

Object Features > Geometry > Extent > Border Length

The border length of an image object is defined as the sum of edges of the image object shared with other image objects, or situated on the edge of the entire scene. For a torus – and other image objects with holes – the border length is the sum of the inner and outer borders.

Parameters

- b_v is the border length of image object
- b_o is the length of outer border
- b_i is the length of inner border

Expression

$b_v = b_o + b_i$

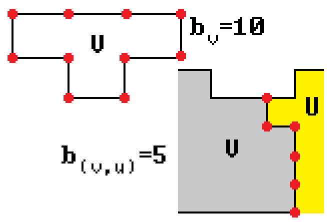


Figure 24.1. Border length of an image object v , or between two objects v and u

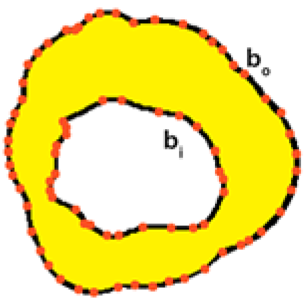


Figure 24.2. Inner and outer border length of a torus image object

Feature Value Range

$[0, \infty]$

24.1.3 Border Length [for 3D Image Objects]

Object Features > Geometry > Extent > Border Length

The border length of a 3D image object is the sum of border lengths of all image object slices, multiplied by the spatial distance between them.

Image object slices are 2D pieces of the image object in each slice. The border length of an image object slice is defined as the sum of edges shared with other image object pieces, or are situated on the edge of the entire slice. For a torus and other image objects with holes the border length sums the inner and outer border.

Parameters

- b_v is the border length of image object v
- $b_v(\text{slice})$ is the border length of image object slice
- $b_v(z)$ is the border length of image object in z -direction
- u_{slices} is the spatial distance between slices in the co-ordinate system unit
- b_o is the length of the outer border
- b_i is the length of the inner border

Expression

$$b_v = \left(\sum_{n=1}^{\#(\text{slices})} b_v(\text{slice}) \right) \times u_{\text{slices}} + b_v(z)$$

where $b_v(\text{slice}) = b_o + b_i$

Feature Value Range

$[0, \infty]$

24.1.4 Length [for 2D Image Objects]

Object Features > Geometry > Extent > Length

The length of a 2D image object is calculated using the length-to-width ratio.

Parameters

- $\#P_v$ is the total number of pixels contained in P_v
- γ_v is the length-width ratio of an image object v

Expression

$$\sqrt{\#P_v \cdot \gamma_v}$$

Feature Value Range

$[0, \infty]$

24.1.5 Length [for 3D Image Objects]

Object Features > Geometry > Extent > Length

The length of an image object is the largest of three eigenvalues of a rectangular 3D space that is defined by the same volume, and same proportions of eigenvalues, as the image object. The length of an image object can be smaller or equal than the largest of dimensions of the smallest rectangular 3D space enclosing the image object.

Feature Value Range

$[0, \infty]$

24.1.6 Length/Thickness

Object Features > Geometry > Extent > Length/Thickness

The length-to-thickness ratio of an image object.

Parameters

- Length of the image object
- Thickness of the image object

Expression

$$\frac{\text{Length}}{\text{Thickness}}$$

Feature Value Range

$[0, \infty]$

24.1.7 Length/Width [for 2D Image Objects]

Object Features > Geometry > Extent > Length/Width

The length-to-width ratio of an image object. There are two methods to approximate this:

1. The ratio of length to width is identical to the ratio of the eigenvalues of the covariance matrix, with the larger eigenvalue being the numerator of the fraction:

$$\gamma_v^{EV} = \frac{\lambda_1(v)}{\lambda_2(v)}$$

2. The ratio of length to width can also be approximated using the bounding box:

$$\gamma_v^{BB} = \frac{\left(k_v^{bb'}\right)^2}{\#P_v}$$

Both calculations are compared; the smaller of both results is returned as the feature value.

Parameters

- $\#P_v$ is the Size of a set of pixels of an image object v
- $\lambda_1 \lambda_2$ are eigenvalues
- γ_v^{EV} is the ratio length of v of the eigenvalues
- γ_v^{BB} is the ratio length of v of the bounding box
- γ_v is the length-width ratio of an image object v
- $k_v^{\mathrm{BB}'}$
- $h_v^{\mathrm{BB}'}$
- a is the bounding box fill rate
- $\#Pxl$
- h
- w is the image layer weight
- $k_v^{bb^1} = \sqrt{(k_v^{bb'})^2 + (1-a)(h_v^{bb'})^2}$
- $a = \frac{\#P_v}{k_v^{bb'} - h_v^{bb'}}$
- $k \cdot h = \#P_v \Rightarrow k = \frac{\#P_v}{w}, h = \frac{\#P_v}{k} \Rightarrow \frac{k}{w} = \frac{k^2}{\#P_{xl}} = \frac{\#P_{xl}}{w^2}$

Expression

$$\gamma_v = \min \gamma_v^{EV}, \max \gamma_v^{BB}$$

Feature Value Range

$[0, \infty]$

24.1.8 Length/Width [for 3D Image Objects]

Object Features > Geometry > Extent > Length/Width

The length-to-width ratio of an image object.

Parameters

- Length of the image object
- Width of the image object

Expression

$$\frac{\text{Length}}{\text{Width}}$$

Feature Value Range

$[0, \infty]$

24.1.9 Number of Pixels

Object Features > Geometry > Extent > Number of Pixels

The number of pixels forming an image object. Unit information is not taken into account.

Parameters

- $\#P_v$ is the total number of pixels contained in P_v

Feature Value Range

$[0, \text{scene size}]$

24.1.10 Thickness

Object Features > Geometry > Extent > Thickness

The thickness of an image object is the smallest of three eigenvalues of a rectangular 3D space with the same volume as the image object and the same proportions of eigenvalues as the image object. The thickness of an image object can be smaller or equal to the smallest of dimensions of the smallest rectangular 3D space enclosing the image object.

Feature Value Range

$[0, \infty]$

24.1.11 Volume

Object Features > Geometry > Extent > Volume

The number of voxels forming an image object rescaled by using unit information for x and y co-ordinates and distance information between slices.

Parameters

- V_v is the volume of image object v
- $\#P_v$ is the total number of voxels contained in P_v
- u is the size of a slice pixel in the co-ordinate system unit
- u_{slices} is the spatial distance between slices in the co-ordinate system unit

Expression

$V_v = \#P_v \times u^2 \times u_{\text{slices}}$

Feature Value Range

$[0, \text{scene size}]$

24.1.12 Width [for 2D Image Objects]

Object Features > Geometry > Extent > Width

The width of an image object is calculated using the length-to-width ratio.

Parameters

- $\#P_v$ is the total number of pixels contained in P_v
- γ_v is the length/width ratio of an image object v

Expression

$\frac{\#P_v}{\gamma_v}$

Feature Value Range

$[0, \infty]$

24.1.13 Width [for 3D Image Objects]

Object Features > Geometry > Extent > Width

The width of an image object is the mid-point of three eigenvalues of a rectangular 3D space with the same volume as the image object and the same proportions of eigenvalues as the image object. The width of an image object can be smaller or equal to the mid-point of dimensions of the smallest rectangular 3D space enclosing the image object.

Feature Value Range

$[0, \infty]$

24.2 Shape

Object Features > Geometry > Shape

24.2.1 Asymmetry [for 2D Image Objects]

Object Features > Geometry > Shape > Asymmetry

The Asymmetry feature describes the relative length of an image object, compared to a regular polygon. An ellipse is approximated around a given image object, which can be expressed by the ratio of the lengths of its minor and the major axes. The feature value increases with this asymmetry.

> **NOTE**: We recommend that you use the length/width ratio because it is more accurate.

Parameters

- Var X is the variance of X
- Var Y is the variance of Y

Expression

$$\frac{2\sqrt{\frac{1}{4}(\mathrm{Var}X+\mathrm{Var}Y)^2+(\mathrm{Var}XY)^2-\mathrm{Var}X\cdot\mathrm{Var}Y}}{\mathrm{Var}X+\mathrm{Var}Y}$$

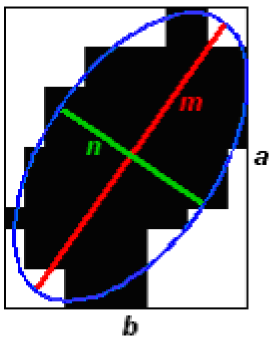


Figure 24.3. Asymmetry for a 2D image object

Feature Value Range

$[0,1]$

24.2.2 Asymmetry [for 3D Image Objects]

Object Features > Geometry >Shape > Asymmetry

The Asymmetry feature describes the relative length of an image object, in the same manner as 2D image objects. The asymmetry is calculated from the ratio between the smallest and largest eigenvalues of the image object.

Parameters

- λ_{min} is the minimal eigenvalue
- λ_{max} is the maximal eigenvalue

Expression

$$1 - \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}$$

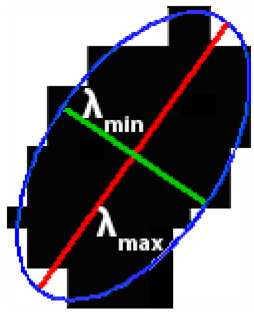


Figure 24.4. Asymmetry based on maximal and minimal eigenvalues

Feature Value Range

$[0, 1]$

24.2.3 Border Index

Object Features > Geometry > Shape > Border Index

The Border Index feature describes how jagged an image object is; the more jagged, the higher its border index. This feature is similar to the Shape Index feature, but the Border Index feature uses a rectangular approximation instead of a square. The smallest rectangle enclosing the image object is created and the border index is calculated as the ratio between the border lengths of the image object and the smallest enclosing rectangle.

Parameters

- b_v is the image object border length
- l_v is the length of an image object v
- w_v is the width of an image object v

Expression

$$\frac{b_v}{2(l_v + w_v)}$$

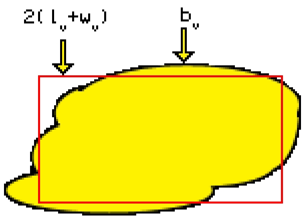


Figure 24.5. Border index of an image object v

Feature Value Range

$[1, \infty]$; 1 = ideal.

24.2.4 Compactness [for 2D Image Objects]

Object Features > Geometry > Shape > Compactness

The Compactness feature describes how compact an image object is. It is similar to Border Index, but is based on area. However, the more compact an image object is, the smaller its border appears. The compactness of an image object is the product of the length and the width, divided by the number of pixels.

Parameters

- l_v is the length of an image object v
- w_v is the width of an image object v
- $\#P_v$ is the total number of pixels contained in P_v

Expression

$$\frac{2\lambda_1 \times 2\lambda_2 \times 2\lambda_3}{V_v}$$

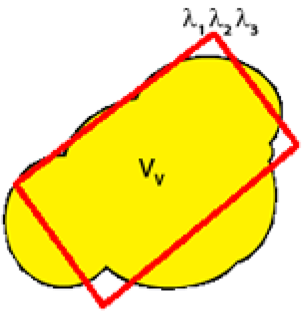


Figure 24.6. Compactness of an image object v

Feature Value Range

$[0,\infty]$; 1 = ideal.

24.2.5 Compactness [for 3D Image Objects]

Object Features > Geometry >Shape > Compactness

Compactness describes how compact a 3D image object is. Appropriately scaled eigenvectors of an image object's covariance matrix provide a rough figure of the object's extent in three dimensions.

A figure for the compactness of a 3D image object is calculated by a scaled product of its three eigenvalues $(2\times\lambda_1, 2\times\lambda_2, 2\times\lambda_3)$ divided by the number of its pixel/voxel. We include a factor of 2 with each eigenvalue, since $\lambda_i\times$ eigenvectors represent otherwise half axes of an ellipsoid defined by its covariance matrix. The chosen approach therefore provides an estimate of a cuboid occupied by the object.

Parameters

- λ_1 is eigenvalue 1 of a 3D image object v
- λ_2 is the eigenvalue 2 of a 3D image object v
- λ_3 is eigenvalue 3 of a 3D image object v
- V_v is the volume of image object v

Expression

$$\frac{2\lambda_1 \times 2\lambda_2 \times 2\lambda_3}{V_v}$$

Feature Value Range

$[0,\infty]$; 1 = ideal.

24.2.6 Density [for 2D Image Objects]

Object Features > Geometry > Shape > Density

The Density feature describes the distribution in space of the pixels of an image object. In Definiens Developer XD 2.0.4 the most "dense" shape is a square; the more an object is shaped like a filament, the lower its density. The density is calculated by the number of pixels forming the image object divided by its approximated radius, based on the covariance matrix.

Parameters

- $\sqrt{\#P_v}$ is the diameter of a square object with $\#P_v$ pixels
- $\sqrt{VarX+VarY}$ is the diameter of the ellipse

Expression

$$\frac{\sqrt{\#P_v}}{1+\sqrt{VarX+VarY}}$$

Feature Value Range

$[0, \text{depending on shape of image object}]$

24.2.7 Density [for 3D Image Objects]

Object Features > Geometry > Shape > Density

Using the same principle as Density [for 2D Image Objects], the most "dense" shape for a 3D object is a cube. The more filament-shaped an image object is, the lower its density. The value is calculated by dividing the edge of the volume of a fitted cuboid by the radius of a fitted sphere.

Parameters

- V_v is the volume of image object v
- $\sqrt[3]{V_v}$ is the edge of the volume fitted cuboid
- $\sqrt{\mathrm{Var}(X)+\mathrm{Var}(Y)+\mathrm{Var}(Z)}$ is the radius of the fitted sphere

Expression

$$\frac{\sqrt[3]{V_v}}{\sqrt{\mathrm{Var}(X)+\mathrm{Var}(Y)+\mathrm{Var}(Z)}}$$

Feature Value Range

$[0, \text{depending on shape of image object}]$

24.2.8 Elliptic Fit [for 2D Image Objects]

Object Features > Geometry > Shape > Elliptic Fit

The Elliptic Fit feature describes how well an image object fits into an ellipse of similar size and proportions. While 0 indicates no fit, 1 indicates a perfect fit.

The calculation is based on an ellipse with the same area as the selected image object. The proportions of the ellipse are equal to the length to the width of the image object. The area of the image object outside the ellipse is compared with the area inside the ellipse that is not filled by the image object.

Parameters

- $\varepsilon_v(x,y)$ is the elliptic distance at a pixel (x,y)
- P_v is the set of pixels of an image object v
- $\#P_v$ is the total number of pixels contained in P_v

Expression

$$\varphi = 2 \cdot \frac{\#\{(x,y)\varepsilon P_v : \varepsilon_v(x,y) \leq 1\}}{\#P_v} - 1$$

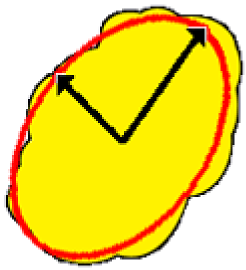


Figure 24.7. Elliptic fit of an image object v

Feature Value Range

$[0,1]$; 1 = complete fitting, 0 = <50% fit.

24.2.9 Elliptic Fit [for 3D Image Objects]

Object Features > Geometry > Shape > Elliptic Fit

The Elliptic Fit feature describes how well an image object fits into an ellipsoid of similar size and proportions. While 0 indicates no fit, 1 indicates a perfect fit.

The calculation is based on an ellipsoid with the same volume as the considered image object. The proportions of the ellipsoid are equal to the proportions of the length, width and thickness of the image object. The volume of the image object outside the ellipsoid is compared with the volume inside the ellipsoid that is not filled out with the image object.

Parameters

- $\varepsilon_v(x,y,z)$ is the elliptic distance at a pixel (x,y,z)
- P_v is the set of pixels of an image object v
- $\#P_v$ is the total number of pixels contained in P_v

Expression

$$\varphi = 2 \cdot \frac{\#\{(x,y,z)\varepsilon P_v : \varepsilon_v(x,y,z) \leq 1\}}{\#P_v} - 1$$

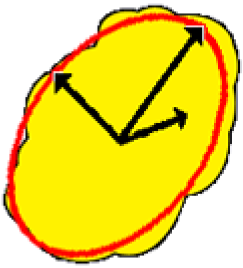


Figure 24.8. Elliptic fit of an image object v

Feature Value Range

$[0,1]$; 1 = complete fitting, whereas 0 = < 50% voxels.

24.2.10 Main Direction [for 2D Image Objects]

Object Features > Geometry > Shape > Elliptic Fit > Main Direction

The Main Direction feature of an image object is defined as the direction of the eigenvector belonging to the larger of the two eigenvalues, derived from the covariance matrix of the spatial distribution of the image object.

Parameters

- $VarX$ is the variance of X
- $VarY$ is the variance of Y
- λ_1 is the eigenvalue

Expression

$$\frac{180^\circ}{\pi}\tan^{-1}(\mathrm{Var}XY, \lambda_1 - \mathrm{Var}Y) + 90^\circ$$

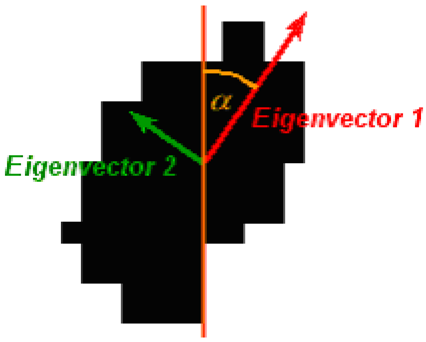


Figure 24.9. The main direction is based on the direction of the larger eigenvector

Feature Value Range

$[0, 180]$

24.2.11 Main Direction [for 3D Image Objects]

Object Features > Geometry > Shape > Main Direction

The Main Direction feature of a three-dimensional image object is computed as follows:

1. For each (2D) image object slice, the centers of gravities are calculated.
2. The co-ordinates of all centers of gravities a reused to calculate a line of best fit, according to the Weighted Least Square method.
3. The angle a between the resulting line of best fit and the z-axis is returned as feature value.

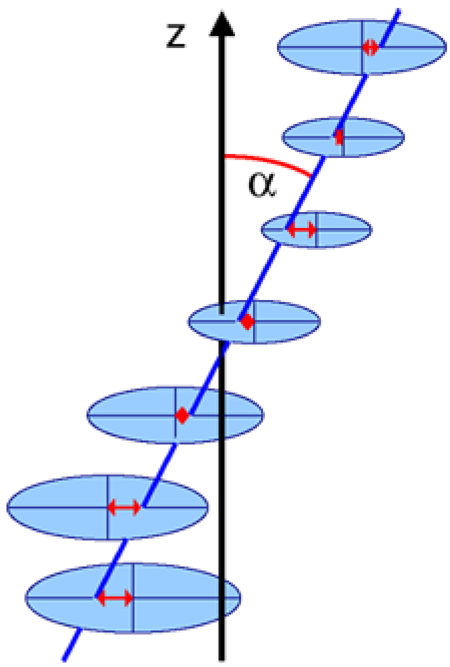


Figure 24.10. The line of best fit (blue) calculated from centers of gravity of image object slices (light blue)

Feature Value Range

$[0, 90]$

24.2.12 Radius of Largest Enclosed Ellipse [for 2D Image Objects]

Object Features > Geometry > Shape > Radius of Largest Enclosed Ellipse

The Radius of Largest Enclosed Ellipse feature describes how similar an image object is to an ellipse. The calculation uses an ellipse with the same area as the object and based on the covariance matrix. This ellipse is scaled down until it is totally enclosed by the image object. The ratio of the radius of this largest enclosed ellipse to the radius of the original ellipse is returned as feature value.

Parameters

- $\varepsilon_v(x,y)$ is the elliptic distance at a pixel (x,y)

Expression

$\varepsilon_v(x_o,y_o)$, where $(x_o,y_o) = min\ \varepsilon_v(x,y), (x,y) \notin P_v$

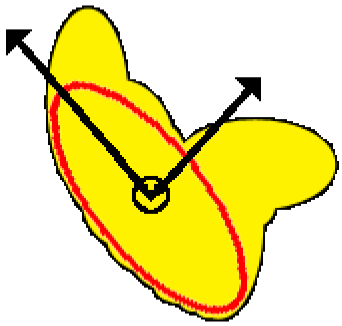


Figure 24.11. Radius of the largest enclosed ellipse of in image object v , for a 2D image object

Feature Value Range

$[0,\infty]$

24.2.13 Radius of Largest Enclosed Ellipse [for 3D Image Objects]

Object Features > Geometry > Shape > Radius of Largest Enclosed Ellipse

The Radius of Largest Enclosed Ellipse feature describes how much the shape of an image object is similar to an ellipsoid. The calculation is based on an ellipsoid with the same volume as the object and based on the covariance matrix. This ellipsoid is scaled down until it is totally enclosed by the image object. The ratio of the radius of this largest enclosed ellipsoid to the radius of the original ellipsoid is returned as feature value.

Parameters

- $\varepsilon_v(x,y,z)$ is the elliptic distance at a pixel (x,y,z)

Expression

$\varepsilon_v(x_o,y_o,z_o)$, where $(x_o,y_o,z_o) = min\ \varepsilon_v(x,y,z), (x,y,z) \notin P_v$

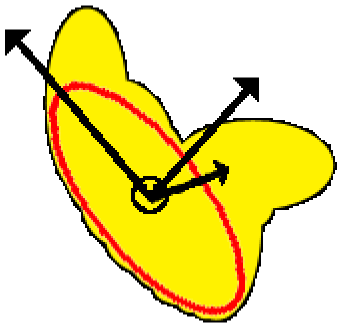


Figure 24.12. Radius of the largest enclosed ellipse of in image object v , for a 3D image object

Feature Value Range

$[0, \infty]$

24.2.14 Radius of Smallest Enclosing Ellipse [for 2D Image Objects]

Object Features > Geometry >Shape > Radius of Smallest Enclosing Ellipse

The Radius of Smallest Enclosing Ellipse feature describes how much the shape of an image object is similar to an ellipse. The calculation is based on an ellipse with the same area as the image object and based on the covariance matrix. This ellipse is enlarged until it encloses the image object in total. The ratio of the radius of this smallest enclosing ellipse to the radius of the original ellipse is returned as feature value.

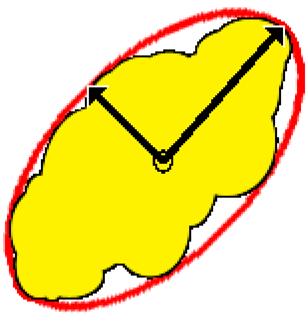


Figure 24.13. Radius of the smallest enclosed ellipse of in image object v , for a 2D image object

Parameters

- $\varepsilon_v(x,y)$ is the elliptic distance at a pixel (x,y)

Expression

$\varepsilon_v(x_o, y_o)$, where $(x_o, y_o) = max\ \varepsilon_v(x,y), (x,y) \notin P_v$

Feature Value Range

$[0, \infty]$

24.2.15 Radius of Smallest Enclosing Ellipse [for 3D Image Objects]

Object Features > Geometry > Shape > Radius of Smallest Enclosing Ellipse

The Radius of Smallest Enclosing Ellipse feature describes how much the shape of an image object is similar to an ellipsoid. The calculation is based on an ellipsoid with the same volume as the image object and based on the covariance matrix. This ellipsoid is enlarged until it encloses the image object in total. The ratio of the radius of this smallest enclosing ellipsoid to the radius of the original ellipsoid is returned as feature value.

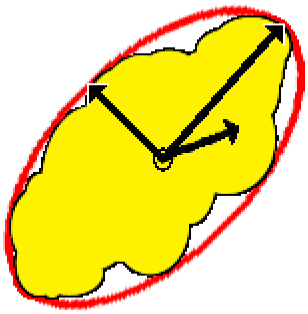


Figure 24.14. Radius of the smallest enclosed ellipse of in image object v , for a 3D image object

Parameters

- $\varepsilon_v(x, y, z)$ is the elliptic distance at a pixel (x, y, z)

Expression

$\varepsilon_v(x_o, y_o, z_o)$, where $(x_o, y_o, z_o) = max\ \varepsilon_v(x, y, z), (x, y, z) \in \sigma P_v$

Feature Value Range

$[0, \infty]$

24.2.16 Rectangular Fit [for 2D Image Objects]

Object Features > Geometry > Shape > Rectangular Fit

The Rectangular Fit feature describes how well an image object fits into a rectangle of similar size and proportions. While 0 indicates no fit, 1 indicates for a complete fitting image object.

The calculation is based on a rectangle with the same area as the image object. The proportions of the rectangle are equal to the proportions of the length to width of the image object. The area of the image object outside the rectangle is compared with the area inside the rectangle.

Parameters

- $\rho_v(x,y)$ is the elliptic distance at a pixel (x,y)

Expression

$$\frac{\left\{\#(x,y)\varepsilon P_v : \rho_v(x,y) \leq 1\right\}}{\#P_v}$$

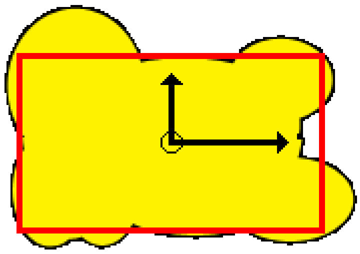


Figure 24.15. Rectangular fit of an image object v

Feature Value Range

$[0,1]$; where 1 is a perfect rectangle.

24.2.17 Rectangular Fit [for 3D Image Objects]

Object Features > Geometry > Shape > Rectangular Fit

The Rectangular Fit feature describes how well an image object fits into a cuboid of similar size and proportions. While 0 indicates no fit, 1 indicates a perfect fit.

The calculation is based on a cuboid with the same volume as the considered image object. The proportions of the cuboid are equal to the proportions of the length to width to thickness of the image object. The volume of the image object outside the rectangle is compared with the volume inside the cuboid that is not filled out with the image object.

Parameters

- $\rho_v(x,y,z)$ is the rectangular distance at a pixel (x,y,z)
- $\#P_v$ is the total number of pixels contained in P_v

Expression

$$\frac{\{\#(x,y,z)\varepsilon P_v : \rho_v(x,y,z) \leq 1\}}{\#P_v}$$

Feature Value Range

$[0,1]$; 1 = complete fitting, whereas 0 = 0% fits inside the rectangular approximation.

24.2.18 Roundness [for 2D Image Objects]

Object Features > Geometry > Shape > Roundness

The Roundness feature describes how similar an image object is to an ellipse. It is calculated by the difference of the enclosing ellipse and the enclosed ellipse. The radius of the largest enclosed ellipse is subtracted from the radius of the smallest enclosing ellipse.

Parameters

- ε_v^{max} is the radius of the smallest enclosing ellipse
- ε_v^{min} is the radius of the largest enclosed ellipse

Expression

$\varepsilon_v^{max} - \varepsilon_v^{min}$

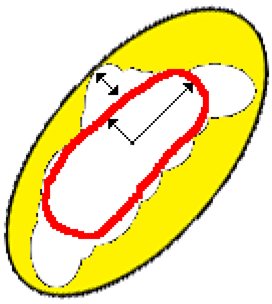


Figure 24.16. Roundness feature of image object v

Feature Value Range

$[0,\infty]$; 0 = ideal.

24.2.19 Roundness [for 3D Image Objects]

Object Features > Geometry > Shape > Roundness

The Roundness feature describes how much the shape of an image object is similar to an ellipsoid. The more the shape of an image object is similar to an ellipsoid, the lower its roundness.

It is calculated by the difference of the enclosing ellipsoid and the enclosed ellipsoid. The radius of the largest enclosed ellipsoid is subtracted from the radius of the smallest enclosing ellipsoid.

Parameters

- $\varepsilon_v^{\max}$ is the radius of the smallest enclosing ellipsoid
- $\varepsilon_v^{\min}$ is the radius of the largest enclosed ellipsoid

Expression

$$\varepsilon_v^{\max} - \varepsilon_v^{\min}$$

Feature Value Range

$[0, \infty]$; 0 = ideal.

24.2.20 Shape Index [for 2D Image Objects]

Object Features > Geometry > Shape > Shape Index

The Shape index describes the smoothness of an image object border. The smoother the border of an image object is, the lower its shape index. It is calculated from the Border Length feature of the image object divided by four times the square root of its area.

Parameters

- b_v is the image object border length
- $4\sqrt{\#P_v}$ is the border of square with area $\#P_v$

Expression

$$\frac{b_v}{4\sqrt{\#P_v}}$$

Feature Value Range

$[1, \infty]$; 1 = ideal.

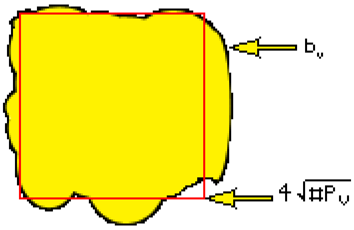


Figure 24.17. Shape index of an image object v

24.2.21 Shape Index [for 3D Image Objects]

Object Features > Geometry > Shape > Shape Index

The Shape Index describes the smoothness of the surface of an image object. The smoother the surface of an image object is, the lower its shape index. It is calculated from the Border Length feature of the image object divided by the volume of the image object.

Parameters

- b_v is the image object border length
- V_v is the volume of image object v

Expression

$$\frac{b_v}{V_v}$$

Feature Value Range

$[0,1]$

24.3 To Super-Object

Object Features > Geometry > To Super-Object

Use the To Super-Object feature to describe an image object by its shape and relationship to one of its superobjects, where appropriate. Editing the feature distance determines which superobject is referred to. When working with thematic layers these features can be of interest.

24.3.1 Rel. Area to Super-Object

Object Features > Geometry > To Super-Object > Rel. Area to Super-Object

The area of an image object divided by the area of its superobject. If the feature value is 1, the image object is identical to its superobject. Use this feature to describe an image object in terms of the amount of area it shares with its superobject.

Parameters

- $\#P_v$ is the total number of pixels contained in P_v
- $\#P_{Uv(d)}$ is the the size of the superobject of v in the image object level of the level distance d

Expression

$$\frac{\#P_v}{\#P_{Uv(d)}}$$

Conditions

If $U_v(d) = \varnothing \therefore$ the formula is undefined.

Feature Value Range

$[0,1]$

24.3.2 Rel. Rad. Position to Super-Object

Object Features > Geometry > To Super-Object > Rel. Rad. Position to Super-Object

This value is calculated by dividing the distance between the center of a selected image object and the center of its superobject, by the distance of the center of the most distant image object (which has the same superobject). Use this feature to describe an image object by its position relative to the center of its superobject.

Parameters

- $\#P_v$ is the total number of pixels contained in P_v
- $\#P_{Uv(d)}$ is the the size of the superobject of an image object v
- $d_g(v, Uv(d))$ is the distance of v to the center of gravity of the superobject $U_v(d)$

Expression

$$\frac{d_g\left(v, U_v(d)\right)}{\max\limits_{u \in S_{Uv(d)}(d)} d_g\left(u, U_v(d)\right)}$$

Conditions

If $U_v(d) = \varnothing \therefore$ the formula is undefined.

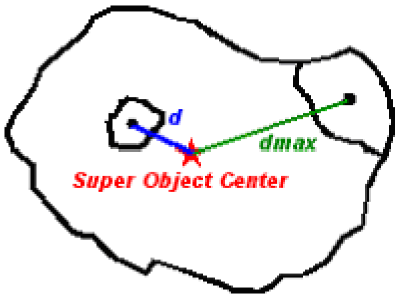


Figure 24.18. Relative radial position to superobject

Feature Value Range

$[0,1]$

24.3.3 Rel. Inner Border to Super-Object

Object Features > Geometry > To Super-Object > Rel. Inner Border to Super-Object

This feature is calculated by dividing the sum of the border shared with other image objects, which have the same superobject, by the total border of the image object. If the relative inner border to the superobject is 1, the image object of concern is not situated on the border of its superobject. Use this feature to describe how much of an image object is situated at the edge of its superobject.

Parameters

- $N_u(v)$ are neighbors of v that exist within the superobject $N_u(v) : \{u \in N_v : U_u(d) - U_v(d)\}$
- b_v is the image object border length

Expression

$$\frac{\sum_{u \in N_U(v)} b(v,m)}{b_v}$$

Conditions

If the feature range is 0 then $v = U_v(d)$

If the feature range is 1 then v is an inner object.

Feature Value Range

$[0,1]$

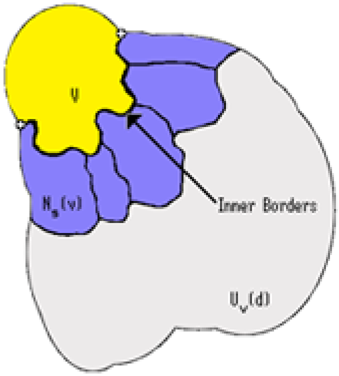
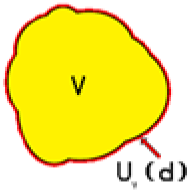


Figure 24.19. Relative inner border of an image object v to superobject u



24.3.4 Distance to Super-Object Center

Object Features > Geometry > To Super-Object > Distance to Super-Object Center

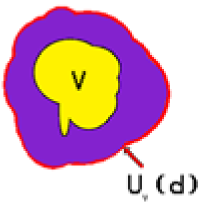
The distance of an image object's center to the center of its superobject. The calculated distance is based upon segmentation on the image object level of the object of interest, within the boundaries of the superobject.

Expression

$d_g(v, U_v(d))$ is the distance of v to the center of gravity of the superobject $U_v(d)$

Feature Value Range

$[0, sx \times sy]$



24.3.5 Elliptic Distance to Super-Object Center

Object Features > Geometry > To Super-Object > Elliptic Distance to Super-Object Center

Distance of objects to the center of the superobject.

Expression

$d_e(v, U_v(d))$

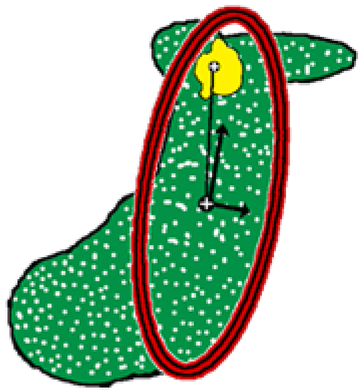


Figure 24.20. Distance between the distance from the superobject's center to the center of a sub-object

Feature Value Range

Typically $[0, 5]$

24.3.6 Is End of Super-Object

Object Features > Geometry > To Super-Object > Is End of Super-Object

This feature is true for two image objects a and b if following conditions are true:

- a and b are sub-objects of the same superobject.
- a is the image object with the maximum distance to the superobject.
- b is the image object with the maximum distance to a .

Editable Parameter

- Level Distance

Feature Value Range

$[0, 1]$

24.3.7 Is Center of Super-Object

Object Features > Geometry > To Super-Object > Is Center of Super-Object

This feature is true if the image object is the center of its superobject.

Editable Parameter

- Level Distance

Feature Value Range

$[0,1]$

24.3.8 Rel. X Position to Super-Object

Object Features > Geometry > To Super-Object Rel. X Position to Super-Object

This feature returns the relative x position of an image object with regard to its superobject, based on the centers of gravity of both objects.

Editable Parameters

- Level Distance – the upward distance of image object levels in the image object hierarchy between the image object and the superobject.

Expression

$\Delta x = x_{CG}$ of current image object $-x_{CG}$ of superobject (where x_{CG} is the center of gravity)

Feature Value Range

$-\frac{\text{scene width}}{2} + \frac{\text{scene width}}{2}$

24.3.9 Rel. Y Position to Super-Object

Object Features > Geometry > To Super-Object > Rel. Y Position to Super-Object

This feature returns the relative y position of an image object with regard to its superobject, based on the centers of gravity of both objects.

Editable Parameter

- Level Distance: Upward distance of image object levels in the image object hierarchy between the image object and the superobject.

Expression

$\Delta y = y_{CG}$ of current image object $-y_{CG}$ of superobject (where y_{CG} is the center of gravity)

Feature Value Range

$$-\frac{\text{scene height}}{2} + \frac{\text{scene height}}{2}$$

24.4 Based on Polygons

Object Features > Geometry > Based on Polygons

The polygon features provided by Definiens Developer XD 2.0.4 are based on the vectorization of the pixels that form an image object.

Figure 24.21. Raster image object (black area) with its polygon object (red lines) after vectorization

24.4.1 Edges Longer Than

Object Features > Geometry > Based on Polygons > Edges Longer Than

Editable Parameters

- Minimum Length

24.4.2 Number of Right Angles With Edges Longer Than

Object Features > Geometry > Based on Polygons > Number of Right Angles with Edges Longer Than

The number of right angles that have at least one side edge longer than a given threshold.

Editable Parameters

- Minimum length

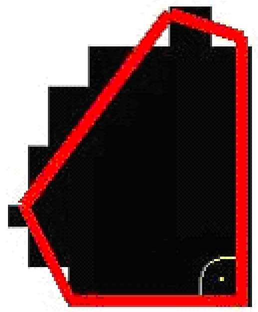


Figure 24.22. A polygon with one rectangular angle

24.4.3 Area (Excluding Inner Polygons)

Object Features > Geometry > Based on Polygons > Area (Excluding Inner Polygons)

The Area (Excluding Inner Polygons) feature calculates the area of a polygon based on Green's Theorem in a plane. In contrast to the Area (Including Inner Polygons) feature, the feature value does not include the areas of any existing inner polygons.

Parameters

- $(x_i, y_i), i = 0, \ldots, n$, with $x_o = x_n$ and $y_o = y_n$ as the given points
- $a_i = x_i y_{i+1} - x_{i+1} y_i$

Expression

$$\frac{1}{2}\sum_{i=0}^{n-1} a_i$$

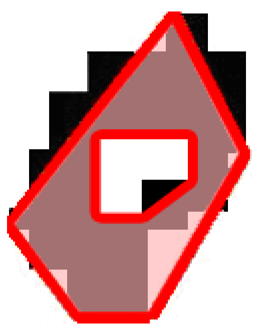


Figure 24.23. A polygon with an inner polygon that is not included in the feature value

Feature Value Range

$[0, \text{scene size}]$

24.4.4 Area (Including Inner Polygons)

Object Features > Geometry > Based on Polygons > Area (Including Inner Polygons)

The Area (Excluding Inner Polygons) feature calculates the area of a polygon based on Green's Theorem in a plane. Different to the Area (Excluding Inner Polygons) feature, the feature value includes the areas of any existing inner polygons (for instance the single polygon formed in the center of a donut-shaped object).

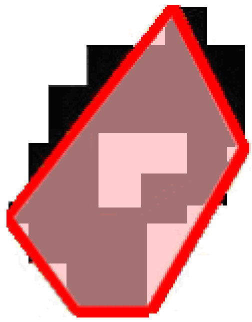


Figure 24.24. A polygon with an inner polygon that is included in the feature value

24.4.5 Average Length of Edges (Polygon)

Object Features > Geometry > Based on Polygons > average Length of Edges (Polygon)

The average length of all edges in a polygon.

Parameters

- X_i is the length of edge i
- n is the total number of edges

Expression

$$\text{Average} = \frac{\sum_{i=1}^{n} X_i}{n}$$

24.4.6 Compactness (Polygon)

Object Features > Geometry > Based on Polygons > Compactness (Polygon)

The ratio of the area of a polygon to the area of a circle with the same perimeter.

Parameters

- Area
- Perimeter

Expression

$$\frac{4 \times \pi \times \text{Area}}{\text{Perimeter}^2}$$

Feature Value Range

$[0, 1 \text{ for a circle}]$

24.4.7 Length of Longest Edge (Polygon)

Object Features > Geometry > Based on Polygons > Length of Longest Edge (Polygon)

The length of the longest edge of a polygon.

24.4.8 Number of Edges (Polygon)

Object Features > Geometry > Based on Polygons > Number of Edges (Polygon)

The number of edges of a polygon.

24.4.9 Number of Inner Objects (Polygon)

Object Features > Geometry > Based on Polygons > Number of Inner Objects (Polygon)

The number of inner polygons that are completely surrounded by a selected outer polygon.

24.4.10 Perimeter (Polygon)

Object Features > Geometry > Based on Polygons > Perimeter (Polygon)

The sum of the lengths of all the edges of a polygon.

24.4.11 Polygon Self-Intersection (Polygon)

Object Features > Geometry > Based on Polygons > Polygon Self-Intersection (Polygon)

The Polygon Self-Intersection (Polygon) feature allows the identification of a rare arrangement of image objects, leading to a polygon self-intersection when exported as a polygon vector file.

This feature enables you to identify the affected image objects and take measures To avoid this self-intersection. All image objects with a value of 1 will cause a polygon self-intersection when exported to a shapefile.

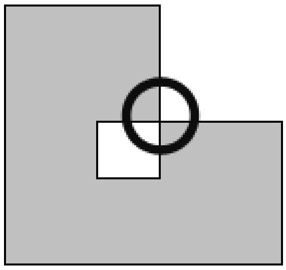


Figure 24.25. This type of image object leads to a self-intersection at the circled point

To avoid the self-intersection, the enclosed image object needs to be merged with the enclosing image object.

> **TIP**: Use the Image Object Fusion algorithm to remove polygon intersections. To do so, set the domain to all image objects with a value larger than 0 for the polygon intersection feature. In the algorithm parameter, set the Fitting Function Threshold to Polygon Self-Intersection (Polygon) feature to zero and in the Weighted Sum group, set Target Value Factor to 1. This will merge all image objects with a value of 1 for the Polygon Self-Intersection (Polygon) feature, so that the resulting image object will not include a self-intersection.

Feature Value Range

$[0, 1]$

24.4.12 Std. Dev. of Length of Edges (Polygon)

Object Features > Geometry > Based on Polygons > Stddev of Length of Edges (Polygon)

The Std. Dev. of Length of Edges (Polygon) feature measures how the lengths of edges deviate from their mean value.

Parameters

- x_i is the length of edge i
- $\bar{X}$ is the mean value of all lengths
- n is the total number of edges

Expression

$$\sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$$

24.5 Based on Skeletons

Object Features > Geometry > Based on Skeletons

For the better understanding of the following descriptions, the skeleton is structured in a main line and subordinate branches. A node is a mid-point of the triangles created by the Delaunay triangulation.

24.5.1 Number of Segments of Order

Object Features > Geometry > Based on Skeletons > Number of Segments of Order

The number of line segments of branches of an object that are of a given order. Note that only segments that do not belong to a lower order are counted

Editable Parameter

- Branch order: The main line of the skeleton has the order 0.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.2 Number of Branches of Order

Object Features > Geometry > Based on Skeletons > Number of Branches of Order

The number of branches of an object that are of a given order.

Editable Parameter

- Branch order: The main line of the skeleton has the order 0

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.3 Average Length of Branches of Order

Object Features > Geometry > Based on Skeletons > average Length of Branches of Order

The average length of branches of an object that are of a given order. The length of the branch of the selected order is measured from the intersect point of the whole branch and the main line to the end of the branch.

Editable Parameter

- Branch order: The main line of the skeleton has the order 0.

Feature Value Range

[0, depending on shape of objects]

24.5.4 Number of Branches of Length

Object Features > Geometry > Based on Skeletons > Number of Branches of Length

The number of branches of an object that are of a special length up to a selected order. All ends of branches are counted up to the selected order.

Editable Parameter

- Branch order: The main line of the skeleton has the order 0.
- Minimum length
- Maximum length

Feature Value Range

[0; depending on shape of objects]

24.5.5 Average Branch Length

Object Features > Geometry > Based on Skeletons > Average Branch Length

The average length of all branches of an object.

Feature Value Range

[0, depending on shape of objects]

24.5.6 Avrg. Area Represented by Segments

Object Features > Geometry > Based on Skeletons > Avrg. Area Represented by Segments

The average area of all triangles created by a Delaunay triangulation.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.7 Curvature/Length (Only Main Line)

Object Features > Geometry > Based on Skeletons > Curvature/Length (only main line)

The length-to-curvature ratio of the main line of an object. The curvature is the sum of all changes in direction of the main line. Changes in direction are expressed by the acute angle a in which sections of the main line, built by the connection between the nodes, cross each other.

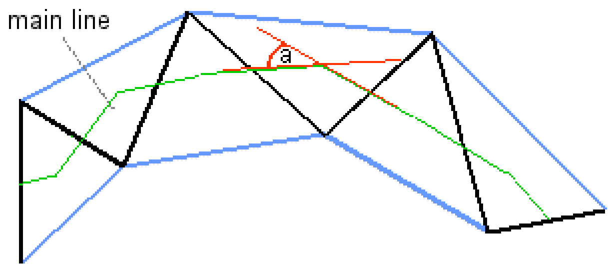


Figure 24.26. The main line (green) connects the mid-points of triagles (black and blue) created by a Delaunay triangulation of the objects' shape (not depicted)

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.8 Degree of Skeleton Branching

Object Features > Geometry > Based on Skeletons > Degree of Skeleton Branching

The highest order of branching in an object.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.9 Length of Main Line (No Cycles)

Object Features > Geometry > Based on Skeletons > Length of Main Line (No Cycles)

The sum of all distances between the nodes of the main line of an object. If an object contains an island polygon – a polygon derived from the inner borders of an image object – it is ignored and the main line may cross it (no cycles). This is different to the Length of Main Line (regarding cycles) feature where the main line goes around the island polygon. This feature does not visualize skeletons.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.10 Length of Main Line (Regarding Cycles)

Object Features > Geometry > Based on Skeletons > Length of Main Line (regarding cycles)

The sum of all distances between the nodes of the main line of an object. If an object contains an island polygon – a polygon derived from the inner borders of an image object – the main line is calculated so as not to cross it (Regarding Cycles). In contrast to the Length of Main Line (No Cycles) feature, the skeletons are visualized.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.11 Length/Width (Only Main Line)

Object Features > Geometry > Based on Skeletons >Length/Width (Only Main Line)

The length-to-width ratio of the main line of an object.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.12 Maximum Branch Length

Object Features > Geometry > Based on Skeletons > Maximum Branch Length

The length of the longest branch of an object. It is measured from the intersect point of the branch and the main line to the end of the branch.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.13 Number of Segments

Object Features > Geometry > Based on Skeletons > Number of Segments

The number of all segments of the main line and the branches of an object.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.14 Stddev Curvature (Only Main Line)

Object Features > Geometry > Based on Skeletons > Stddev Curvature (Only Main Line)

The standard deviation of the curvature is the result of the standard deviation of the changes in direction of the main line. Changes in direction are expressed by the acute angle in which sections of the mainline, built by the connection between the nodes, cross each other.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.15 Stddev of Area Represented by Segments

Object Features > Geometry > Based on Skeletons > Stddev. of Area Represented by Segments

The standard deviation of all triangles created by the Delaunay triangulation.

Feature Value Range

$[0, \text{depending on shape of objects}]$

24.5.16 Width (Only Main Line)

Object Features > Geometry > Based on Skeletons > Width (Only Main Line)

The width of an object based on the height of triangles created by a Delaunay triangulation. It is calculated by the average height h of all triangles crossed by the main line.

An exception is triangles where the height h does not cross one of the sides of the triangle. In this case, the nearest side s is used to define the height.

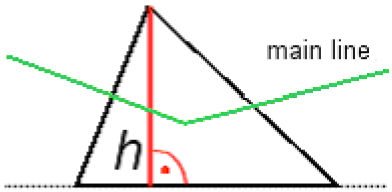


Figure 24.27. Height h of an triangle that is crossed by the main line

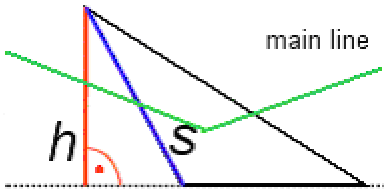


Figure 24.28. Height of an triangle that is crossed by the main line. In this case the side s defines the height

Feature Value Range

$[0, \text{depending on shape of objects}]$

25 Object Features: Position

Object Features > Position

Position features refer to the position of an image object relative to the entire scene. These features are of special interest when working with geographically referenced data, as an image object can be described by its geographic position. Position features refer to the pixel co-ordinate definition.

25.1 Distance

Object Features > Position > Distance

25.1.1 Distance to Line

Object Features > Position > Distance > Distance to Line

The distance between the center of gravity of a two-dimensional image object and a given line. The line is defined manually by entering two points that are a part of this line. Note that the line has neither a start nor an end.

Editable Parameters

To set, right-click the Distance to Line feature, select Edit Feature and adapt the co-ordinates of the two points:

- First Co-ordinate (X)
- First Co-ordinate (Y)
- Second Co-ordinate (X)
- Second Co-ordinate (Y)

Feature Value Range

$[0, \infty]$

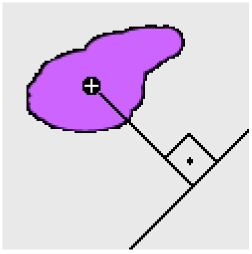


Figure 25.1. Distance between an image object and a line

25.1.2 Distance to Scene Border

Object Features > Position > Distance > Distance to Scene Border

The distance of an image object slice (a 2D piece of the image object in a slice) to the nearest border of the scene within the current slice. (The Z-direction is ignored.)

Parameters

- min x is the minimum distance from the scene border at x -axis
- max x is the maximum distance from the scene border at x -axis
- min y is the minimum distance from the scene border at y -axis
- max y is the maximum distance from the scene border at y -axis
- (sx, sy) is the scene size

Expression

$\min\{\min x, sx - \max x, \min y, sy - \max y\}$

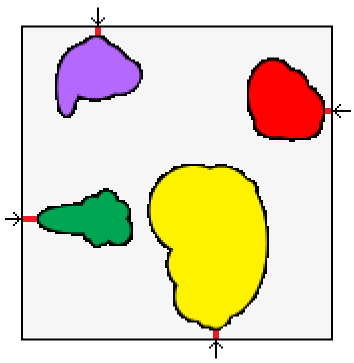


Figure 25.2. Examples of the distance between an image object and the nearest scene border

Feature Value Range

$[0, \max sx - 1, sy - 1]$

25.1.3 T Distance to First Frame (Pxl)

Object Features > Position > Distance > T Distance to First Frame (Pxl)

The distance of an image object to the first frame of the scene.

25.1.4 T Distance to Last Frame

Object Features > Position > Distance > T Distance to Last Frame

The distance of an image object to the last frame of the scene.

25.1.5 X Distance to Scene Left Border

Object Features > Position > Distance > X Distance to Scene Left Border

Horizontal distance of an image object slice (a 2D piece of the image object in a slice) to the left border of the scene within the current slice.

Parameters

- *sx* is the scene size at the *x* -axis
- *min x* is the minimum distance from the scene border at *x* -axis

Expression

$$\min_{(x,y)\in P_v} \mathrm{x}$$

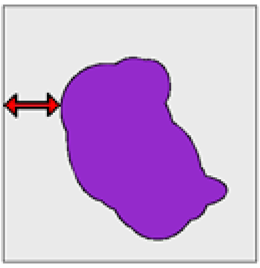


Figure 25.3. X-distance between the image object and the left border

Feature Value Range

$[0, sx-1]$

25.1.6 X Distance to Scene Right Border

Object Features > Position > Distance > X Distance to Scene Right Border

Horizontal distance of an image object slice (a 2D piece of the image object in a slice) to the right border of the scene within the current slice.

Parameters

- *sx* is the scene size at the *x* -axis
- *max x* is the maximum distance from the scene border at *x* -axis

Expression

$$\mathrm{sx} - \max_{(x,y)\in P_v} \mathrm{x}$$

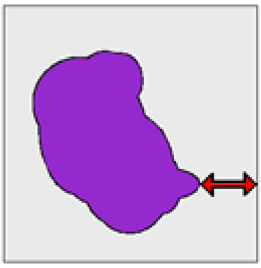


Figure 25.4. X-distance to the image object right border

Feature Value Range

$[0, sx-1]$

25.1.7 Y Distance to Scene Bottom Border

Object Features > Position > Distance > Y Distance to Scene Bottom Border

Vertical distance of an image object slice (a 2D piece of the image object in a slice) to the bottom border of the scene within the current slice.

Parameters

- *sy* is the scene size
- *min y* is the minimum distance from the scene border at *y* -axis

Expression

$$\min_{(x,y)\in P_v} \mathrm{y}$$

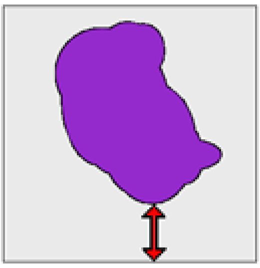


Figure 25.5. Y-distance between the image object and the bottom border

Feature Value Range

$[0, sy - 1]$

25.1.8 Y Distance to Scene Top Border

Object Features > Position > Distance > Y Distance to Scene Top Border

Vertical distance of an image object slice (a 2D piece of the image object in a slice) to the top border of the scene within the current slice.

Parameters

- *sy* is the scene size
- *max y* is the maximum distance from the scene border at *y* -axis

Expression

$$\mathrm{sy} - \max_{(x,y) \in P_v} \mathrm{y}$$

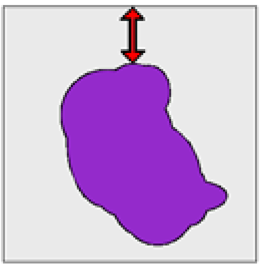


Figure 25.6. Y-distance between the image object and the top border of the scene

Feature Value Range

$[0, sy - 1]$

25.1.9 Z Distance to First Slice (Pxl)

Object Features > Position > Distance > Z Distance to First Slice (Pxl)

Distance of an image object to the first slice of the scene.

25.1.10 Z Distance to Last Slice (Pxl)

Object Features > Position > Distance > Z Distance to Last Slice (Pxl)

Distance of an image object to the last slice of the scene.

25.2 Co-ordinate

Object Features > Position > Co-ordinate

25.2.1 Hilbert Index

Object Features > Position > Co-ordinate > Hilbert Index

Generates a *Hilbert Curve*[1] within an object; for a given curve, the feature returns the position on the Hilbert curve on which an image object lies.

Parameters

- Pxl – enter a value for the resolution of the grid in pixels

25.2.2 Is at Active Pixel

Object Features > Position > Co-ordinate > Is at Active Pixel

This feature returns 1 if the current active pixel is within the image object, otherwise 0.

25.2.3 Time (Pxl)

Object Features > Position > Co-ordinate > Time

T-position of the center of an image object. The calculation is based on the center of gravity (geometric center) of the image object in the internal map.

Parameters

- $\bar{t}_v$ is the t -center of an image object v
- $\bar{x}_v$ is the x -center of an image object v in an internal map with $\bar{x}_v(\mathrm{map}) = \frac{1}{\#P_v} \times \sum_{(x_{\mathrm{map}}, y_{\mathrm{map}}) \in P_v} x_{\mathrm{map}}$
- sx_{frame} is the extent in x of each slice and frame

1. en.wikipedia.org/wiki/Hilbert_curve

Expression

$$\bar{t}_v = \text{floor}\left(\frac{\bar{x}_v(\text{map})}{sx_{\text{frame}}}\right)$$

Feature Value Range

$[0.5, \text{number of frames in map} - 0.5]$

25.2.4 Time Max (Pxl)

Object Features > Position > Co-ordinate > Time Max (Pxl)

Maximum t -position of an image object derived from its bounding box. The calculation is based on the maximum t -position of the image object in the internal map.

Parameters

- $t_{\max}(v)$ is the maximum t -position of an image object v
- $x_{\max}(v, \text{m}ap)$ is the maximum x -position of an image object v in an internal map
- sx_{frame} is the extent in x of each slice and frame

Expression

$$\bar{t}_v = \text{floor}\left(\frac{\bar{x}_v(\text{map})}{sx_{\text{frame}}}\right)$$

Feature Value Range

$[1, \text{number of frames in map}]$

25.2.5 Time Min (Pxl)

Object Features > Position > Co-ordinate > Time Min (Pxl)

Minimum t -position of an image object derived from its bounding box. The calculation is based on the minimum t -position of the image object in the internal map.

Parameters

- $t_{\min}(v)$ is the minimum t -position of an image object v
- $x_{\min}(v, \text{m}ap)$ is the minimum x -position of an image object t in an internal map
- sx_{frame} is the extent in x of each slice and frame

Expression

$t_{min}(v) = \text{floor}\left(\frac{x_{min}(v,\text{map})}{sx_{\text{frame}}}\right)$

Feature Value Range

$[0.5, \text{number of frames in map} - 1]$

25.2.6 X Center

Object Features > Position > Co-ordinate > X Center

X-position of the center of an image object. The calculation is based on the center of gravity (geometric center) of the image object in the internal map.

Parameters

- $\bar{x}_v$ is the x -center of an image object v
- $\bar{x}_v(\text{map})$ is the x -center of an image object v in an internal map with $\bar{x}_v(\text{map}) = \frac{1}{\#P_v} \times \sum_{(x_{\text{map}},y_{\text{map}})\in P_v} x_{\text{map}}$
- $\#P_v$ is the total number of pixels contained in P_v
- $(x_{\text{map}}, y_{\text{map}})$ are the co-ordinates in an internal map
- sx_{frame} is the extent in x of each slice and frame

Expression

$\bar{x}_v = \bar{x}_v(map) - \text{floor}\left(\frac{\bar{x}_v(\text{map})}{sx_{\text{frame}}}\right) \times sx_{\text{frame}}$

Feature Value Range

$[\text{scene extent in } x, 0.5]$

25.2.7 X Max.

Object Features > Position > Co-ordinate > X Max

Maximum x -position of an image object derived from its bounding box. The calculation is based on the maximum x -position of the image object in the internal map.

Parameters

- $x_{\max}(v)$ is the minimum x -position of an image object v
- $x_{\max}(v, \text{map})$ is the maximum x -position of an image object t in an internal map
- sx_{frame} is the extent in x of each slice and frame

Expression

$$x_{max}(v) = x_{max}(v, \text{map}) - \text{floor}\left(\frac{\bar{x}_{\text{max}}(\text{map})}{sx_{\text{frame}}}\right) \times sx_{\text{frame}}$$

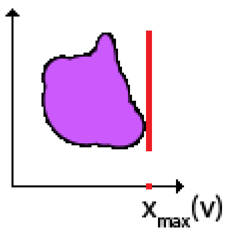


Figure 25.7. Maximum value of the x-coordinate at the image object border

Feature Value Range

$[1, \text{scene extent in } x]$

25.2.8 X Min.

Object Features > Position > Co-ordinate > X Min

Minimum x-position of an image object derived from its bounding box. The calculation is based on the minimum x-position of the image object in the internal map.

Parameters

- $x_{\text{min}}(v)$ is the minimum x -position of an image object v
- $x_{\text{min}}(v, \text{map})$ is the minimum x -position of an image object v in an internal map
- sx_{frame} is the extent in x of each slice and frame

Expression

$$x_{min}(v) = x_{min}(v, \text{map}) - \text{floor}\left(\frac{x_{min}(v, \text{map})}{sx_{\text{frame}}}\right) \times sx_{\text{frame}}$$

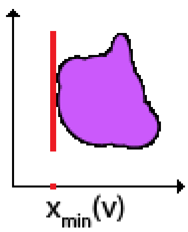


Figure 25.8. Minimum value of the x-coordinate at the image object border

Feature Value Range

$[0, \text{scene extent in } x - 1]$

25.2.9 Y Center

Object Features > Position > Co-ordinate > Y Center

Y-position of the center of an image object. The calculation is based on the center of gravity (geometric center) of the image object in the internal map.

Parameters

- $\bar{y}_v$ is the y -center of an image object v
- $\bar{y}_v(\text{map})$ is the y -center of an image object v in an internal map with $\bar{y}_v(\text{map}) = \frac{1}{\#P_v} \times \sum_{(x_{\text{map}}, y_{\text{map}}) \in P_v} y_{\text{map}}$
- $\#P_v$ is the total number of pixels contained in P_v
- $(x_{\text{map}}, y_{\text{map}})$ are the co-ordinates in an internal map
- sy_{slice} is the extent in y of each slice and frame

Expression

$$\bar{y}_v = \bar{y}_v(\text{map}) - \text{floor}\left(\frac{\bar{y}_v(\text{map})}{sy_{\text{slice}}}\right) \times sy_{\text{slice}}$$

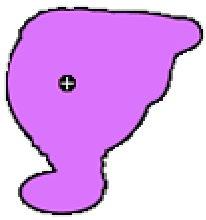


Figure 25.9. Center of gravity of an image object

Feature Value Range

$[0.5, \text{scene extent in } x - 0.5]$

25.2.10 Y Max.

Object Features > Position > Co-ordinate > Y Max

Maximum y-position of an image object derived from its bounding box. The calculation is based on the maximum y-position of the image object in the internal map.

Parameters

- $y_{\max}(v)$ is the maximum y -position of an image object v
- $y_{\max}(v,\mathrm{map})$ is the minimum y -position of an image object v in an internal map
- sy_{slice} is the extent in y of each slice and frame

Expression

$$y_{\max}(v) = \bar{y}_{max}(v,\mathrm{map}) - \mathrm{floor}\left(\frac{y_{\max}(v,\mathrm{map})}{sy_{\mathrm{slice}}}\right) \times sy_{\mathrm{slice}}$$

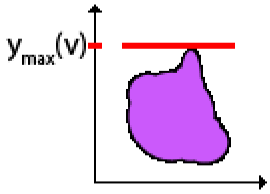


Figure 25.10. Maximum value of the y-coordinate at the image object border

Feature Value Range

$[1, \text{scene extent in } y]$

25.2.11 Y Min.

Object Features > Position > Co-ordinate > Y Min

Minimum y-position of an image object derived from its bounding box. The calculation is based on the minimum y-position of the image object in the internal map.

Parameters

- $y_{\min}(v)$ is the minimum y -position of an image object v
- $y_{\min}(v,\mathrm{map})$ is the minimum y -position of an image object v in an internal map
- sy_{slice} is the extent in y of each slice and frame

Expression

$$y_{\min}(v) = \bar{y}_{min}(v,\mathrm{map}) - \mathrm{floor}\left(\frac{y_{\min}(v,\mathrm{map})}{sy_{\mathrm{slice}}}\right) \times sy_{\mathrm{slice}}$$

Feature Value Range

$[0, \text{scene extent in } y - 1]$

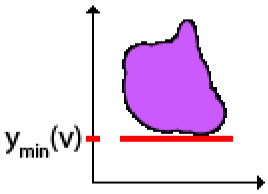


Figure 25.11. Minimum value of the y-coordinate at the image object border

25.2.12 Z Center

Object Features > Position > Co-ordinate > Z Center

Z -position of the center of an image object. The calculation is based on the center of gravity (geometric center) of the image object in the internal map.

Parameters

- $\bar{z}_v$ is the z -center of an image object v
- $\bar{y}_v(\text{map})$ is the y -center of an image object v in an internal map with $\bar{y}_v(\text{map}) = \frac{1}{\#P_v} \times \sum_{(x_{\text{map}}, y_{\text{map}}) \in P_v} y_{\text{map}}$
- sy_{slice} is the extent in y of each slice and frame

Expression

$$\bar{z}_v = \text{floor}\left(\frac{\bar{y}_v(\text{map})}{sy_{\text{slice}}}\right)$$

Feature Value Range

$[0.5, \text{number of slices in map} - 0.5]$

25.2.13 Z Max

Object Features > Position > Co-ordinate > Z Max

Maximum z -position of an image object derived from its bounding box. The calculation is based on the maximum z -position of the image object in the internal map.

Parameters

- $z_{\max}(v)$ is the maximum z -position of an image object v
- $y_{\max}(v, \text{map})$ is the minimum y -position of an image object v in an internal map
- sy_{slice} is the extent in y of each slice and frame

Expression

$$z_{\max}(v) = \text{floor}\left(\frac{y_{\max}(v,\text{map})}{sy_{\text{slice}}}\right)$$

Feature Value Range

$[1, \text{number of slices in map}]$

25.2.14 Z Min

Object Features > Position > Co-ordinate > Z Min

Minimum z -position of an image object derived from its bounding box. The calculation is based on the minimum z -position of the image object in the internal map.

Parameters

- $z_{\min}(v)$ is the maximum z -position of an image object v
- $y_{\min}(v, \text{map})$ is the minimum y -position of an image object v in an internal map
- sy_{slice} is the extent in y of each slice and frame

Expression

$$z_{\min}(v) = \text{floor}\left(\frac{y_{\min}(v,\text{map})}{sy_{\text{slice}}}\right)$$

Feature Value Range

$[0, \text{number of slices in map} - 1]$

25.3 Is Object in Region

Object Features > Position > Is Object in Region

The Is Object In Region feature checks if an image object is located in a given region. If this is true, the feature value is 1 (= true), otherwise it is 0 (= false).

25.3.1 Editable Parameters

- Region

25.3.2 Feature Value Range

$[0, 1]$

26 Object Features: Texture

Object Features > Texture

Texture features are used to evaluate the texture of image objects and include features based on an analysis of sub-objects helpful for evaluating highly textured data. In addition, features based upon the gray level co-occurrence matrix after Haralick are available. Features are structured in the following groups:

- Texture concerning the spectral information of the sub-objects
- Texture concerning the form of the sub-objects
- Texture after Haralick based on the gray level co-occurrence matrix (GLCM), which is a tabulation of how often different combinations of pixel gray levels occur in an image.

26.1 Layer Value Texture Based on Sub-objects

Object Features > Texture > Layer Value Texture Based on Sub-objects

Layer Value Texture Based on Sub-objects features refer to the spectral information provided by the image layers. To refer to an image object level of sub-objects, you can edit the level distance.

26.1.1 Mean of Sub-objects: Std. Dev.

Object Features > Texture > Layer Value Texture Based on Sub-objects > Mean of Sub-objects: Stddev.

Standard deviation of the different layer mean values of the sub-objects. This feature might appear very similar to the simple standard deviation computed from the single pixel values (layer values); however it can be more meaningful because – assuming an adequate segmentation – the standard deviation is computed over homogeneous and meaningful areas.

The smaller the sub-objects, the more the feature value approaches the standard deviation calculated from single pixels.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d

- $\bar{c}_k(u)$ is the mean intensity of layer k of an image object u
- d is the image object level distance

Expression

$$\sqrt{\frac{1}{\#S_v(d)}\left(\sum_{u\varepsilon S_v(d)}(\bar{c}_k(u))^2-\frac{1}{\#S_v\,(d)}\sum_{u\varepsilon S_v(d)}\bar{c}_k(u)\sum_{u\varepsilon S_v(d)}\bar{c}_k(u)\right)}$$

Feature Value Range

$[0,\text{depending on bit depth of data}]$

Conditions

If $S_v(d)=\varnothing \therefore$ the expression is invalid.

26.1.2 Avrg. Mean Diff. to Neighbors of Sub-objects

Object Features > Texture > Layer Value Texture Based on Sub-objects > Avrg. Mean Diff. to Neighbors of Sub-objects

The contrast inside an image object expressed by the average mean difference of all its sub-objects for a specific layer.

This feature has a certain spatial reference, because a local contrast inside the area covered by the image object is described. For each single sub-object the layer L mean difference (absolute values) to adjacent sub-objects of the same superobject is calculated. The feature value is the mean value of the layer L mean differences.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $\bar{\delta}_k(u)$ is the mean difference to neighbor of layer k of an image object u
- d is the image obejct level distance

Expression

$$\frac{1}{\#S_v(d)}\varepsilon_{u\varepsilon S_v(d)}\bar{\Delta}_k(u)$$

Feature Value Range

$[0,\text{depending on bit depth of data}]$

Conditions

If $S_v(d) = \varnothing \therefore$ the formula is invalid

26.2 Shape Texture Based on Sub-objects

Object Features > Texture > Shape Texture Based on Sub-objects

Shape Texture Based on Sub-objects features refer to the shape of sub-objects. To use these features successfully we recommend an accurate segmentation of the map to ensure sub-objects are as meaningful as possible. To refer to an image object level of sub-objects, you can edit the level distance.

26.2.1 Area of Sub-objects: Mean

Object Features > Texture > Shape Texture Based on Sub-objects > Area of Sub-objects: Mean

Mean value of the areas of the sub-objects.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $\#P_u$ is the total number of pixels contained in u
- d is the image object level distance

Expression

$$\frac{1}{\#S_v(d)} \sum_{\varepsilon S_v(d)} \#P_u$$

Feature Value Range

$[0, \text{scene size}]$

Condition

If $S_v(d) = \varnothing \therefore$ the formula is invalid.

26.2.2 Area of Sub-objects: Std. Dev.

Object Features > Texture > Shape Texture Based on Sub-objects > Area of Sub-objects: Std Dev.

Standard deviation of the areas of the sub-objects.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- P_u is the total number of pixels contained in u
- d is the image object level distance

Expression

$$\sqrt{\frac{1}{\#S_v(d)}\left(\sum_{u \varepsilon S_v(d)} \#P_u^2 - \frac{1}{\#S_v(d)} \sum_{u \varepsilon S_v(d)} \#P_u \sum_{u \varepsilon S_v(d)} \#P_u\right)}$$

Feature Value Range

$[0, \text{scene size}]$

Condition

If $S_v(d) = \varnothing \therefore$ the formula is invalid.

26.2.3 Density of Sub-objects: Mean

Object Features > Texture > Shape Texture Based on Sub-objects > Density of Sub-objects: Mean

Mean value calculated from the densities of the sub-objects.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $a(u)$ is the density of u
- d is the image object level distance

Expression

$$\frac{1}{\#S_v(d)} \sum_{u \varepsilon S_v(d)} a(u)$$

Feature Value Range

$[0, \text{depending on image object shape}]$

Condition

If $S_v(d) = \varnothing \therefore$ the formula is invalid.

26.2.4 Density of Sub-objects: Std. Dev.

Object Features > Texture > Shape Texture Based on Sub-objects > Density of Sub-objects: Stddev.

Standard deviation calculated from the densities of the sub-objects.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $a(u)$ is the density of u
- d is the image object level distance

Expression

$$\sqrt{\frac{1}{\#S_v(d)}\left(\sum_{u\varepsilon S_v(d)} a^2(u) - \frac{1}{\#S_v(d)} \sum_{u\varepsilon S_v(d)} a(u) \sum_{u\varepsilon S_v(d)} a(u)\right)}$$

Feature Value Range

$[0, \text{depending on image object shape}]$

Condition

If $S_v(d) = \varnothing$ $\therefore$ the formula is invalid.

26.2.5 Asymmetry of Sub-objects: Mean

Object Features > Texture > Shape Texture Based on Sub-objects > Asymmetry of Sub-objects: Mean

Mean value of the asymmetries of the sub-objects.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $a(u)$ is the asymmetry of u
- d is the image object level distance

Expression

$$\frac{1}{\#S_v(d)} \sum_{u\varepsilon S_v(d)} a(u)$$

Feature Value Range

$[0, \text{depending on image object shape}]$

Condition

If $S_v(d) = \varnothing \therefore$ the formula is invalid.

26.2.6 Asymmetry of Sub-objects: Std. Dev.

Object Features > Texture > Shape Texture Based on Sub-objects > Asymmetry of Sub-objects: Stddev.

Standard deviation of the asymmetries of the sub-objects.

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $a(u)$ is the asymmetry of u
- d is the image object level distance

Expression

$$\frac{1}{\#S_v(d)} \sum_{u \varepsilon S_v(d)} a(u)$$

Feature Value Range

$[0, \text{depending on image object shape}]$

Condition

If $S_v(d) = \varnothing \therefore$ the formula is invalid.

26.2.7 Direction of Sub-objects: Mean

Object Features > Texture > Shape Texture Based on Sub-objects > Direction of Sub-objects: Mean

The mean value of the directions of the sub-objects. In the computation, the directions are weighted with the asymmetry of the respective sub-objects. The more asymmetric an image object, the more significant its main direction.

Before computing the actual feature value, the algorithm compares the variance of all sub-object main directions with the variance of the sub-object main directions, where all directions between 90° and 180° are inverted. The set of sub-object main directions which has the lower variance is selected for the calculation of the main direction mean value, weighted by the sub-object asymmetries.

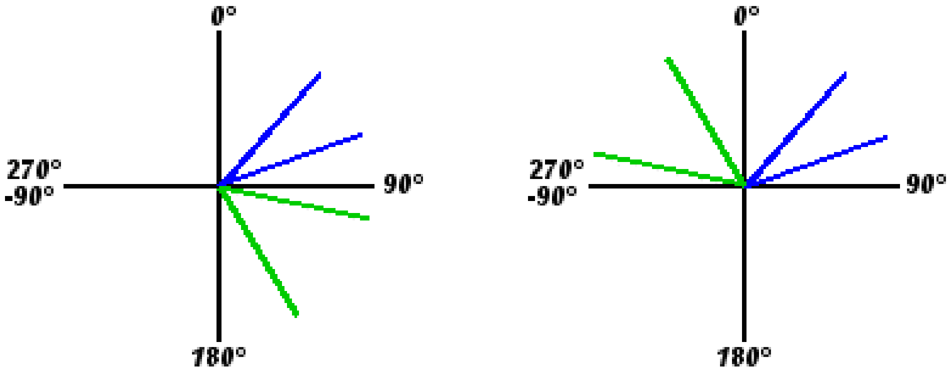


Figure 26.1. For calculation, the directions between 90° and 180° are inverted, which means the direction is reduced by 180°

Parameters

- $S_v(d)$ is the sub-object of an image object v at distance d
- $a(u)$ is the main direction of u
- d is the image object level distance

Expression

$$\frac{1}{\#S_v(d)} \sum_{u \varepsilon S_v(d)} a(u)$$

Feature Value Range

$[0°, 180°]$

Condition

If $S_v(d) = \varnothing \therefore$ the formula is invalid.

26.2.8 Direction of Sub-objects: Std. Dev.

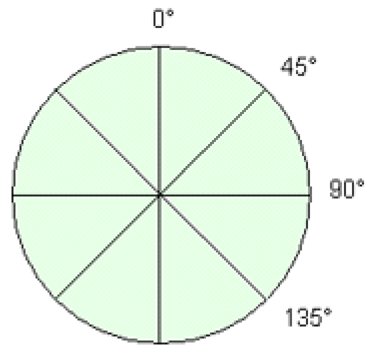
Object Features > Texture > Shape Texture Based on Sub-objects > Direction of Sub-objects: Stddev

Standard deviation of the directions of the sub-objects. Similar to Direction of Sub-objects: Mean feature, the sub-object main directions are weighted by the asymmetries of the respective sub-objects. The set of sub-object main directions from which the standard deviation is calculated is determined in the same way.

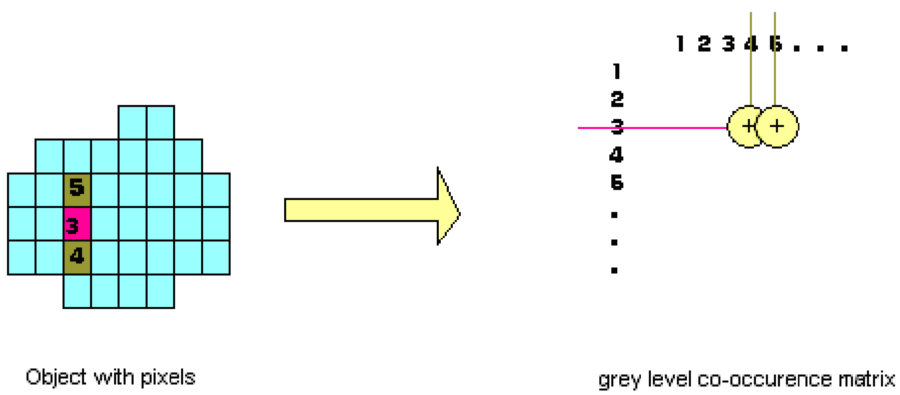
26.3 Texture After Haralick

Object Features > Texture > Texture After Haralick

The gray level co-occurrence matrix (GLCM) is a tabulation of how often different combinations of pixel gray levels occur in a scene. A different co-occurrence matrix exists for each spatial relationship. To receive directional invariance, the sum of all four directions (0°, 45°, 90°, 135°) are calculated before texture calculation. An angle of 0° represents the vertical direction, an angle of 90° the horizontal direction. In Definiens software, texture after Haralick is calculated for all pixels of an image object. To reduce border effects, pixels bordering the image object directly (surrounding pixels with a distance of 1) are additionally taken into account.



The normalized GLCM is symmetrical. The diagonal elements represent pixel pairs with no gray level difference. Cells, which are one cell away from the diagonal, represent pixel pairs with a difference of only one gray level. Similarly, values in cells that are two pixels away from the diagonal, show how many pixels have two gray levels and so forth. The more distant to the diagonal, the greater the difference between the pixels' gray levels is. Summing-up the values of these parallel diagonals, gives the probability for each pixel to be 0, 1, 2 or 3 etc. different from its neighbor pixels.



Another approach to measure texture is to use a gray-level difference vector (GLDV) instead of the GLCM. The GLDV is the sum of the diagonals of the GLCM. It counts the occurrence of references to the neighbor pixels' absolute differences. In Definiens software the GLCM and GLDV are calculated based on the pixels of an object. They are computed for each input layer. Within each Texture after Haralick feature you have the choice of either one of the above directions or of all directions.

The calculation of Texture after Haralick is independent of the image data's bit-depth. The dynamic range is interpolated to 8 bit before evaluating the co-occurrence. However, if 8 bit data is used directly the results will be most reliable. When using data of higher dynamic than 8 bit, the mean and standard deviation of the values is calculated. Assuming a Gaussian distribution of the values, a value greater than 95% is inbetween the interval, which is subdivided into 255 equal sub-intervals to obtain an 8-bit representation.

$\bar{x} - 3 \times \sigma < x < \bar{x} + 3 \times \sigma$

The result of any texture after Haralick analysis is dependent upon the direction of the analysis, whether it is All Directions, Direction 0°, Direction 45°, Direction 90° or Direction 135°. In addition, each feature is calculated based upon the gray values of one selectable layer.

The calculation of any Haralick texture feature is very processor intensive because of the calculation of the GLCM; as a result, performance may suffer.

26.3.1 Calculation of GLCM

Image object pixels are scanned and put into a 256 x 256 matrix, with pixel value and neighboring pixel value as co-ordinates. The value of the matrix is a normalized number of pair occurrences (number of occurrences ÷ number of all pairs). So GLCM matrix values are always within the range $[0, 1]$. The pixel value is either the pixel value of a particular layer or the mean value of all layers (depending on the Haralick feature chosen). The neighboring pixel calculation depends upon the feature angle:

- 0° – $(x; y-1)$ and $(x; y+1)$
- 45° – $(x-1, y-1)$ and $(x+1, y+1)$
- 90° – $(x-1; y)$ and $(x+1; y)$
- 135° – $(x-1, y+1)$ and $(x+1, y-1)$
- All dir – all neighbor pixels described above.

The actual Haralick feature is calculated using the GLCM matrix and algorithm of a particular feature. For example, the contrast feature is calculated using the formulae:

- for (int i=0; i<mtrx.Rows(); i++)
- for (int j=0; j<mtrx.Cols(); j++)
- ctrst += mtrx[i][j] * ((i-j)*(i-j)).

> **TIP**: For each Haralick texture feature there is a performance optimized version labeled quick 8/11. The performance optimization works only on data with a bit depth of 8 bit or 11-bit. Hence the label quick 8/11. Use the performance optimized version whenever you work with 8- or 11-bit data. For 16 bit data, use the conventional Haralick feature.

26.3.2 Parameters

- i is the row number
- j is the column number
- $V_{i,j}$ is the value in the cell i, j of the matrix
- $P_{i,j}$ is the normalized value in the cell i, j
- N is the number of rows or columns

26.3.3 Expression

Every GLCM is normalized according to the following operation:

$$P_{i,j} = \frac{V_{i,j}}{\sum\limits_{i,j=0}^{N-1} V_{i,j}}$$

26.3.4 References

Haralick RM, Shanmugan K and Dinstein I (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics.* Vol SMC-3, No. 6 pp. 610–621

Haralick RM (1979). Statistical and Structural Approaches to Texture. *Proceedings of the IEEE.* Vol. 67, No. 5, pp. 786–804

Conner RW and Harlow CA (1980). A Theoretical Comparison of Texture Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol PAMI-2, No. 3.

26.3.5 GLCM Homogeneity

Object Features > Texture > Texture After Haralick > GLCM Homogeneity

If the image is locally homogeneous, the value is high if GLCM concentrates along the diagonal. Homogeneity weights the values by the inverse of the contrast weight with weights, decreasing exponentially according to their distance to the diagonal.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i, j
- N is the number of rows or columns

Expression

$$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

Feature Value Range

$[0,1]$

26.3.6 GLCM Contrast

Object Features > Texture > Texture After Haralick > GLCM Contrast

Contrast is the opposite of homogeneity. It is a measure of the amount of local variation in the image. It increases exponentially as $i-j$ increases.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i,j
- N is the number of rows or columns

Expression

$$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2$$

Feature Value Range

$[0, 65025]$

26.3.7 GLCM Dissimilarity

Object Features > Texture > Texture After Haralick > GLCM dissimilarity

Similar to contrast, but increases linearly. High if the local region has a high contrast.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i,j
- N is the number of rows or columns

Expression

$$\sum_{i,j=0}^{N-1} P_{i,j}\,|i-j|$$

Feature Value Range

$[0, 255]$

26.3.8 GLCM Entropy

Object Features > Texture > Texture After Haralick > GLCM Entropy

The value for entropy is high, if the elements of GLCM are distributed equally. It is low if the elements are close to either 0 or 1. Since $\ln(0)$ is undefined, it is assumed that $0 \times \ln(0) = 0$.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i, j
- N is the number of rows or columns

Expression

$$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$$

Feature Value Range

$[0, 10404]$

26.3.9 GLCM Ang. 2nd Moment

Object Features > Texture > Texture After Haralick > GLCM Ang. 2nd Moment

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i, j
- N is the number of rows or columns

Expression

$$\sum_{i,j=0}^{N-1} (P_{i,j})^2$$

Feature Value Range

$[0, 1]$

26.3.10 GLCM Mean

Object Features > Texture > Texture After Haralick > GLCM Mean

The GLCM mean is the average expressed in terms of the GLCM. The pixel value is not weighted by its frequency of occurrence itself, but by the frequency of its occurrence in combination with a certain neighbor pixel value.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i,j
- N is the number of rows or columns

Expression

$$\mu_{i,j} = \frac{\sum_{i,j=0}^{N-1} P_{i,j}}{N^2}$$

Feature Value Range

$[0,255]$

26.3.11 GLCM Std. Dev.

Object Features > Texture > Texture After Haralick > GLCM Std Dev

GLCM standard deviation uses the GLCM, therefore it deals specifically with the combinations of reference and neighbor pixels. Thus, it is not the same as the simple standard deviation of gray levels in the original image.

Calculating the standard deviation using i or j gives the same result, since the GLCM is symmetrical.

Standard deviation is a measure of the dispersion of values around the mean. It is similar to contrast or dissimilarity.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i,j
- N is the number of rows or columns
- $\mu_{i,j}$ is the GLCM mean

Expression

$$\sigma_{i,j}^2 = \sum_{i,j=0}^{N-1} P_{i,j}(i, j - \mu_{i,j})$$

Standard Deviation

$$\sigma = \sqrt{\sigma_{i,j}^2}$$

Feature Value Range

$[0, 255]$

26.3.12 GLCM Correlation

Object Features > Texture > Texture After Haralick > GLCM Correlation

Measures the linear dependency of gray levels of neighboring pixels.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i, j
- N is the number of rows or columns
- $\mu_{i,j}$ is the GLCM mean
- $\sigma_{i,j}$ is the GLCM standard deviation

Expression

$$\sum_{i,j=0}^{N-1} P_{i,j} \left[\frac{(i,\mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_i^2)}} \right]$$

Feature Value Range

$[0, 1]$

26.3.13 GLDV Angular 2nd Moment

Object Features > Texture > Texture After Haralick > GLDV Ang. 2nd Moment

Similar to GLCM Angular Second Moment: it measures the local homogeneity. The value is high if some elements are large and the remaining ones are small.

Parameters

- N is the number of rows or columns
- V_k is the image object level, $k = 1, \dots n$

Expression

$$\sum_{k=0}^{N-1} V_k^2$$

Feature Value Range

$[0, 1]$

26.3.14 GLDV Entropy

Object Features > Texture > Texture After Haralick > GLDV Entropy

The values are high if all elements have similar values. It is the opposite of GLDV Angular Second Moment.

Parameters

- i is the row number
- j is the column number
- $P_{i,j}$ is the normalized value in the cell i, j
- N is the number of rows or columns
- V_k is the image object level, $k = 1, \dots n$

Expression

Since $\ln(0)$ is undefined, it is assumed that $0 \times \ln(0) = 0$.

$$\sum_{k=0}^{N-1} V_k(-\ln V_k)$$

Feature Value Range

$[0, 10404]$

26.3.15 GLDV Mean

Object Features > Texture > Texture After Haralick > GLDV Mean

The mean is mathematically equivalent to *GLCM Dissimilarity* on page 321. It is only left here for compatibility reasons.

Parameters

- N is the number of rows or columns
- V_k is the image object level, $k = 1, \dots n$

Expression

$$\sum_{k=0}^{N-1} k(V_k)$$

Feature Value Range

$[0, 255]$

26.3.16 GLDV Contrast

Object Features > Texture > Texture After Haralick > GLDV Contrast

It is mathematically equivalent to the GLCM Contrast measure, described previously. It is still included for compatibility reasons.

Parameters

- N is the number of rows or columns
- V_k is the image object level, $k = 1, \dots n$

Expression

$$\sum_{k=0}^{N-1} V_k k^2$$

Feature Value Range

$[0, 65025]$

26.3.17 GLCM/GLDV ... (Quick 8/11)

Object Features > Texture > Texture After Haralick GLCM/GLDV [quick 8/11]

For each Haralick texture feature there is a performance optimized version with a 'quick 8/11' suffix.

They are labeled 'quick 8/11' because the performance optimization works only on data with a bit depth of 8 or 11-bits. Use the performance optimized version whenever you work with 8- or 11-bit data; for 16 bit data, use the conventional Haralick feature.

27 Object Features: Variables

Object Features > Variables

Object Variables are local variables. In contrast to scene variables they store values for each individual image object. Think of it as each image object having its own version of the variable. There is one instance per image object in a project.

27.1 [Object Variable]

Object Features > Variables > Object Variable

If existing, all object variables are listed in the feature tree.

You can define a new object variable by clicking on Create new Object Variable to store interim values for each individual image object.

27.1.1 Editable Parameters

- Name
- Value: Insert an initial value for the variable.
- Choose whether the new variable is numeric (Double) or textual (String)
- Shared: Select if you want to share the new variable among different rule sets

28 Object Features: Hierarchy

Object Features > Hierarchy

Hierarchy features provide information about the embedding of an image object within the image object hierarchy.

28.1 Level

Object Features > Hierarchy > Level

Returns the level object associated with an image object.

28.2 Level Number

Object Features > Hierarchy > Level Number

The number of the image object level an image object is situated in.

This feature is helpful when performing classifications on different image object levels to define which class description is valid for which image object level.

28.2.1 Parameters

- $U_v(d)$ are superobjects of an image object v at distance d

28.2.2 Expression

$$\min_d U_v(d) = 0$$

28.2.3 Feature Value Range

$[1, \text{number of image object levels}]$

28.2.4 Conditions

This feature requires more than one image object level.

28.3 Number of Higher Levels

Object Features > Hierarchy > Number of Higher Levels

28.3.1 Parameters

- d is the distance between neighbors
- $S_v(d)$ are sub-objects of an image object v at a distance d

28.3.2 Expression

$$\left(\min_d S_v(d) = \varnothing\right) - 1$$

28.3.3 Feature Value Range

$[1, \text{number of image object levels} - 1]$

28.4 Number of Neighbors

Object Features > Hierarchy > Number of Neighbors

The number of the direct neighbors of an image object (in other words, neighbors with which it has a common border) on the same image object level in the image object hierarchy.

28.4.1 Parameters

- $N_v(d)$ are neighbors of an image object v at a distance d

28.4.2 Expression

$$\#N_v(d)$$

28.4.3 Feature Value Range

$[0, \text{number of pixels of entire scene}]$

28.5 Number of Sub-Objects

Object Features > Hierarchy > Number of Sub-objects

Concerning an image object, the number of sub-objects that are located on the next lower image object level in the image object hierarchy.

28.5.1 Parameters

- $S_v(d)$ are sub-objects of an image object v at a distance d

28.5.2 Expression

$\#S_v(d)$

28.5.3 Feature Value Range

$[0, \text{number of pixels of entire scene}]$

28.6 Number of Sublevels

Object Features > Hierarchy > Number of Sublevels

The number of image object levels situated below a given image object.

28.6.1 Parameters

- d is the distance between neighbors
- $U_v(d)$ are superobjects of an image object v at a distance d

28.6.2 Expression

$\left(\min\limits_{d} U_v(d) = \varnothing\right) - 1$

28.6.3 Feature Value Range

$[1, \text{number of image object levels} - 1]$

29 Object Features: Thematic Attributes

Object Features > Thematic Attributes

Thematic attribute features are used to describe an image object using information provided by thematic layers. If your scene contains a thematic layer, its thematic object's properties, can be used to create thematic attribute features, which can be used for developing ruleware. Depending on the attributes of the thematic layer, a large range of different features becomes available.

29.1 Number of Overlapping Thematic Objects

Object Features > Hierarchy > Number of Overlapping Thematic Objects

The number of thematic objects that an image object overlaps with. The scene contains a thematic layer, otherwise the feature value is 0.

29.1.1 Editable Parameter

- Thematic Layer

29.1.2 Feature Value Range

$[0, \text{number of thematic objects}]$

29.2 Thematic Objects Attribute

Object Features > Hierarchy > [Thematic Objects Attribute]

If existing, Thematic Objects Attribute features referring to a thematic layer are listed in the feature tree. Preconditions:

- The scene includes a thematic layer, otherwise the feature value is 0.
- Available only for image objects that overlap with one thematic object, otherwise the feature value is undefined.

29.2.1 Editable Parameters

- Thematic Layer
- Thematic Layer Attribute

30 Class-Related Features

Class-related features depend on image object features and refer to the class assigned ot image objects situated at any location in the image object hierarchy.

This location is specified for superobjects and sub-objects by the level distance defining the vertical distance in the image object hierarchy. For neighbor image objects the location is specified by the spatial distance defining the horizontal distance in the image object hierarchy. These feature distances can be edited.

Class-related features are global features because they are not related to individual image objects. Class-Related Features are grouped as follows:

- Customized class-related features refer to class-related features. They are available after they have been created.
- Relations to neighbor objects features are used to describe an image object by its relationships to other image objects of a given class on the same image object level.
- Relations to Sub-objects features describe an image object by its relationships to other image objects of a given class on a lower image object level in the image object hierarchy. You can use these features to evaluate sub-scale information because the resolution of image objects increases the lower you move in the image object hierarchy.
- Relations to Superobjects features describe an image object by its relations to other image objects of a given class on a higher image object level in the image object hierarchy. You can use these features to evaluate super-scale information because the resolution of image objects decreases the higher you move in the image object hierarchy.
- Relations to Classification features are used to find out about the current or potential classification of an image object.

30.1 Relations to Neighbor Objects

Class-Related Features > Relations to Neighbor Objects

Relations to neighbor objects features are used to describe an image object by its relationships to other image objects of a given class on the same image object level.

30.1.1 Existence of

Class-Related Features > Relations to Neighbor Objects > Existence Of

Existence of an image object assigned to a defined class in a certain perimeter (in pixels) around the image object concerned. If an image object of the defined classification is found within the perimeter, the feature value is 1 (= true), otherwise it would be 0 (= false). The radius defining the perimeter can be determined by editing the feature distance.

Expression

0 if $N_v(d,m) = \varnothing$, 1 if $N_v(d,m) \neq \varnothing$

Feature Value Range

$[0,1]$

30.1.2 Number Of

Class-Related Features > Relations to Neighbor Objects > Number Of

Number of objects belonging to the selected class in a certain distance (in pixels) around the image object.

Parameters

- v is the image object
- d is the distance between neighbors
- m is a class containing image objects

Expression

$\#N_v(d,m)$

Feature Value Range

$[0,\infty]$

30.1.3 Border To

Class-Related Features > Relations to Neighbor Objects > Border To

The absolute border of an image object shared with neighboring objects of a defined classification. If you use geo-referenced data, the feature value is the real border to image objects of a defined class; otherwise it is the number of pixel edges shared with the adjacent image objects, as by default the pixel edge-length is 1.

Parameters

- $b(v,u)$ is the length of common border between v and u
- $N_v(d)$ are neighbors to an image object v at a distance d

Expression

$$\sum_{u \in N_v(d,m)} b(v,u)$$

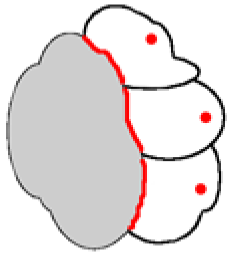


Figure 30.1. The absolute border between unclassified and classified image objects

Feature Value Range

$[0, \infty]$

30.1.4 Rel. Border to

Class-Related Features > Relations to Neighbor Objects > Rel. Border To

The feature Rel. Border To determines the relative border length an object shares with the image border. It describes the ratio of the shared border length of an image object (with a neighboring image object assigned to a defined class) to the total border length. If the relative border of an image object to image objects of a certain class is 1, the image object is totally embedded in them.

If the relative border is 0.5 then the image object is surrounded by half of its border. The relative border length can only be expressed in pixels.

Parameters

- $b(v,u)$ is the length of common border between v and u
- $N_v(d)$ are neighbors to an image object v at a distance d
- b_v is the image object border length

Expression

$$\frac{\sum_{u \in N_v(d,m)} b(v,u)}{b_v}$$

Feature Value Range

$[0, 1]$

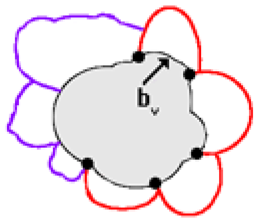


Figure 30.2. Relative border between neighbors

Conditions

If the relative border is 0 then the class m does not exist. If the relative border is 1 then the object v is completely surrounded by class m

30.1.5 Rel. Area of

Class-Related Features > Relations to Neighbor Objects > Rel. Area Of

The area covered by image objects of a selected class, found within a user-defined circular area around the selected image object, divided by the total area of image objects inside this area.

Editable Parameters

- Class
- Feature Distance: Radius (from the selected image object) of the area in pixels.

Parameters

- $N_v(d)$ are neighbors to an image object v at a distance d
- $\#P_u$ is total number of pixels contained in P_u

Expression

$$\frac{\sum_{u \in N_v(d,m)} \#P_u}{\sum_{u \in N_v(d)} \#P_u}$$

Feature Value Range

$[0, 1]$

Conditions

If the relative border is 0 then the class m does not exist. If the relative border is 1 then the object v is completely surrounded by class m

30.1.6 Distance to

Class-Related Features > Relations to Neighbor Objects > Distance To

The distance (in pixels) of the image object's center concerned to the closest image object's center assigned to a defined class. The image objects on the line between the image objects' centers have to be of the defined class.

Parameters

- $d(v,u)$ is the distance between v and u
- $V_i(m)$ is the image object level of a class m
- b_v is the image object border length

Expression

$$\min_{u \in V_i(m)} d(v,u)$$

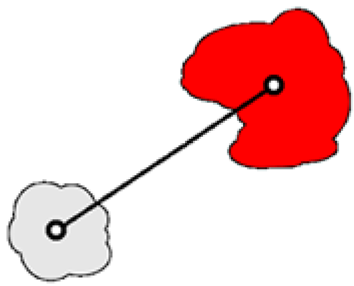


Figure 30.3. Distance between the centers of neighbors

Feature Value Range

$[0,\infty]$

30.1.7 Mean Diff. to

Class-Related Features > Relations to Neighbor Objects > Mean Diff. To

The mean difference of the layer L mean value of the image object concerned to the layer L mean value of all image objects assigned to a defined class.

Parameters

- v is the image object
- $N_v m$ is a neighbors to an image object v of a class m

Expression

$\bar{\Delta}(v, N_v(m))$

Feature Value Range

$[0, \infty]$

30.2 Relations to Sub-Objects

Class-Related Features > Relations to Sub Objects

Relations to sub-objects features describe an image object by its relationships to other image objects of a given class on a lower image object level in the image object hierarchy. You can use these features to evaluate sub-scale information because the resolution of image objects increases the lower you move in the image object hierarchy.

30.2.1 Existence Of

Class-Related Features > Relations to Sub Objects > Existence Of

The Existence Of feature checks if there is at least one sub-object assigned to a defined class. If there is one, the feature value is 1 (= true), otherwise the feature value is 0 (= false).

Parameters

- v is the image object
- d is the distance between neighbors
- m is a class containing image objects

Expression

0 if $S_v(d, m) = \varnothing$, 1 if $S_v(d, m) \neq \varnothing$

Feature Value Range

$[0, 1]$

30.2.2 Number of

Class-Related Features > Relations to Sub Objects > Number Of

The number of sub-objects assigned to a defined class.

Parameters

- v is the image object
- d is the distance between neighbors
- m is a class containing image objects

Expression

$$\#S_v(d,m)$$

Feature Value Range

$$[0,\infty]$$

30.2.3 Area of

Class-Related Features > Relations to Sub Objects > Area Of

The absolute area covered by sub-objects assigned to a defined class. If your data are geo-referenced, the feature value represents the real area.

Parameters

- d is the distance
- m is the class
- M is a sub-object in class m

Expression

$$\sum_{MES_v(d,m)} \#P_M$$

Feature Value Range

$$[0,\infty]$$

30.2.4 Rel. Area of

Class-Related Features > Relations to Sub Objects > Rel. Area Of

Area covered by sub-objects assigned to a given class divided by the total area of the image object concerned.

Editable Parameters

- Class
- Image object level Distance: Downward distance of image object levels in the image object hierarchy between the image object and the sub-object.

Parameters

- d is the image object level distance
- m is the class
- M is a sub-object in class m

Expression

$$\frac{\sum_{MES_v(d,m)} \#P_M}{\#P_v}$$

Feature Value Range

$[0,1]$

30.2.5 Clark Aggregation Index

Class-Related Features > Relations to Sub Objects > Clark Aggregation Index

For a superobject the Clark aggregation index gives evidence about the spatial distribution of its sub-objects of a certain class.

Parameters

- $D(x)$ is the mean spatial distance to next neighbor of the sub-objects of the class x
- $N(x)$ is the number of sub-objects of class x
- A is the number of pixels of the superobject (Area)
- Obs_mean_dist is the observed mean distance of sub objects to their spatial nearest neighbor with $\text{Obs\_mean\_dist} = \frac{\Sigma(D(x))}{N(x)}$
- Exp_mean_dist is the expected mean distance of sub objects to their spatial nearest neighbor with $\text{Exp\_mean\_dist} = \frac{1}{\sqrt[2]{\frac{N(x)}{A}}}$
- CAI is the Clark aggregation index

Expression

$$\text{CAI} = \frac{\text{Obs\_Mean\_Distance}}{\text{Exp\_Mean\_Distance}}$$

Feature Value Range

$[0, 2.149]$

- 0 – heavily clumped sub-objects
- 1 – homogeneous spatial distribution of sub-objects
- 2.149 – hexagonal distribution (edges of a honeycomb) of the sub-objects

30.3 Relations to Superobjects

Class-Related Features > Relations to Super Objects

Relations to superobjects features describe an image object by its relations to other image objects of a given class on a higher image object level in the image object hierarchy. You can use these features to evaluate super-scale information because the resolution of image objects decreases the higher you move in the image object hierarchy.

30.3.1 Existence of

Class-Related Features > Relations to Super Objects > Existence of

Checks if the superobject is assigned to a defined class. If this is true, the feature value is 1, otherwise 0.

Parameters

- v is the image object
- d is the distance to superobjects
- m is the a class containing image objects

Expression

0 if $U_v(d,m) = \varnothing$, 1 if $U_v(d,m) \neq \varnothing$

Feature Value Range

$[0, 1]$

30.4 Relations to Classification

Class-Related Features > Relations to Classification

Relations to classification features are used to find out about the current or potential classification of an image object.

30.4.1 Membership to

Class-Related Features > Relations to Classification > Membership to

In some cases it is important to incorporate the membership value to different classes in one class. This function allows explicit addressing of the membership values to different classes. If the membership value is below the assignment threshold, this value turns to 0.

Parameters

- v is the image object
- m is the a class containing image objects
- $\tilde{\varphi}(v,m)$ stored membership value of an image object v to a class m

Expression

$\tilde{\varphi}(v,m)$

Feature Value Range

$[0,1]$

30.4.2 Assigned Class

Class-Related Features > Relations to Classification > Assigned Class

Returns the class associated with an image object.

30.4.3 Classified as

Class-Related Features > Relations to Classification > Classified As

The idea of this feature is to enable the user to refer to the classification of an image object without regard to the membership value. It can be used to freeze a classification.

Parameters

- v is the image object
- m is a class containing image objects

Expression

$m(v)$

Feature Value Range

$[0,1]$

30.4.4 Classification Value of

Class-Related Features > Relations to Classification > Classification Value Of

This feature Classification value of allows you to explicitly address the membership values to all classes. As opposed to the feature Membership to it is possible to apply all membership values to all classes without restrictions.

Parameters

- v is the image object
- m is the a class containing image objects
- $\tilde{\varphi}(v,m)$ is a fuzzy membership value of an image object v to a class m

Expression

$\varphi(v,m)$

Feature Value Range

$[0,1]$

30.4.5 Class Name

Class-Related Features > Relations to Classification > Class Name

The Class name feature returns the name of the class (or superclass) of an image object (or its superobject).

Parameters

Distance in class hierarchy specifies the number of hierarchical levels when navigating from class to superclass. Using a distance of 0 the class name is returned, a distance of 1 will return the superclass name and so on. Distance in image object hierarchy specifies the number of hierarchical levels when navigating from object to superobject. Using a distance of 0 the class of the image object is used as a starting point for the navigation in the class hierarchy, a distance of 1 will start at the class of the superobject.

30.4.6 Class Color

Class-Related Features > Relations to Classification > Class Color

The Class color feature returns either the red, green or blue color component of the class (or superclass) of an image object (or its superobject).

Parameters

Color component is red, green or blue.

Distance in class hierarchy specifies the number of hierarchical levels when navigating from class to superclass. Using a distance of 0 the class name is returned, a distance of 1 will return the superclass name and so on.

Distance in image object hierarchy specifies the number of hierarchical levels when navigating from object to superobject. Using a distance of 0 the class of the image object is used as a starting point for the navigation in the class hierarchy, a distance of 1 will start at the class of the superobject.

31 Linked Object Features

Linked Object features are calculated by evaluating linked objects themselves.

31.1 Linked Object Count

Linked Object Features > Linked Objects Count

The number of objects connected by object links fulfilling the filter parameters.

This feature traverses all links complying with the filter parameters and counts all objects visited while traversing. You can use this feature to measure the number of objects in a linked structure.

31.1.1 Editable Parameters

- **Link Class Filter**: Select the classes that contain the links to be counted
- **Link Direction**: Filter on the link direction:
  - **All**: Count all links, independent of link direction.
  - **In**: Count only inbound links.
  - **Out**: Count only outbound links.
- **Max. Distance**: Maximum number of links in a path that will be taken into account. This parameter lets you limit the counting to an object's local neighborhood. Use 0 for objects one link away, 1 for objects two links away and 999999 to count all linked objects.

31.2 Linked Objects Statistics

Linked Object Features > Linked Objects Statistics

This feature allows computing a variety of statistical values for a group of linked objects.

31.2.1 Editable Parameters

- **Operation**: Select the type of statistical operation to be performed:
  - **Sum**: Compute the sum of the selected feature for all linked objects
  - **Mean**: Compute the mean value of the selected feature for all linked objects

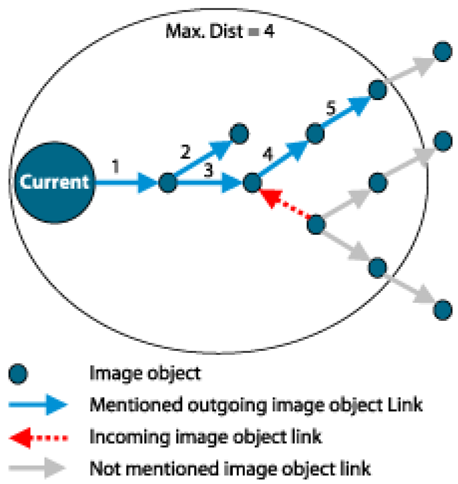


Figure 31.1. Example of a linked objects domain of outgoing direction within a maximum distance of 3

  - **Std Dev**: Compute the standard deviation of the selected feature for all linked objects
  - **Min**: Returns the minimum value of the selected feature for all linked objects
  - **Max**: Returns the maximum value of the selected feature for all linked objects
  - **Is Min**: Returns 1 if the image object is the minimum for the selected feature for all linked objects
  - **Is Max**: Returns 1 if the image object is the maximum for the selected feature for all linked objects
  - **Count**: Counts the linked objects. Ignores the selected feature
- **Feature**: Select the image object feature that will be used for the statistical operation.
- **Link Class Filter**: Select the classes that contain the links to be counted
- **Candidate Condition**: Select the condition the linked objects must fulfill to be included in the statistical computation
- **Link Direction**: Filter on the link direction:
  - **All**: Use all links, independent of link direction.
  - **In**: Use only inbound links.
  - **Out**: Use only outbound links.
- **Max. Distance**: The maximum number of links in a path that will be taken into account. This parameter allows you to limit the counting to the local neighborhood of the object. Use 0 for objects one link away, 1 for objects two links away and 999999 to count all linked objects

31.3 Linked Weight to PPO

Linked Object Features > Linked Weight to PPO

This feature calculates the link weight to a selected PPO. The weight is a measure of the degree of the overlap: if the overlap of the seed and target objects is 100% then the weight is 1. If the overlap is 50% only, then the weight is 0.5. If there are many links available, this should return the maximum of the given link weights.

In order to display meaningful values, create a feature with default values, select the image object of interested using Ctrl + left-mouse click (the object will outlined in dark green), then select a linked image object.

31.3.1 Editable Parameters

- **Hierarchical Distance to PPO** (the default is 1)
- **Link Class Filter**: Select the classes that contain the links to be taken into account.
- **Link Direction**: Filter on the link direction:
  - **All**: Use all links, independent of link direction
  - **In**: Use only inbound links
  - **Out**: Use only outbound links.

Hierarchical Distance to PPO

The number of hierarchical levels to reach PPO.

Link Class Filter

Restricts the active link classes.

Link Direction

Link Direction specifies the permitted link directions that create the linked object domain.

32 Scene Features

Scene features return properties referring to the entire scene or map. They are global because they are not related to individual image objects. Scene Features are grouped as follows:

- Scene variables are global variables that exist only once within a project. They are independent of the current image object.
- Class-related scene features provide information on all image objects of a given class per map.
- Scene-related features provide information on the scene.
- Customized scene features refer to scene features. They are available after they have been created.

32.1 Scene Variables

Scene Features > Scene Variables

Scene Variables are global variables that exist only once within a project. They are independent of the current image object. If they exist, all scene variables are listed in the feature tree. You can define a new scene variable by clicking on Create New Scene Variable. to store global interim values.

32.1.1 Editable Parameters

- Name
- Value: Insert an initial value for the variable
- Choose whether the new variable is numeric (Double) or textual (String)
- Part of Rule Set Parameters: When checked, the variable is added to the default parameter set, if not already present. If unchecked, the variable is removed from the default parameter set if already present.
- Shared: Select if you want to share the new variable among different rule sets.

32.2 Class-Related

Scene Features > Class-Related

Class-related scene features provide information on all image objects of a given class per map.

32.2.1 Number of Classified Objects

Scene Features > Class-Related > Number of Classified Objects

The absolute number of all image objects of the selected class on all image object levels.

Editable Parameters

- Class

Parameters

- $V(m)$ is all image objects classified as m
- m is the given class

Expression

$\#V(m)$

Feature Value Range

$[0, \text{number of image objects}]$

32.2.2 Number of Samples Per Class

Scene Features > Class-Related > Number of Samples Per Class

The number of all samples of the selected class on all image object levels.

Editable Parameters

- Class

Feature Value Range

$[0, \text{number of samples}]$

32.2.3 Area of Classified Objects

Scene Features > Class-Related > Area of Classified Objects

Editable Parameters

- Class

Parameters

- v is the image object
- m is the given class
- $V(m)$ is all image objects classified as m
- $\#P_v$ is the total number of pixels of image object v

Expression

$$\sum_{veV(m)} \#P_v$$

Feature Value Range

$[0, sx \times sy \times sz \times st]$

32.2.4 Layer Mean of Classified Objects

Scene Features > Class-Related > Layer Mean of Classified Objects

The mean of all image objects of the selected class on the selected image object levels.

Editable Parameters

- Class
- Image layer

Parameters

- v is the image object
- m is the given class
- $V(m)$ is all image objects classified as m
- $\bar{c}k_v$ is the mean intensity layer of an image object v

Expression

$$\frac{1}{\#V(m)} \sum_{veV(m)} \bar{c}_k(v)$$

Feature Value Range

$[0, 1]$

32.2.5 Layer Std. Dev. of Classified Objects

Scene Features > Class-Related > Layer Stddev. of Classified Objects

Editable Parameters

- Class
- Image layer

Parameters

- v is the image object
- m is the given class
- $V(m)$ is all image objects classified as m
- $\bar{c}k_v$ is the mean intensity layer of an image object v

Expression

$$\sigma_k\big(V(m)\big) = \sqrt{\frac{1}{\#V(m)}\left(\sum_{veV(m)}\big(c_k(v)\big)^2 - \frac{1}{\#V(m)}\left(\sum_{veV(m)} c_k(v)\right)^2\right)}$$

Feature Value Range

$[0,1]$

32.2.6 Statistic of Object Value

Scene Features > Class-Related > Statistic of Object Value

This feature allows the calculation of statistical operators over an existing object's feature.

Editable Parameters

- Class
- Statistical Operation
- Feature

32.2.7 Class Variables

Scene Features > Class-Related > Class Variable

Class Variables use classes as values. In a rule set they can be used instead of ordinary classes to which they point. If any exist, all variables that use classes as values are listed in the feature tree. You can define a new class variable by clicking on Create new Class Variable.

Editable Parameters

- Name: To define an new class variable, enter a name and click OK. The Create Class Variable dialog box opens for class variable settings.

32.3 Scene-Related

Scene Features > Scene-Related

Scene-related features provide information on the scene.

32.3.1 Existence of Object Level

Scene Features > Scene-Related > Existence of Object Level

This feature lets you test the existence of an object level with a given name. If the object level with the given name does not exist within the project, the feature value is 1 (true), otherwise it is 0 (false).

Editable Parameters

- Level Name

Feature Value Range

$[0,1]$

32.3.2 Existence of Image Layer

Scene Features > Scene-Related > Existence of Image Layer

Existence of a given image layer.

If the image layer with the given alias exists within the map the feature value is 1 (= true), otherwise it is 0 (= false).

Editable Parameter

- Image Layer alias

Feature Value Range

$[0,1]$

32.3.3 Existence of Thematic Layer

Scene Features > Scene-Related > Existence of Thematic Layer

Existence of a given thematic layer. If the thematic layer with the given alias exists within the map the feature value is 1 (true), otherwise it is 0 (false).

Editable Parameter

- Thematic layer alias

Feature Value Range

$[0,1]$

32.3.4 Existence of Map

Scene Features > Scene-Related > Existence of Map

This feature lets you test the existence of a map with a given name.

If the map with the given name does exist within the project, the feature value is 1 (true), otherwise it is 0 (false).

Editable Parameter

- Map

Feature Value Range

$[0,1]$

32.3.5 Mean of Scene

Scene Features > Scene-Related > Existence of Thematic Layer > Mean of Scene

Mean value of the selected image layer.

Editable Parameters

- Image Layer

Expression

$\bar{c}_k$

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

32.3.6 Std. Dev.

Scene Features > Scene-Related > Std Dev

Standard deviation of the selected layer.

Expression

σ_k

32.3.7 Smallest Actual Pixel Value

Scene Features > Scene-Related > Smallest Actual Pixel Value

Darkest actual intensity value of all pixel values of a given image layer.

Editable Parameter

- Image Layer

Expression

$c'^{\min}_k$

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

32.3.8 Magnification Read from Metadata

Scene Features > Scene-Related > Magnification Read from Metadata

Returns a value if the software has read a magnification value in the metadata.

Feature Value Range

$[0,1]$

32.3.9 Resolution Read from Metadata

Scene Features > Scene-Related > Resolution Read from Metadata

Returns a value if the software has read a resolution value in the metadata.

Feature Value Range

$[0,1]$

32.3.10 Bit Depth Read from Metadata

Scene Features > Scene-Related > Bit Depth Read from Metadata

Returns a value if the software has read a bit-depth value in the metadata.

Feature Value Range

$[0,1]$

32.3.11 Existence of Magnification in Metadata

Scene Features > Scene-Related > Existence of Magnification in Metadata

Checks whether magnification metadata is present in an imported project.

Feature Value Range

$[0,1]$

32.3.12 Existence of Resolution in Metadata

Scene Features > Scene-Related > Existence of Resolution in Metadata

Checks whether resolution metadata is present in an imported project.

Feature Value Range

$[0,1]$

32.3.13 Existence of Bit Depth in Metadata

Scene Features > Scene-Related > Existence of Bit Depth in Metadata

Checks whether bit-depth metadata is present in an imported project.

Feature Value Range

$[0,1]$

32.3.14 Largest Actual Pixel Value

Scene Features > Scene-Related > Largest Actual Pixel Value

Brightest actual intensity value of all pixel values of a given image layer.

Editable Parameter

- Image Layer

Expression

$c'^{\max}_k$

Feature Value Range

$[c_k^{\min}, c_k^{\max}]$

32.3.15 Validity of Region

Scene Features > Scene-Related > Validity of Region

The Validity of region feature checks if a given region is located within the scene.

If the given regions is located within the scene the feature value is 1 (true), otherwise it is 0 (false).

Editable Parameter

- Region

Feature Value Range

$[0, 1]$

32.3.16 Random

Scene Features > Scene-Related > Random

This feature generates identical sets of random numbers for processes that require random numbers but need to be run several times. Examples include test and training tables, and cross-validation.

Editable Parameters

- Min – the lower limit for generated random numbers
- Max – the upper limit for generated random numbers
- Seed variable – an optional variable that, if specified, is used to seed the random number generator.

32.3.17 Active Pixel T

Scene Features > Scene-Related > Active Pixel T

The pixel/voxel t -co-ordinate of the last-clicked cursor position.

Feature Value Range

$[0, \text{scene extent}]$

32.3.18 Active Pixel X

Scene Features > Scene-Related > Active Pixel X

The pixel x -co-ordinate of the last-clicked cursor position.

Feature Value Range

$[0, \text{scene extent}]$

32.3.19 Active Pixel Y

Scene Features > Scene-Related > Active Pixel Y

The pixel y -co-ordinate of the last-clicked cursor position.

Feature Value Range

$[0, \text{scene extent}]$

32.3.20 Active Pixel Z

Scene Features > Scene-Related > Active Pixel Z

The pixel/voxel z -co-ordinate of the last-clicked cursor position.

Feature Value Range

$[0, \text{scene extent}]$

32.3.21 Existence of Scene Resolution

Scene Features > Scene-Related > Existence of Scene Resolution

Existence of a scene resolution.

If scene resolution metadata exists, the feature value is 1 (= true), otherwise it is 0 (= false).

Feature Value Range

$[0,1]$

32.3.22 Is Active Vector Layer Changed

Scene Features > Scene-Related > Is Active Vector Layer Changed

Returns a value if a user has edited a polygon (a value of one is true).

Feature Value Range

$[0,1]$

32.3.23 Is Aperio Server Alive

Scene Features > Scene-Related > Is Aperio Server Alive

Returns a value of 1 if a user is connected to an Aperio Server; if not, a value of 0 is returned.

Feature Value Range

$[0,1]$

32.3.24 Is Aperio Server Slide

Scene Features > Scene-Related > Is Aperio Server Slide

Returns a value of 1 if an Aperio slide is being used; if not, a value of 0 is returned.

Feature Value Range

$[0,1]$

32.3.25 Map Origin T

Scene Features > Scene-Related > Map Origin T

The t -co-ordinate of the origin of a map. If the unit is a pixel, the origin is 0; if it is a co-ordinate, the origin is the absolute position of the map.

32.3.26 Map Origin X

Scene Features > Scene-Related > Map Origin X

The x -co-ordinate of the origin of a map. If the unit is a pixel, the origin is 0; if it is a co-ordinate, the origin is the absolute position of the map.

32.3.27 Map Origin Y

Scene Features > Scene-Related > Map Origin Y

The y -co-ordinate of the origin of a map. If the unit is a pixel, the origin is 0; if it is a co-ordinate, the origin is the absolute position of the map.

32.3.28 Map Origin Z

Scene Features > Scene-Related > Map Origin Z

The z -co-ordinate of the origin of a map. If the unit is a pixel, the origin is 0; if it is a co-ordinate, the origin is the absolute position of the map.

32.3.29 Map Size T

Scene Features > Scene-Related > Map Size T

The extent of a scene in time, calculated by the number of frames multiplied by the temporal distance between slices in time co-ordinate units.

Parameters

- st is the extent of a scene s in the t -direction
- u_{frames} is the temporal distance between slices in the time co-ordinate unit

Expression

$st = \#(\text{frames}) \times u_{\text{frames}}$

Feature Value Range

$[1, \infty]$

32.3.30 Map Size X

Scene Features > Scene-Related > Map Size X

The horizontal extent of a scene in the display unit.

Parameters

- sx is the extent of a scene s in the x -direction
- $\#\,(\text{pixels})_x \times u$ is number of pixels in the x -direction
- u is the size of a pixel in the co-ordinate system unit

Expression

$sx = \#(\text{pixels})_x \times u$

Feature Value Range

$[1, \infty]$

32.3.31 Map Size Y

Scene Features > Scene-Related > Map Size Y

The horizontal extent of a scene in the display unit.

Parameters

- sy is the extent of a scene s in the y -direction
- $\#\,(pixels)_y \times u$ is number of pixels in the y -direction
- u is the size of a pixel in the co-ordinate system unit

Expression

$sy = \#(\text{pixels})_y \times u$

Feature Value Range

$[1, \infty]$

32.3.32 Map Size Z

Scene Features > Scene-Related > Map Size Z

The extent of a scene in depth, calculated by the number of slices multiplied by the spatial distance between slices in the co-ordinate system units.

Parameters

- sx is the extent of a scene s in the x -direction
- {$u_{\mathrm {slices}} is the spatial distance between slices in the co-ordinate system unit

Expression

$sx = \#(\text{slices})_y \times u_{slices}$

Feature Value Range

$[1, \infty]$

32.3.33 Number of Image Layers

Scene Features > Scene-Related > Number of Image Layers

Number of layers K of a scene.

32.3.34 Number of Maps

Scene Features > Scene-Related > Number of Maps

Number of existing maps.

32.3.35 Number of Objects

Scene Features > Scene-Related > Number of Objects

Number of image objects of any class on all image object levels of the scene including unclassified image objects.

Expression

$\#V$

Feature Value Range

$[0, 1]$

32.3.36 Number of Pixels in Scene

Scene Features > Scene-Related > Number of Pixels in Scene

Number of pixels in the pixel layer of the scene.

Parameters

- sx is the scene extent in x
- sy is the scene extent in y
- sz is the scene extent in z
- st is the scene extent in t
- (sx, sy, sz, st) is the scene extent

Expression

$sx \times sy \times sz \times st$

Feature Value Range

$[0, \text{scene extent}]$

32.3.37 Number of Samples

Scene Features > Scene-Related > Number of Samples

Number of all samples on all image object levels of the map.

Feature Value Range

$[0, \text{number of samples}]$

32.3.38 Number of Thematic Layers

Scene Features > Scene-Related > Number of Thematic Layers

The number of thematic layers layers included in the map.

32.3.39 Original Scene ID

Scene Features > Scene-Related > Original Scene ID

The original identification number of the scene. It is generated automatically by Definiens Developer XD 2.0.4.

32.3.40 Original Scene Name

Scene Features > Scene-Related > Original Scene Name

The original name of the scene. It is generated automatically by Definiens Developer XD 2.0.4.

32.3.41 Scene ID

Scene Features > Scene-Related > Scene ID

The identification number of the scene. It is generated automatically by Definiens Developer XD 2.0.4.

32.3.42 Scene Magnification

Scene Features > Scene-Related > Scene Magnification

Returns the scene magnification defined in a rule set.

32.3.43 Scene Name

Scene Features > Scene-Related > Scene Name

Returns the scene name.

32.3.44 Scene Pixel Size

Scene Features > Scene-Related > Scene Pixel Size

The scene unit that corresponds to one pixel, as defined in the metadata or in the Modify Project dialog box.

32.3.45 Scene Resolution

Scene Features > Scene-Related > Scene Resolution

The resolution of the scene as given in the metadata of the project. The resulting number represents the size of a pixel in co-ordinate system unit. The value is 1 if no resolution is set.

32.3.46 Scene Bits Per Sample

Scene Features > Scene-Related > Scene Bits per Sample

Returns a value if the program has read a bits per sample value in the metadata.[1]

32.3.47 Second Level Scene Name

Scene Features > Scene-Related > Second Level Scene Name

Displays the name of second-level scenes.

1. This feature was previously called Scene Bit Depth

32.3.48 Slice Distance

Scene Features > Scene-Related > Slice Distance

The spatial distance between slices in the co-ordinate system unit.

Expression

u_{slices}

32.3.49 TMA Core Position

Scene Features > Scene-Related > TMA Core Position

The TMA Core Position feature returns a string in the format “A7” for TMA Cores and their sibling scenes, such as tiles or subsets. For non-TMA cores, an empty string (“ “) is returned.

32.3.50 TMA Core Type

Scene Features > Scene-Related > TMA Core Type

The TMA Core Type feature returns a string in the form of “positive”,”negative”, “empty” or “normal” for TMA cores and their child scenes, such as tiles or subsets. For non-TMA cores, an empty string (“ “) is returned.

32.3.51 TMA Number of Annotations

Scene Features > Scene-Related > TMA Number of Annotations

Returns the number of annotations in a TMA core.

32.3.52 Time Series Distance

Scene Features > Scene-Related > Time Series Distance

The temporal distance between slices in the time co-ordinate unit.

Expression

u_{frames}

32.3.53 Top Scene ID

Scene Features > Scene-Related > Top Scene ID

The identification number of the top scene.

A top scene is the original scene that was used to create scene copies, subsets, or tiles.[2]

32.3.54 User Name

Scene Features > Scene-Related > User Name

The user name.

32.4 Rule-Set Related

32.4.1 Class Variable

Scene Features > Rule-Set Related > Class Variable

Enter a name, color and value.

32.4.2 Level Variable

Scene Features > Rule-Set Related > Level Variable

Enter a name and value.

32.4.3 Map Name Variable

Scene Features > Rule-Set Related > Map Name Variable

Enter a name and value.

32.4.4 Image Layer Variable

Scene Features > Rule-Set Related > Image Layer Variable

Enter a name and value.

32.4.5 Thematic Layer Variable

Scene Features > Rule-Set Related > Thematic Layer Variable

Enter a name and value.

2. The Top Scene ID feature returns correct values in Definiens Developer XD only.

32.4.6 Number of Features in Feature List

Scene Features > Rule-Set Related > Number of Features in Feature List

Enter a name and feature(s).

32.4.7 Number of Specified Features in Feature List

Scene Features > Rule-Set Related > Number of Specified Features in Feature List

Enter a name and feature(s) for the Feature List and Feature(s).

32.4.8 Rule Set Array Values

Scene Features > Rule-Set Related > Rule Set Array Values

Enter the name of an array.

32.4.9 Rule Set Array Size

Scene Features > Rule-Set Related > Rule Set Array Size

Enter the name of an array.

32.4.10 Rule Set Array Item

Scene Features > Rule-Set Related > Rule Set Array Item

Enter the name of an array and the index value to access an array item.

32.5 Architect Related

Scene Features > Rule-Set Related

32.5.1 Number of Actions

Scene Features > Rule-Set Related > Number of Actions

The Number of Actions features allows a rule set to keep track of the structure of the analysis.

- If the value is 0, the action is not present in the library
- If the value is 1, the action is present in the library
- Where there are multiple actions, the value increases.

32.6 File System

Scene Features > File System

32.6.1 Resolve Path

Scene Features > File System > Resolve Path

Resolve Path lets you enter a string that will be converted to a file path relative to the Workspace folder. Paths can contain shortcuts and variables, which will be converted to the file path.

Supported Shortcuts

Definiens Developer XD 2.0.4 will replace the following variables with actual values:

Shortcut	Description
`"{:Workspc.Name}"`	Name of the workspace
`"{:Workspc.Guid}"`	GUID of the workspace
`"{:Workspc.Dir}"`	Path of the workspace file
`"{:Project.Name}"`	Name of the scene (absolute in the workspace)
`"{:Project.Guid}"`	GUID of the scene
`"{:Scene.Dir}"`	Directory of the scene
`"{:Scene.Name}"`	Name of the scene
`"{:ImgLayer(n).Dir}"`	Path to the input image layer *n*
`"{:Workspc.OutputRoot}"`	Path of the output root
`"{:Workspc.InputRoot}"`	Path of the input root
`"{:ActionLib.Dir}"`	Directory of the action library (if any)
`"{:Variable:abcde}"`	Value of variable abcde
`"{:Application.Dir}"`	Directory of the loaded application/portal (if any)

Example

The Workspace file is located at C:\My Samples\Workspaces\1206\WKSP_20110111–1206.dpj and the value of the variable str_temp1 is 91.32.12. The Resolve Path feature will convert the input path
"{:Workspc.OutputRoot}\results\report_Variable [str_temp1].html" to
"C:\My Samples\Workspaces\WKSP_20110111–1206\results\report_91.32.12.html"

32.7 UI Related

Scene Features > UI Related

32.7.1 Equalization

Scene Features > UI Related > Equalization

Equalization features return the level equalization values selected by users (which can be changed via the Edit Image Layer Mixing dialog box, or by dragging the mouse across the image with the right mouse button, if this option is enabled in Tools > Options).

Equalization Mode

Values may be:

- Linear
- Linear – Inverse
- Gamma Correction (Positive)
- Gamma Correction (Negative)
- Gamma Correction (Positive) – Inverse
- Gamma Correction (Negative) – Inverse

Window Level Center

The mid-point of the defined equalization range.

Window Level Max

The end-point of the defined equalization range.

Window Level Min

The beginning of the defined equalization range.

Window Level Width

The equalization range either side of the center (Window Level Max minus Window Level Min)

33 Process-Related Features

Process-related features are image object dependent features. They involve the relationship of a child process image object to a parent process. They are used in local processing. A process-related feature refers to a relation of an image objects to a parent process object (PPO) of a given process distance in the process hierarchy.

The process distance in the process hierarchy represents the upward distance of hierarchical levels in process tree between a process and the parent process. The process distance is a basic parameter of process-related features. Practically, that is the number of hierarchy levels in the Process Tree window above the current editing line, in which you find the definition of the parent object.

In the Process Tree, hierarchical levels are indicated by different indentations.

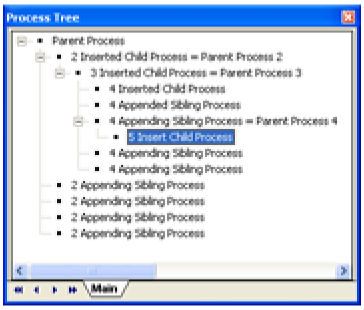


Figure 33.1. Process Tree window displaying a prototype of a process hierarchy. The processes are named according to their connection mode

For example, a process distance of 1 means that the parent process is located one hierarchical level above the current process. A process distance of 2 means that the parent process is located two hierarchical levels above the current process. Figuratively, you may say, a process distance of 2 defines the "grandparent" process.

33.1 Customized

Process-Related Features > Customized

33.1.1 Diff. PPO

Process-Related Features > Customized > Create New Diff. PPO

Define a customized feature in relation to the difference between a given feature value of an image object and the feature value of its parent process object (PPO).

Editable Parameters

- Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.
- Feature

Parameters

- v is the image object
- f is any feature
- ρ is a parent process object (PPO)

Expression

$f(v) - f(\rho)$

Feature Value Range

The range depends on the value of the feature in use.

Conditions

If $f(\rho) = 0 \therefore$ the formula is undefined.

33.1.2 Ratio PPO

Process-Related Features > Customized > Ratio PPO

Define a customized feature by the ratio between a given feature value of an image object and the feature value of its parent process object (PPO).

Editable Parameters

- Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.
- Feature

Parameters

- v is the image object
- f is any feature
- ρ is a parent process object (PPO)

Expression

$\frac{f(v)}{f(\rho)}$

Feature Value Range

The range depends on the value of the feature in use.

Conditions

If $f(\rho) = 0 \therefore$ the formula is undefined.

33.2 Border to PPO

Process-Related Features > Border to PPO

The absolute border of an image object shared with its parent process object.

33.2.1 Editable Parameters

- Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.

33.2.2 Parameters

- $b(v, \rho)$ is the length of common border between v and ρ with the parent process object

33.2.3 Expression

$b(v, r)$

33.2.4 Feature Value Range

$[0, \text{maximum size}]$

33.3 Distance to PPO

The distance between two parent process objects.

33.3.1 Editable Parameter

- Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.

33.4 Elliptic Dist. from PPO

Process-Related Features > Elliptic Dist. from PPO

The elliptic distance of an image object to its parent process object (PPO).

The ellipse is derived from area and center of gravity of the parent process object. The feature value is the distance between the center of gravity of the image object and the ellipse.

33.4.1 Parameters

- $\bar{x}_v, \bar{y}_v$

33.4.2 Editable Parameters

- Metadata Name: Name of the metadata source as used in the Modify Project dialog box. This name will be used in the feature tree for grouping corresponding metadata items. You can type a name of a not existing metadata source, to create a feature group in advance.
- Name: Name of the metadata item as used in the source data.
- Type: Select the type of the metadata item: string, double, or integer.
- Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.

33.4.3 Expression

$\varepsilon_{\rho}(\bar{x}_v, \bar{y}_v)$

33.4.4 Feature Value Range

$[0, \infty]$

33.5 Rel. border to PPO

Process-Related Features > Rel. Border to PPO

The ratio of the border length of an image object shared with the parent process object (PPO) to its total border length.

33.5.1 Editable Parameters

Process Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.

33.5.2 Parameters

- b_v is the image object border length
- $b(v, \rho)$ is the length of common border between v and ρ with the parent process object

33.5.3 Expression

$$\frac{b(v, \rho)}{b_v}$$

33.5.4 Feature Value Range

$[0, 1]$

33.6 Same Superobject as PPO

Process-Related Features > Same Super Object as PPO

The Same super object as PPO feature checks whether an image object and its parent process object (PPO) are parts of the same superobject.

33.6.1 Editable Parameters

- Process Distance: Upward distance of hierarchical levels in process tree between a process and the parent process.

33.6.2 Parameters

- v is the image object
- ρ is the parent process object (PPO)
- $U_v(d)$ is the superobject of an image object v at a distance d

33.6.3 Expression

$1 : U_v(d) = U_\rho(d)$
$0 : U_v(d) \neq U_\rho(d)$

33.6.4 Feature Value Range

$[0,1]$

33.7 Series ID

Process-Related Features > Series ID

The Execute Child as Series algorithm executes its child domains based on the Number of Cycles parameter. For each execution, a unique identifier is generated and the parameter Series Name prefixes this identifier. Therefore, if the domain has four image objects, and the number of cycles is set to 1, the child processes will be executed four times (once for each image object). On each cycle, the value attached to the series increases by one, for example:

- 1st cycle = “series1”
- 2nd cycle = “series2”

Therefore, the feature acts as a loop counter inside the Execute Child as Series algorithm.

34 Customized Features

34.1 Stain2 Isolation

Customized > Stain2 Isolation

Stain2 Isolation lets you separate two stains (for example, Brown stain and Blue stain).

34.1.1 Editable Parameters

Mean of which stain

Choose from 1 or 2. These values correspond to the numbers in the Stain Calibration parameters.

Input Layers

Select the image layers for the red, green and blue channels.

- Image layer red
- Image layer green
- Image layer blue

Stain Calibration

Enter the cx and cy values for the first (0) and second (1) stain these are converted from RGB.[1]

- cx stain 0
- cy stain 0
- cx stain 1
- cy stain 1

1. Van der Laak JAWM, Pahlplatz MMM, Hanselaar AGJM and De Wilde PCM (2000). Hue-Saturation-Density (HSD) Model for Stain Recognition in Digital Images From Transmitted Light Microscopy. *Cytometry* 39: 275–284.

White Calibrations

Sample the the values for each layer in the background and enter the RGB values.

- White background red
- White background green
- White background blue

34.2 Stain3 Isolation

Customized > Stain3 Isolation

Stain2 Isolation lets you separate three stains.

34.2.1 Editable Parameters

Mean of which stain

Choose from 1, 2 or 3. These values correspond to the numbers in the Stain Calibration parameters.

Input Layers

Select the image layers for the red, green and blue channels.

- Image layer red
- Image layer green
- Image layer blue

Stain Calibration

Enter the cx and cy values for the first (0), second (1) and third (2) stain these are converted from RGB.[2]

- cx stain 0
- cy stain 0
- cx stain 1
- cy stain 1
- cx stain 2
- cy stain 2

2. Van der Laak JAWM, Pahlplatz MMM, Hanselaar AGJM and De Wilde PCM (2000). Hue-Saturation-Density (HSD) Model for Stain Recognition in Digital Images From Transmitted Light Microscopy. *Cytometry* 39: 275–284.

White Calibrations

Sample the the values for each layer in the background and enter the RGB values.

- White background red
- White background green
- White background blue

35 Region Features

Region features return properties referring to a given region. They are global because they are not related to individual image objects. They are grouped as follows:

- Region-related features provide information on a given region.
- Layer-related region features evaluate the first and second statistical moment (mean, standard deviation) of a region's pixel value.
- Class-related region features provide information on all image objects of a given class per region.

35.1 Region-Related

Region Features > Region-Related

Region-related features provide information on a given region.

35.1.1 Number of Pixels in Region

Region Features > Region-Related > Number of pixels in region

Number of pixel, voxels, or voxel series forming a region.

Editable Parameters

- Region

Parameters

- R_x is the region extent in x
- R_y is the region extent in y
- R_z is the region extent in z
- R_t is the region extent in t

Expression

$R_x \times R_y \times R_z \times R_t$

Feature Value Range

$[0, \infty]$

35.1.2 T Extent

Region Features > Region Related > T Extent

Number of frames within a region.

Editable Parameters

- Region

Expression

R_t

Feature Value Range

$[0, \infty]$

35.1.3 T Origin

Region Features > Region Related > T Origin

The t -co-ordinate of the origin of a region.

Editable Parameters

- Region

Expression

t_G

Feature Value Range

$[0, \infty]$

35.1.4 X Extent

Region Features > Region Related > X Extent

The extent of a region in x

Editable Parameters

- Region

Expression

R_x

Feature Value Range

$[0, \infty]$

35.1.5 X Origin

Region Features > Region Related > X Origin

The x -co-ordinate of the origin of a region.

35.1.6 Y Extent

Region Features > Region Related > Y Extent

The pixel extent of a region in y .

Editable Parameters

- Region

Expression

x_G

Feature Value Range

$[0, \infty]$

35.1.7 Y Origin

Region Features > Region Related > Y Origin

The y -co-ordinate of the origin of a region.

Editable Parameters

- Region

Expression

y_G

Feature Value Range

$[0,\infty]$

35.1.8 Z Extent

Region Features > Region Related > Z Extent

The number of slices within a region.

Editable Parameters

- Region

Expression

R_z

Feature Value Range

$[0,\infty]$

35.1.9 Z Origin

Region Features > Region Related > Z Origin

The z -co-ordinate of the origin of a region.

Editable Parameter

- Region

Expression

z_G

Feature Value Range

$[0,\infty]$

35.2 Layer-Related

Region Features > Layer-Related

Layer-related region features evaluate the first and second statistical moment (mean, standard deviation) of a region's pixel value.

35.2.1 Mean

Region Features > Layer Related > Mean

Mean image layer value of a given image layer within a region.

Editable Parameters

- Region
- Image layer

Parameters

- $\bar{c}_k$ is the mean intensity of image layer k

Expression

$\bar{c}_k(R)$

Feature Value Range

$[c_R^{\min}, c_R^{\max}]$

35.2.2 Standard Deviation

Region Features > Layer Related > Std Dev

The layer standard deviation of a given image layer within a region.

Editable Parameters

- Region
- Image layer

Parameters

- $\bar{\sigma}k$ is the standard deviation of image layer k

Expression

$\sigma_k(R)$

Feature Value Range

$[c_R^{\min}, c_R^{\max}]$

35.3 Class-Related

Region Features > Class Related

Class-related region features provide information on all image objects of a given class per region.

35.3.1 Area of Classified Objects

Region Features > Class Related > Area of Classified Objects

The absolute area in pixels, voxels, voxel series of all image objects classified as a given class on a given image object level within a region.

Editable Parameters

- Region
- Class
- Image object level

Parameters

- $\#P$ is the total number of pixels
- i is the the given image object level
- R is the the given region
- m is the the given class

Expression

$\#P_i(R,m)$

Feature Value Range

$[0, R_x \times R_y \times R_z \times R_t]$

35.3.2 Relative Area of Classified Objects

Region Features > Region-Related > Rel. Area of classified objects

The relative area in pixel, voxel, or voxel series of all image objects of a given class on a given image object level within a region.

Editable Parameters

- Region
- Class
- Image object Level

Parameters

- $\#P$ is the total number of pixels
- i is the the given image object level
- R is the the given region
- m is the the given class

Expression

$$\frac{\#P_i(R,m)}{\#P_i(R)}$$

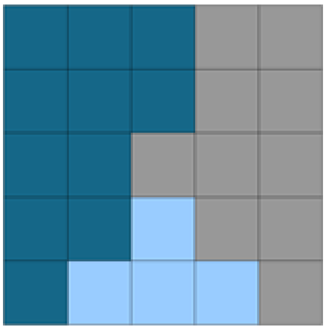


Figure 35.1. Schematic display of the relative area of a class within a two dimensional region. In this 5×5 sized region the feature values are:

Class	Area of classified objects	Relative area of classified objects
	11 pixels	11/25 = 0.44
	10 pixels	10/25 = 0.4
	4 pixels	4/25 = 0.16

Feature Value Range

$[0,1]$

36 Image Registration Features

Region Features > Image Registration Features

36.1 Object-Related

Region Features > Image Registration Features > Object Related

36.1.1 Object Landmarks on the Map

Region Features > Image Registration Features > Object Related > Object Landmarks on the Map

Returns the number of landmarks within the image object in the current map.

36.2 Scene-Related

Region Features > Image Registration Features > Scene Related

36.2.1 Landmarks on the Map

Region Features > Image Registration Features > Object Related > Landmarks on the Map

Returns the number of landmarks with the current map.

37 Metadata

Metadata items can be used as a feature in rule set development. To do so, you have to provide external metadata in the feature tree. If you are not using data import procedures to convert external source metadata to internal metadata definitions, you can create individual features from a single metadata item.

37.1 [Metadata Item]

Metadata > [Metadata item]

If existing, all metadata items are listed in the feature tree.

You can define a new metadata item by clicking on Create new Metadata item.

37.1.1 Editable Parameters

- Map
- Metadata Name: Name of the metadata source as used in the Modify Project dialog box. This name will be used in the feature tree for grouping corresponding metadata items. You can type a name of a not existing metadata source, to create a feature group in advance.
- Metadata Access: Select the mode of accessing the metadata:
  - Direct: Metadata is accessed by the name of the metadata item
  - Conversion: Metadata is converted as defined in the Metadata Conversion dialog box
- Name: Name of the metadata item as used in the source data.
- Value: If Metadata Access is set to Conversion , you can define an internal metadata item value , which can be different from the external value.
- Type: Select the type of the metadata item: string, double, or integer.

37.2 [Active Slice Metadata Item]

Metadata > [Active Slice Metadata Item]

Where a project are made up of multiple image slices, there may be different metadata values on every slice this is especially the case for DICOM images. The Active Slice Metadata Item feature allows you to capture metadata values for the image slice in the current active view.

37.2.1 Editable Parameters

- Metadata Name: Name of the metadata source as used in the Modify Project dialog box. This name will be used in the feature tree for grouping corresponding metadata items. You can type a name of a not existing metadata source, to create a feature group in advance.
- Name: Name of the metadata item as used in the source data.
- Type: Select the type of the metadata item: string, double, or integer.

38 Feature Variables

Feature Variables have features as their values. Once a feature is assigned to a feature variable, the feature variable can be used like that feature. It returns the same value as the feature to which it points. It uses the unit of whatever feature is assigned as a variable. It is possible to create a feature variable without a feature assigned, but the calculation value would be invalid.

In a rule set, feature variables can be used like the corresponding feature.

38.1 [Feature Variable]

Feature Variables > [Feature variable]

If existing, all variables that use features as values are listed in the feature tree. You can define a new feature variable by clicking on Create New Feature Variable.

38.1.1 Editable Parameters

- Name
- Value: The selected feature.

39 Widget Parameters for Architect Action Libraries

Listed below are all available widgets and parameters for use in Action Libraries. Definiens Developer XD 2.0.4 now has additional features that allow you to display TIFF images: if you use these functions, ensure your images are stored in the same folder as your Action Library, to ensure they will be visible to other users.

39.1 Add Checkbox

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control
- **Value Checked**: The value of the variable if the checkbox is checked
- **Value Unchecked**: The value of the variable if the checkbox is unchecked
- **Process on Change**: The name of the process that will be executed when a user changes a value

39.2 Add Drop-down List

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both

- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Selection Change**: The name of the process that will be executed when the selection changes
- **Items**: Add or edit the items in the drop-down list

39.3 Add Button

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Process on Press**: Enter the name of the process executed when the button is pressed
- **Process on Release**: Enter the name of the process executed when the button is released
- **Button Text**: Enter the text to be displayed on the button
- **Image**: Allows you to display a TIFF image (transparency is supported)
- **Radio Button**: If set to 'yes', a radio button is displayed
- **Ruleset**: Navigate to the ruleset (.dcp file) containing the processes to be executed

39.4 Add Radio Button Row

- **Rule Set**: Navigate to the ruleset (.dcp file) containing the processes to be executed
- **Radio Mode**: If set to 'true', the buttons become radio buttons; if set to 'false' they are normal buttons
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Action Buttons**:
  - Text: Enter the text to be displayed on the action button
  - Description: Enter a description to appear at the bottom of the description pane. The text will also display as a tooltip. You may have a text description or an image description, but not both

  – Description Image: Add a TIFF image to the description area (transparency is supported). You may have a text description or an image description, but not both
  – Process Path: Enter the process path to be executed by the action button
  – Process Path on Release: Enter the process path to be executed when the button is released
  – Execute on a Separate Thread: If set to 'yes', the process will be executed on a separate thread
  – Image File: Navigate to the path of the image file for the button image
  – Highlight Image File: Navigate to the path of the image file for the highlighted button image
  – Hot Key: Lets you define a single hot key to execute an action
  – Radio Mode: Select 'push button' to use this style specifically; if 'from widget' is selected, the radio mode from the widget is used

39.5 Add Toolbar

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Change**: The name of the process that will be executed when a user changes a value

39.6 Add Editbox

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control

- **Process on Change**: The name of the process that will be executed when a user changes a value

39.7 Add Editbox With Slider

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Change**: The name of the process that will be executed when a user changes a value
- **Maximum Value**: Enter the maximum value for the slider
- **Minumum Value**: Enter the minimum value for the slider
- **Tick Frequency**: Enter a value to define how often tick marks appear next to the slider
- **Jump Value**: Enter a value to define the increments when the slider is moved
- **Ruleset**: Navigate to the ruleset (.dcp file) containing the processes to be executed

39.8 Add Select Class

- **Text**: Enter the text that will be displayed to the user
- **Variable**: Select or enter the variable that gets updated by this control
- **Available Classes**: Select the classes to be made available (if none are selected, classes are displayed based on action dependencies)
- **Process on Selection Change**: Enter the name of the process to be executed when a selection changes
- **Dependency Handling**: The dependency effect of the selected class. Choose from one of the following:
  - None
  - Required. This activates the parameter Dependency Error Message, which is displayed when a dependency conflict occurs. Use the tag `#class` within the error text to identify the class name
  - Forbidden. This activates the parameter Dependency Error Message, which is displayed when a dependency conflict occurs. Use the tag `#class` within the error text to identify the class name
  - Added
  - Removed
- **Allow Create New Class**: Select ‘yes’ to allow the user to create a new class from the widget.

- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled

39.9 Add Select Feature

- **Text**: Enter the text that will be displayed to the user
- **Variable**: Select or enter the variable that gets updated by this control
- **Type**: Select the filter type for selectable features – all, scene features or object features
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Process on Change**: The name of the process that will be executed when a user changes a value

39.10 Add Select Multiple Features

- **Text**: Enter the text that will be displayed to the user
- **Image File**: Navigate to the path of the image file for the button image
- **Highlight Image File**: Navigate to the path of the image file for the highlighted button image
- **Variable**: Select or enter the variable that gets updated by this control
- **Type**: Select the filter type to be used:
  - All
  - Object Features
  - Scene Features
  - Custom Filter – this option brings up the Custom Feature Filter parameter, which allows you to select which features are available for selection
- **Show Template Features**: Select 'yes' to show template features, while selecting features

- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Process on Change**: The name of the process that will be executed when a user changes a value

39.11 Add Select File

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable for File**: Add or select the variable that gets updated by this control
- **File Filter**: Add a filter for file type, for example `*.tif` or `*.*`
- **Type**: Select the file to be selected:
  - Image
  - Thematic: If this option is selected, two more parameters appear:
    - Variable for Attribute File, to be updated by this control
    - Variable for Attribute ID Column, to be updated by this control
- **Mode**: Select between ‘open’ and ‘save’
- **Process on Change**: The name of the process that will be executed when a user changes a value

39.12 Add Select Level

- **Text**: Enter the text that will be displayed to the user
- **Variable**: Select or enter the variable that gets updated by this control
- **Show all Levels**: If set to ‘yes’, action dependencies for listing available levels are ignored
- **Process on Selection Change**: The name of the process that will be executed when the selection changes
- **Show Pixel Level**: Select ‘yes’ to show the pixel level

- **Dependency Handling**: The dependency effect of the selected class. Choose from one of the following:
  - None
  - Required. This activates the parameter Dependency Error Message, which is displayed when a dependency conflict occurs. Use the tag `#class` within the error text to identify the class name
  - Forbidden. This activates the parameter Dependency Error Message, which is displayed when a dependency conflict occurs. Use the tag `#class` within the error text to identify the class name
  - Added
  - Removed
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled

39.13 Add Select Image Layer

- **Text**: Enter the text that will be displayed to the user
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Selection Change**: The name of the process that will be executed when the selection changes
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled

39.14 Add Select Thematic Layer

- **Text**: Enter the text that will be displayed to the user
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Selection Change**: The name of the process that will be executed when the selection changes

- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled

39.15 Add Select Folder

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Change**: The name of the process that will be executed when a user changes a value

39.16 Add Slider

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Variable**: Select or enter the variable that gets updated by this control
- **Process on Change**: The name of the process that will be executed when a user changes a value
- **Maximum Value**: Enter the maximum value for the slider
- **Minumum Value**: Enter the minimum value for the slider

- **Tick Frequency**: Enter a value to define how often tick marks appear next to the slider
- **Jump Value**: Enter a value to define the increments when the slider is moved
- **Ruleset**: Navigate to the ruleset (.dcp file) containing the processes to be executed

39.17 Add Edit Layer Names

- **Text**: Enter the text that will be displayed to the user
- **Tooltip**: Enter the text to appear as a tooltip
- **Image File**: Navigate to the path of the image file for the button image
- **Highlight Image File**: Navigate to the path of the image file for the highlighted button image
- **Button Size**: Button size in pixels
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled

39.18 Add Layer Drop-down List

- **Text**: Enter the text that will be displayed to the user
- **Description**: Enter a description of your widget. The text will appear in the Description pane when the cursor hovers over the widget. You may have a text description or an image description, but not both
- **Description Image**: Allows you to display a TIFF image in the Description pane (transparency is supported). You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Items**: Add or edit the items in the drop-down list
- **Process on Selection Change**: The name of the process that will be executed when the selection changes

39.19 Add Manual Classification Buttons

- **Rule Set**: Navigate to the ruleset (.dcp file) containing the processes to be executed
- **Mode**: Choose between push button or checkbox

- **Show Edit Classes Button**: Select 'yes' to display the Edit Classes button, which allows users to change the class names and colors of widget classes
- **Image File for Edit Classes**: Navigate to the path of the image file for the button image
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled or disabled. A zero value defines the widget state as disabled
- **Action Buttons**:
  - Class: Assign a class to the button
  - Description: Enter a description to appear at the bottom of the description pane. The text will also display as a tooltip
  - Description Image: Add a TIFF image to the description area (transparency is supported). Uploading an image will replace the text in the description in the description area. You may have a text description or an image description, but not both
  - Process Path: Enter the process path to be executed by the action button
  - Process Path on Release: Enter the process path to be executed when the button is released
  - Hot Key: Lets you define a single hot key to execute an action

39.20 Add Select Array Items

- **Text**: Enter the text that will be displayed to the user
- **Items**: Enter a name then select the array that defines the selectable items
- **Selected Items**: Enter a name then select the array that receives the selectable items
- **Process on Change**: The name of the process that will be executed when a user changes a value
- **Description**: Enter a description to appear at the bottom of the description pane. The text will also display as a tooltip
- **Description Image**: Add a TIFF image to the description area (transparency is supported). Uploading an image will replace the text in the description in the description area. You may have a text description or an image description, but not both
- **Show/Hide Variable**: Enter or select the name of the variable that defines whether the widget is visible. If you enter 0, the widget is hidden
- **Enable/Disable Variable**: Enter or select the name of the variable that defines whether the widget is enabled.

40 General Reference

40.1 Use Variables as Features

The following variables can be used as features:

- Scene variables
- Object variables
- Class variables
- Feature variables.

They are displayed in the feature tree of, for example, the Feature View window or the Select Displayed Features dialog box.

40.2 About Metadata as a Source of Information

Many image data formats include metadata or come with separate metadata files, which provide information about the related image (for example, the acquisition time). This metadata information can be converted into features, to use in image analysis.

The available metadata depends on the image reader or camera used, the industry-specific environment, and the settings. Some industry-specific examples are:

- Satellite image data containing metadata providing information on cloudiness
- Microscopy image data that contains metadata on the magnification used. The metadata provided can be displayed in the Image Object Information window and can also be displayed in the feature tree, in the Feature View window or the Select Displayed Features dialog box.

40.2.1 Convert Metadata and Add it to the Feature Tree

When importing data, you can load metadata source files to make them available within the map. To provide external metadata items as features in the feature tree, you must convert external source metadata to an internal metadata definition.

When developing rule sets, metadata definitions will be included in rule sets allowing the serialization of metadata usage. Metadata conversion is available through import functions and creating metadata item features.

40.3 General Reference

40.3.1 Rendering a Displayed Image

1. The first step reads out the displayed area from the selected image layers according to the screen size and zoom settings. Then image layer equalization is applied. The result is an 8-bit raw gray value image for each image layer. These gray value images are mixed into one raw RGB image by a layer mixer according to the current layer mixing settings
2. Finally the image equalizing is applied to create the output RGB image that is displayed on the screen.

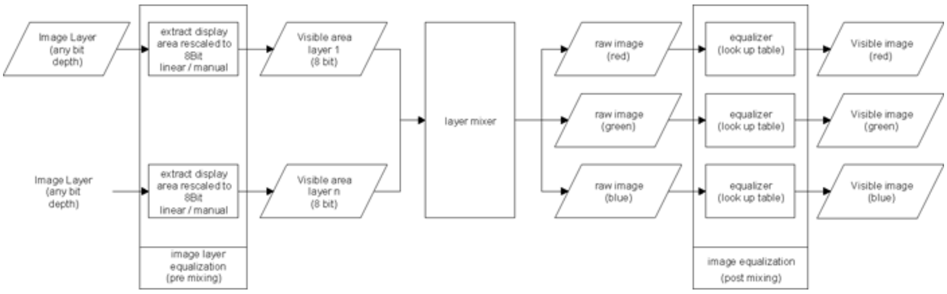


Figure 40.1. The rendering process of an image displayed in the map view

Image Layer Equalization

Image layer equalization is part of the rendering process of the scene display within the map view.

Image layer equalization maps the input data of an image layer which may have different intensity ranges to the unified intensity range $[0 \ldots 255]$ of an 8-bit gray value image. For 8-bit data no image layer equalization is necessary. All other data types have to be converted into an 8-bit representation at this step of the rendering process.

This function is implemented as a mapping of the input range to the display range of $[0 \ldots 255]$. Image layer equalization can be either linear or manual. By default, the input data is mapped to the gray value range by a linear function.

Data Type	Input Range	Mapping Function
8-bit	$[0 \ldots 255]$	$c_s = c_k$(no transformation)
16-bit unsigned; 32-bit unsigned	$[0 \ldots \max_2(c_k)]$	$c_s = \frac{255 \times c_k}{\max_2(c_k)}$
16-bit signed; 32-bit signed	$[\min_2(c_k) \ldots \max_2(c_k)]$	$c_s = \frac{255 \times (c_k - \min_2(c_k))}{\max_2(c_k) - \min_2(c_k)}$
32 bit float	$[\min_{10}(c_k) \ldots \max_{10}(c_k)]$	$c_s = \frac{255 \times (c_k - \min_{10}(c_k))}{\max_{10}(c_k) - \min_{10}(c_k)}$

- c_k is the intensity value in image layer k
- c_s is the intensity value on the screen

- $\min_2(c_k) = \max\{x : x = -2^n; x \leq c'^{\min}_k\}$ is the highest integer number that is a power of 2 and darker than the darkest actual intensity value of all pixel values of the selected image layer
- $\max_2(c_k) = \max\{x : x = 2^n; x \geq c'^{\max}_k\}$ is the lowest integer number that is a power of 2 and is brighter than the brightest actual intensity value of all pixel values of the selected image layer
- $\min_{10}(c_k) = \max\{x : x = -10^n; x \leq c'^{\min}_k\}$ is the highest integer number that is a power of 10 and darker than the darkest actual intensity value of all pixel values of the selected image layer
- $\max_{10}(c_k) = \max\{x : x = 10^n; x \geq c'^{\max}_k\}$ is the lowest integer number that is a power of 10 and is brighter than the brightest actual intensity value of all pixel values of the selected image layer.

Manual Image Layer Equalization With manual image layer equalization you can specify the mapping function for each layer individually by defining both the input range $[c_{\min} \dots c_{\max}]$ and the mapping function.

Data Type	Mapping Function
Linear	$c_s = 255 \times (c_k - c_{\min}) \div (c_{\max} - c_{\min})$
Linear – inverse	$c_s = 255 - 255 \times (c_k - c_{\min}) \div (c_{\max} - c_{\min})$
Gamma correction (positive)	$c_s = 255 \times ((c_k - c_{\min}) \div (c_{\max} - c_{\min}))^2$
Gamma correction (negative)	$c_s = 255 \times ((c_k - c_{\min}) \div (c_{\max} - c_{\min}))^{0.5}$
Gamma correction (positive) – inverse	$c_s = 255 - 255 \times ((c_k - c_{\min}) \div (c_{\max} - c_{\min}))^2$
Gamma correction (negative) – inverse	$c_s = 255 - 255 \times ((c_k - c_{\min}) \div (c_{\max} - c_{\min}))^{0.5}$

- c_k is the intensity value in image layer k
- c_s is the intensity value on the screen
- $c_{\min}$ is the smallest input intensity value (adjustable)
- $c_{\max}$ is the largest input intensity value (adjustable).

Image Equalization

Image equalization is performed after all image layers are mixed into a raw RGB (red, green, blue) image. If more than one image layer is assigned to one screen color (red, green or blue) this approach leads to higher quality results. Where only one image layer is assigned to each color, as is common, this approach is the same as applying equalization to the individual raw layer gray value images

There are different modes for image equalization available. All of them rely on image statistics. These are computed on the basis of a 256 × 256 pixel sized thumbnail of the current raw RGB image.

None No (None) equalization allows you to see the image data as it is, which can be helpful at the beginning of rule set development, when looking for an approach. The output from the image layer mixing is displayed without further modification.

- $[0 \dots 255]$ is the input range

- $[0 \ldots 255]$ is the mapping function

Linear Equalization Linear equalization with 1.00% is the default for new scenes. Commonly it displays images with a higher contrast as without image equalization.

Linear equalization maps each color – red, green, and blue – from an input range $[c_{\min} \ldots c_{\max}]$ to the available screen intensity range $[0 \ldots 255]$ by a linear mapping. The input range can be modified by the percentage parameter p. The input range is computed such that p per cent of the pixels are not part of the input range. In case $p = 0$ this means the range of used color values is stretched to the range $[0 \ldots 255]$. For $p > 0$ the mapping ignores $\frac{p}{2}$ per cent of the darkest pixels and $\frac{p}{2}$ per cent of the brightest pixels. In many cases a small value of p leads to better results because the available color range can be better used for the relevant data by ignoring the outliers.

- $c_{\min} = \dfrac{\max\{c : \#\{(x,y) : c_k(x,y) < c_{\min}\}}{(sx \times sy) \geq \frac{p}{2}}$
- $c_{\max} = \dfrac{\min\{c : \#\{(x,y) : c_k(x,y) > c_{\max}\}}{(sx \times sy) \geq \frac{p}{2}}$
- $[c_{\min} \ldots c_{\max}]$ is the input range
- $c_s = 255 \times \left(\dfrac{c_k - c_{\min}}{c_{\max} - c_{\min}} \right)$ is the mapping function.

Standard Deviation Equalization With its default parameter of 3.0, standard deviation renders a similar display as linear equalization. Use a parameter around 1.0 for an exclusion of dark and bright outliers.

Standard deviation equalization maps the input range to the available screen intensity range $[0 \ldots 255]$ by a linear mapping. The input range $[c_{\min} \ldots c_{\max}]$ can be modified by the width p. The input range is computed such that the center of the input range represents the mean value of the pixel intensities mean (c_k). The left and right borders of the input range are computed by taking n times the standard deviation σ_k to the left and the right. You can modify the parameter n.

- $c_{\min} = \text{mean}(c_k) - n \times \sigma_k$
- $c_{\max} = \text{mean}(c_k) + n \times \sigma_k$
- $[c_{\min} \ldots c_{\max}]$ is the input range
- $c_s = 255 \times \left(\dfrac{c_k - c_{\min}}{c_{\max} - c_{\min}} \right)$ is the mapping function.

Gamma Correction Equalization Gamma correction equalization is used to improve the contrast of dark or bright areas by spreading the corresponding gray values. Gamma correction equalization maps the input range to the available screen intensity range $[0 \ldots 255]$ by a polynomial mapping. The input range $[c_{\min} \ldots c_{\max}]$ cannot be be modified and is defined by the smallest and the largest existing pixel values.

- $c_{\min} = {c'}_k^{\min}$ – the smallest existing pixel values
- $c_{\max} = {c'}_k^{\max}$ – the largest existing pixel values
- $[c_{\min} \ldots c_{\max}]$ is the input range
- $c_s = 255 \times \left(\dfrac{c_k - c_{\min}}{c_{\max} - c_{\min}} \right)^n$ is the mapping function.

You can be modify the exponent of the mapping function by editing the equalization parameter. Values of n less than 1 emphasize darker regions of the image, values larger than one emphasize darker areas of the image. A value from $n = 1$ represents the linear case.

Histogram Equalization Histogram equalization is well suited for LANDSAT images but can lead to substantial over-stretching on many normal images. It can be helpful in cases you want to display dark areas with more contrast. Histogram equalization maps the input range to the available screen intensity range $[0\ldots255]$ by a nonlinear function. Simply said, the mapping is defined by the property that each color value of the output image represents the same number of pixels. The respective algorithm is more complex and can be found in standard image processing literature.

Manual Image Layer Equalization Manual image layer equalization allows you to control equalization in detail. For each image layer individually, you can set the equalization method specifying the mapping function. Further you can define the input range by setting minimum and maximum values.

Acknowledgments

Page collection published 05 September 2012 ¶ Typeset by Wikipublisher