Towards the Probabilistic Fusion of Learned Priors into Standard Pipelines for 3D Reconstruction

Tristan Laidlow[1], Jan Czarnowski[1], Andrea Nicastro[1], Ronald Clark[1] and Stefan Leutenegger[2]

***Abstract*— The best way to combine the results of deep learning with standard 3D reconstruction pipelines remains an open problem. While systems that pass the output of traditional multi-view stereo approaches to a network for regularisation or refinement currently seem to get the best results, it may be preferable to treat deep neural networks as separate components whose results can be probabilistically fused into geometry-based systems. Unfortunately, the error models required to do this type of fusion are not well understood, with many different approaches being put forward. Recently, a few systems have achieved good results by having their networks predict probability distributions rather than single values. We propose using this approach to fuse a learned single-view depth prior into a standard 3D reconstruction system.**

Our system is capable of incrementally producing dense depth maps for a set of keyframes. We train a deep neural network to predict discrete, nonparametric probability distributions for the depth of each pixel from a single image. We then fuse this "probability volume" with another probability volume based on the photometric consistency between subsequent frames and the keyframe image. We argue that combining the probability volumes from these two sources will result in a volume that is better conditioned. To extract depth maps from the volume, we minimise a cost function that includes a regularisation term based on network predicted surface normals and occlusion boundaries. Through a series of experiments, we demonstrate that each of these components improves the overall performance of the system.

I. INTRODUCTION

There has been continued research interest in using Structure-for-Motion (SfM) and Visual Simultaneous Localisation and Mapping (SLAM) for the incremental creation of dense 3D scene geometry due to its potential applications in safe robotic navigation, augmented reality and manipulation. Until recently, dense monocular reconstruction systems typically worked by minimising the photometric error over several frames. As this minimisation problem is not well-constrained due to occlusion boundaries or regions of low texture, most reconstruction systems employ regularisers based on smoothness ([1], [2]) or planar ([3]–[5]) assumptions.

With the continued success of deep learning in computer vision, there have been many suggestions for data-driven approaches to the monocular reconstruction problem. Several of these approaches propose a completely end-to-end framework, predicting the scene geometry from either a single image ([6]–[9]) or several consecutive frames ([10]–[15]). Most promising, however, are those systems that combine deep learning with standard geometric constraints ([16]–[22]). It was shown in [23] that learning-based and geometry-based approaches have a complementary nature as learning-based systems tend to perform better on the interior points of objects but blur edges, whereas geometry-based systems typically do well on areas with a high image gradient but perform poorly on interior points that may lack texture.

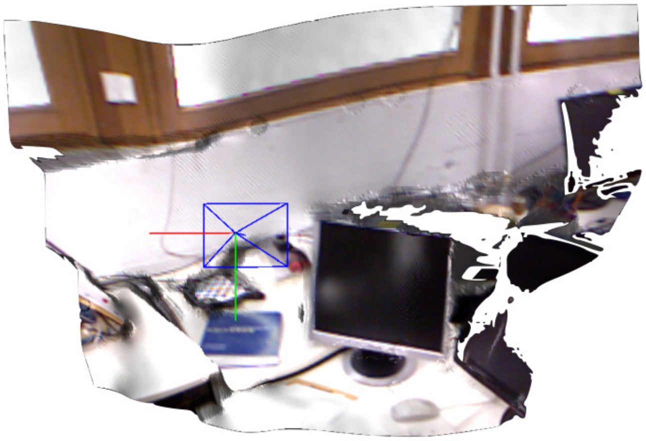

Fig. 1: Fusing a single-view depth probability distribution predicted by a DNN with standard photometric error terms helps to resolve ambiguities in the photometric error due to occlusions or lack of texture. The above projected keyframe depth map was created by our system.

The optimal way to combine these two approaches, however, is not clear. The best current results seem to come from systems that take the output of traditional geometry-based systems and feed these into a deep neural network (DNN). A particularly impressive example of this type of system includes DeepTAM [13], which passes a photometric cost volume through a network to extract a depth map.

It may be desirable, however, to use learning-based systems as an additional component that is fused into the pipeline of a traditional system. Such a framework would prevent the necessity of having to perform an expensive neural network pass every time the geometric information is updated. Also, as DNNs perform best on images close to the training dataset, it would be possible to switch the network component on or off or switch between different networks depending on the environment being reconstructed.

[1] Dyson Robotics Laboratory at Imperial College, Imperial College London, UK

[2] Smart Robotics Lab, Imperial College London, UK

Research presented in this paper has been supported by Dyson Technology Ltd. Corresponding author: Tristan Laidlow, t.laidlow15@imperial.ac.uk

The difficulty of this approach, however, is that to probabilistically fuse the network outputs into a 3D reconstruction system, some measure of the uncertainty associated with each prediction is required.

In general, uncertainty can be classified into two categories: model or epistemic uncertainty and statistical or aleatoric uncertainty. In [24], the authors suggest using a Monte Carlo dropout technique to estimate the model uncertainty of a network, but this requires multiple expensive network passes.

Like [25], the authors of [26] propose having the network predict its own aleatoric uncertainty and using a Gaussian or Laplacian likelihood as the loss function during training, which was used by [21] for 3D reconstruction. The problem with this approach is that it forces the network to predict a parametric and unimodal distribution. As shown in [27], this type of distribution may be particularly ill-suited to dense reconstruction where there is a clear need for a multi-hypothesis prediction.

One proposal has been to use a multi-headed network ([13], [28]) with each head making a separate prediction. From these many predictions, one can calculate the mean and covariance to use in a probabilistic fusion algorithm. The drawbacks of this approach are that it increases the size of the network and requires a careful balancing of the relative size of the network body and heads.

Recently, both [9] and [29] achieved impressive results by having their networks predict discrete, nonparametric probability distributions. While [29] uses these distributions to fuse the output with other network predictions, to the best of our knowledge, no one has used this method to fuse the predictions of networks with the output of standard reconstruction pipelines.

In this paper, we propose a 3D reconstruction system that fuses together the output of a DNN with a standard photometric cost volume to create dense depth maps for a set of keyframes. We train a network to predict a discrete, non-parametric probability distribution for the depth of each pixel over a given range from a single image. Like [29], we refer to this collection of probability distributions for each pixel in the keyframe as a "probability volume". Then, with each subsequent frame, we create a probability volume based on the photometric consistency between the current frame and the keyframe image and fuse this into the keyframe volume. The main contribution of this paper is to demonstrate that combining the probability volumes from these two sources often results in a better conditioned probability volume. We extract depth maps from the probability volume by optimising a cost function that includes a regularisation term based on network predicted surface normals and occlusion boundaries. Please see Figure 1 for an example keyframe reconstruction created by our system.

II. METHOD

In this section, we describe our method for fusing predictions from DNNs into a standard 3D reconstruction pipeline to produce dense depth maps.

Our system represents the observed geometry as a collection of keyframe-based "probability volumes". That is, instead of representing the surface as a depth map with a single depth estimate per pixel, the depth is represented with a per-pixel discrete probability distribution over a given depth range. These probability volumes are initialised with the output of a monocular depth prediction network. With each additional RGB image, the system computes a cost volume based on the photometric consistency. This cost volume is then converted to a probability volume and fused into the volume of the current keyframe. Once the number of inliers drops below a given threshold, a new keyframe is created. To propagate information from one keyframe to another, we warp the previous distribution and fuse it into the new one.

When we want to extract a depth map from the probability volume, we could take the maximum probability depth values, but in featureless regions where there is also high network uncertainty this would be susceptible to false minima and cause local inconsistencies in the prediction. Also, as the probability distribution is discrete, taking the maximum would result in a quantisation of the final depth prediction. To overcome these shortcomings, we first construct a smooth probability density function (PDF) from the volume using a kernel density estimation (KDE) technique. We then minimise the negative log probability of this PDF along with a regularisation term. While many dense systems propose using regularisers based on smoothness ([1], [2]) or planar ([3], [4], [30]) assumptions, we follow the examples of [16] and [21] and penalise our reconstruction for deviating from the surface normals predicted by a DNN.

A. Multi-Hypothesis Monocular Depth Prediction

Rather than predict a single depth value for each pixel, our network predicts a discrete depth probability distribution over a given range, similar to [9] and [29]. Not only does this allow the network to express uncertainty about its prediction, but it also allows the network to make a multi-hypothesis depth prediction. As discussed in [9], the prediction of the depth probability distribution can be improved by having a variable resolution over the depth range. We choose a log-depth parameterisation, following the examples of [31] and [6]. By uniformly dividing the depth range in log-space, we achieve the desired result of having higher resolution in the areas close to the camera and lower resolution farther away.

For our network architecture (see Figure 2), we use a ResNet-50 encoder [32] followed by three upsample blocks, each consisting of a bilinear upsampling layer, a concatenation with the input image, and then two convolutional layers to bring the output back up to the input resolution. All inputs and outputs have a resolution of 256x192.

As we are having the network predict a discrete distribution rather than a depth map, we cannot use a standard loss function based on the sum of squared errors. A cross-correlation loss would not be ideal either, as we would like to penalise the network less for predicting high probabilities in incorrect bins that are close to the true bin than in bins farther

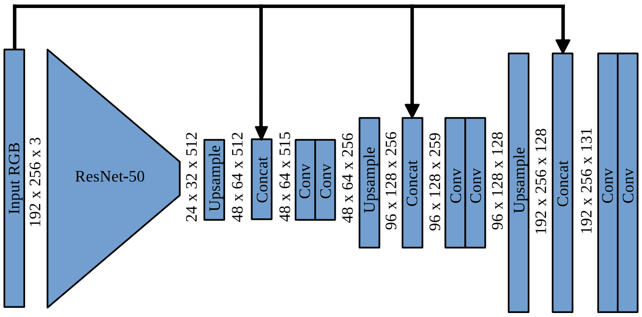

Fig. 2: Our network consists of a ResNet-50 encoder with an output stride size of 8 and no global pooling layer. We then pass the output of the encoder through three upsample blocks consisting of a bilinear resize, concatenation with the input image, and then two convolutional layers to match the output resolution to the input. The probability distribution that the network outputs is discretised over 64 channels.

away. Instead, we choose to use the ordinal loss function proposed in [9]:

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_i \left[\sum_{k=0}^{k_i^*} \log(p_{\boldsymbol{\theta},i}(k_i^* \geq k)) + \sum_{k=k_i^*+1}^{K-1} \log(1 - p_{\boldsymbol{\theta},i}(k_i^* \geq k)) \right], \tag{1}$$

where

$$p_{\boldsymbol{\theta},i}(k_i^* \geq k) = \sum_{j=k}^{K-1} p_{\boldsymbol{\theta},i}(k_i^* = j), \tag{2}$$

$\boldsymbol{\theta}$ is the set of network weights, K is the number of bins over which the depth range is discretised, k_i^* is the index of the bin containing the ground truth depth for pixel i, and $p_{\boldsymbol{\theta},i}(k_i^* = j)$ is the network prediction of the probability that the ground truth depth is in bin j.

Like [29], we train our network on the ScanNet RGB-D dataset [33]. No fine-tuning was done on our evaluation dataset, the TUM RGB-D dataset [34]. We set the depth range to be between 10cm and 12m and group the log-depth values uniformly into 64 bins.

Each keyframe created by our system is initialised with this network output.

B. *Fusion with Photometric Error Terms*

For each additional reference frame, we construct a DTAM-style cost volume [1]. First, we normalise both the keyframe and reference frame images by subtracting their means and dividing by their standard deviations. We then calculate the photometric error by warping the normalised keyframe image into the reference frame for each depth value in the cost volume and taking the sum of squared differences on 3x3 patches. To simplify the later fusion, we use the midpoint of each of the depth bins used for the network prediction as the depth values in the cost volume. Poses are obtained from an oracle, such as a separate tracking system like ORB-SLAM2 [35].

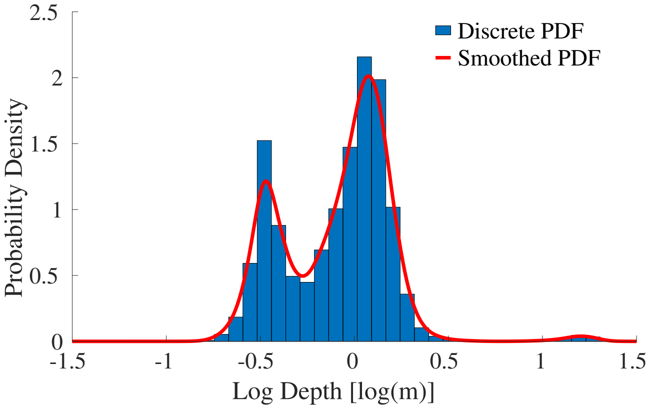

Fig. 3: Our fusion algorithm produces a discrete probability distribution for each pixel in the keyframe. To reduce discretisation errors and to have a continuous cost function for the optimiser, we convert the probability values along each ray into a smooth probability density function using a kernel density estimation technique.

To convert to a probability volume, we separately scale the negative of the squared photometric error for each pixel such that it sums to one over the ray. We then fuse this new probability volume, p_{RF}, into the current keyframe volume, p_{KF}:

$$p_i(k_i^* = k) = p_{\text{KF},i}(k_i^* = k) p_{\text{RF},i}(k_i^* = k), \tag{3}$$

for each pixel i, which is then scaled to sum to one.

C. *Kernel Density Estimation*

To avoid a quantisation of the final depth prediction and to have a smooth function to use in the optimisation step, we construct a PDF for the depth of each pixel using a KDE technique with Gaussian basis functions:

$$f_i(d) = \sum_{k=0}^{K-1} p_i(k_i^* = k) \phi\left(d(k), \sigma\right) \tag{4}$$

where $\phi\left(\mu, \sigma\right)$ is the probability density of the Gaussian distribution with mean μ and standard deviation σ, $d(k)$ is the depth value at the midpoint of bin k, and σ is a constant smoothing parameter across all pixels and depth values. The value of σ is a hyperparameter that needs to be tuned empirically; we found that $\sigma = 0.1$ works well in our setting.

An example of a discrete PDF produced by our system and the smoothed result after applying the KDE technique is shown in Figure 3.

D. *Regularisation*

Although the fused probability volume will have more local consistency than using the photoconsistency terms alone, the result can still be improved by adding a regularisation term to the optimisation used to extract the depth map. While most dense reconstruction systems base their regularisers on smoothness or planar assumptions, we propose using the surface normals predicted by a DNN as was done in both

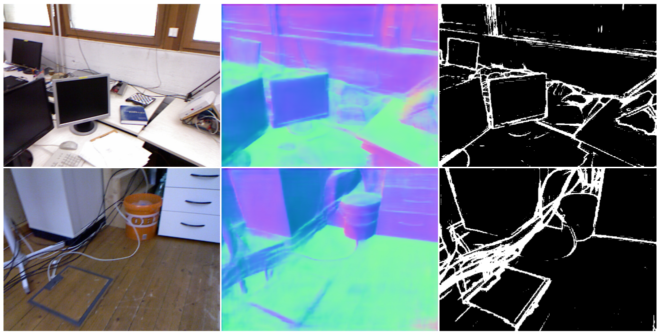

Fig. 4: To regularise our depth estimate, we use the surface normals and occlusion boundaries predicted by SharpNet [36]. Some examples of the predictions made by SharpNet on the TUM RGB-D dataset [34] are shown above. From left to right: input RGB images, predicted normals, predicted occlusion boundaries with a probability greater than 0.4.

[16] and [21] as this may allow for better preservation of fine-grained local geometry. To predict the surface normals from the keyframe image, we use the state-of-the-art network SharpNet [36]. As we determine the local surface orientation of our depth estimation from neighbouring pixels and we do not wish incur high costs at depth discontinuities, we mask the regularisation term at occlusion boundaries, which are also predicted by SharpNet. Since SharpNet actually predicts a probability of each pixel belonging to an occlusion boundary, we include all pixels with a probability higher than 0.4 in the mask. Example predictions of surface normals and occlusion boundaries made by SharpNet on the TUM RGB-D dataset [34] are shown in Figure 4.

E. Optimisation

To extract a depth map from the probability volume, we minimise a cost function consisting of two terms:

$$c(\mathbf{d}) = c_f(\mathbf{d}) + \lambda c_{\hat{\mathbf{n}}}(\mathbf{d}), \tag{5}$$

where $\mathbf{d}$ is the set of depth values to be estimated, and λ is a hyperparameter used to adjust the strength of the regularisation term. Empirically, we found $\lambda = 1.0 \cdot 10^7$ to work well.

The first term, c_f, imposes a unary constraint on each of the pixels:

$$c_f(\mathbf{d}) = -\sum_i \log\left(f_i(d_i)\right) \tag{6}$$

where $f_i(d_i)$ is the smoothed PDF of pixel i evaluated at depth d_i.

The second term, $c_{\hat{\mathbf{n}}}$, is a regularisation term that combines two pairwise constraints:

$$\begin{aligned} c_{\hat{\mathbf{n}}}(\mathbf{d}) = \sum_i & b_i \left(\langle \hat{\mathbf{n}}_i, d_i \mathbf{K}^{-1}\tilde{\mathbf{u}}_i - d_{i+1}\mathbf{K}^{-1}\tilde{\mathbf{u}}_{i+1}\rangle\right)^2 \\ & + b_i \left(\langle \hat{\mathbf{n}}_i, d_i \mathbf{K}^{-1}\tilde{\mathbf{u}}_i - d_{i+\mathrm{W}}\mathbf{K}^{-1}\tilde{\mathbf{u}}_{i+\mathrm{W}}\rangle\right)^2 \end{aligned} \tag{7}$$

where $b_i \in \{0, 1\}$ is the value of the occlusion boundary mask for pixel i, $\langle \cdot, \cdot \rangle$ is the dot product operator, $\hat{\mathbf{n}}_i$ is the normal vector predicted by SharpNet, $\mathbf{K}$ is the camera intrinsics matrix, $\tilde{\mathbf{u}}_i$ is the homogeneous pixel coordinates for pixel i, and W is the width of the image in pixels.

We minimise the cost function by applying 100 iterations of gradient descent with a step size of 0.2, and initialise the optimisation with the maximum probability depth values from the fused probability volume. The process of going from a fused probability volume through the smoothing and optimisation to an extracted depth map is currently only able to run at a few Hz; however, this could be improved significantly by using Newton's method or the primal-dual algorithm. As the main contribution of this work is to show the benefit of the fusion, we leave this for future work.

F. Keyframe Warping

To avoid throwing away information on the creation of each new keyframe, we warp the probability volume of the current keyframe into the new one. As the probability volume is a distribution over the depth values of a pixel, however, warping the probability volume is not trivial. To do this, we propose using a discrete variation of the method described in [37], where we first convert the depth probability distribution to an occupancy-based probability volume, where for each depth bin along the ray there is a probability that the associated point in space is occupied. We then warp this occupancy grid into the new frame and convert back to a depth probability distribution.

We start by defining the probability that the voxel $S_{k,i}$ (correpsonding to depth bin k along the ray of pixel i) is occupied, conditioned on the depth belonging to bin j:

$$p(S_{k,i} = 1 | k_i^* = j) = \begin{cases} 0 & \text{if } k < j \\ 1 & \text{if } k = j \\ \frac{1}{2} & \text{if } k > j \end{cases} \tag{8}$$

To convert a depth probability distribution into an occupancy probability, we marginalise out the conditional:

$$p(S_{k,i} = 1) = \sum_{k=0}^{K-1} p_i(k_i^* = k) p(S_{k,i} = 1 | k_i^* = k) \tag{9}$$

where $p_i(k_i^* = k)$ is the value of the depth probability volume in bin k for pixel i.

As the occupancy grid represents probabilities for locations in 3D space, we can directly warp this into the new keyframe, filling in any unknown values with a default occupancy probability (we use a value of 0.01).

After warping, the occupancy grid can be converted back into a depth probability distribution:

$$p_i(k_i^* = k) = \prod_{j<k} \left[1 - p(S_{j,i} = 1)\right] p(S_{k,i} = 1), \tag{10}$$

and scaled so that the distribution sums to one along the ray.

III. Experimental Results

We evaluate our system on a sample of sequences from the TUM RGB-D dataset [34]. Please note that only the RGB images are processed by our system and the depth channel is only used as a "ground truth" with which to validate our results against.

A. Qualitative Results

Figure 5 shows the various PDFs for a sample of four pixels taken from a keyframe in the TUM RGB-D sequence *fr1/desk*. The PDFs in the first row are those predicted by the DNN. Note that the network is able to make multi-hypothesis predictions and can have varying degrees of certainty. The PDFs in the second row are those that result from the photometric cost volume. For some of the pixels (such as pixels A and C), the photometric error results in a clear peak. This situation is most often found on corners and edges in the image where there are large intensity gradients. For pixels in textureless regions or on occlusion boundaries or areas with repeating patterns, the photometric PDF may have many peaks (such as pixel B) or no peak at all (such as Pixel D). The final row of the figure shows the fused PDF for each of the pixels. By fusing the two PDFs together, uncertainty can be reduced and ambiguous photometric data can be resolved.

An example reconstruction for a single keyframe with various ablations is shown in Figure 6.

B. Quantitative Evaluation

We demonstrate the value of fusion on the reconstruction pipeline by comparing the performance of the system on the first 400 frames of three TUM RGB-D sequences under three different scenarios: using only the network probability volume, using only the photometric probability volume and using the fused probability volume. To isolate the performance of our reconstruction system, we use the ground truth poses provided in the dataset. We evaluate the performance using three metrics defined in [6]: the absolute relative difference (L1-rel), the squared relative difference (L2-rel) and the root mean squared error (RMSE). Note that since the photometric probability volume has extremely noisy results on textureless surfaces, we found that the results were improved by initialising the optimisation with the expected value of the depth from the probability volume rather than the highest probability depth.

The results are presented in Table I. While there is a large performance gain in using the network over the photometric probability volume, the best outcome is achieved by fusing the two together.

Sequence	System	L1-rel	L2-rel	RMSE
fr1/desk	Network-Only	0.275	0.185	0.450
	Photometric-Only	0.532	0.474	0.837
	Fused	**0.260**	**0.165**	**0.431**
fr1/room	Network-Only	0.242	0.151	0.487
	Photometric-Only	0.588	0.752	1.171
	Fused	**0.234**	**0.140**	**0.479**
fr1/xyz	Network-Only	0.191	0.101	0.354
	Photometric-Only	0.543	0.450	0.823
	Fused	**0.187**	**0.090**	**0.339**

TABLE I: Comparison of reconstruction errors on select TUM RGB-D [34] sequences showing the relative performance of using only the network-predicted probability volume, only the photometric probability volume, and the fused probability volume.

Sequence	System	L1-rel	L2-rel	RMSE
fr1/desk	No Optimisation	0.310	0.274	0.552
	Smoothing-Only	0.308	0.271	0.548
	Total Variation	0.280	0.213	0.481
	Normals + Occlusions	**0.260**	**0.165**	**0.431**
fr1/room	No Optimisation	0.292	0.233	0.591
	Smoothing-Only	0.289	0.228	0.586
	Total Variation	0.265	0.183	0.530
	Normals + Occlusions	**0.234**	**0.140**	**0.479**
fr1/xyz	No Optimisation	0.245	0.213	0.512
	Smoothing-Only	0.242	0.208	0.506
	Total Variation	0.205	0.135	0.405
	Normals + Occlusions	**0.187**	**0.090**	**0.339**

TABLE II: Comparison of reconstruction errors on select TUM RGB-D [34] sequences showing the relative performance of different regularisation schemes. No Optimisation: results from taking the depth value with the maximum probability in the probability volume. Smoothing-Only: results from minimising the smoothed negative log probability density function without including a regularisation term. Total Variation: results from using the total variation of the depth as a regulariser. Normals + Occlusions: the pipeline as described in this paper.

Sequence	System	L1-rel	L2-rel	RMSE
fr1/desk	No Keyframe Warping	0.290	0.203	0.471
	Keyframe Warping	**0.260**	**0.165**	**0.431**
fr1/room	No Keyframe Warping	0.254	0.166	0.513
	Keyframe Warping	**0.234**	**0.140**	**0.479**
fr1/xyz	No Keyframe Warping	0.270	0.194	0.469
	Keyframe Warping	**0.187**	**0.090**	**0.339**

TABLE III: Comparison of reconstruction errors on select TUM RGB-D [34] sequences showing the performance gain from using our method to warp keyframe probability volumes.

To show the benefit of our method of regularisation, we compare the performance of the full system against three other regularisation schemes: using no optimisation at all (taking the depth values that maximise the discrete probability distribution), optimising without any regularisation (this will allow for the smoothing of the depth maps based on the continuous PDF, but provide no regularisation), and regularising using the total variation.

For the total variation, we tuned the hyperparameters of our system for the best performance ($\lambda = 1.0 \cdot 10^2$ and a step size of 0.05).

The results are presented in Table II. In all cases the best performance is achieved when using the surface normals and occlusion masks predicted by SharpNet.

Finally, to evaluate our method for warping probability volumes between keyframes, we compare our system against a version without warping where each keyframe is initialised only with the network output and does not receive any information from other keyframes.

The results are presented in Table III. Using our warping method improves the performance of the system in all cases.

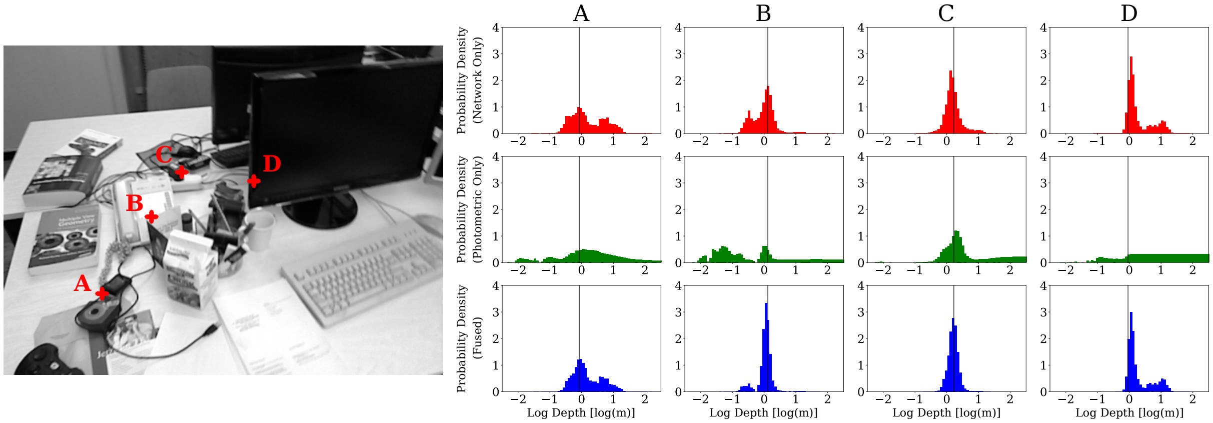

Fig. 5: This figure shows a grid of probability densities for a sample of four pixels from a keyframe (left). The first row, in red, shows the probability densities predicted by the network. The second row, in green, shows the probability densities estimated from the photometric error after the addition of 25 reference frames. The final row, in blue, shows the fused probability densities that results from our algorithm. Note that both the network and the photometric error are capable of producing multiple peaks. In some cases (such as pixel C), both the network and the photometric methods produce good estimates. In others (such as pixel A), both the network and photometric error are relatively uncertain, but together produce a strong peak. In pixels B and D, the network helps resolve ambiguous photometric peaks from either a repetition or lack of texture. The vertical black bars show the location of the ground truth depth.

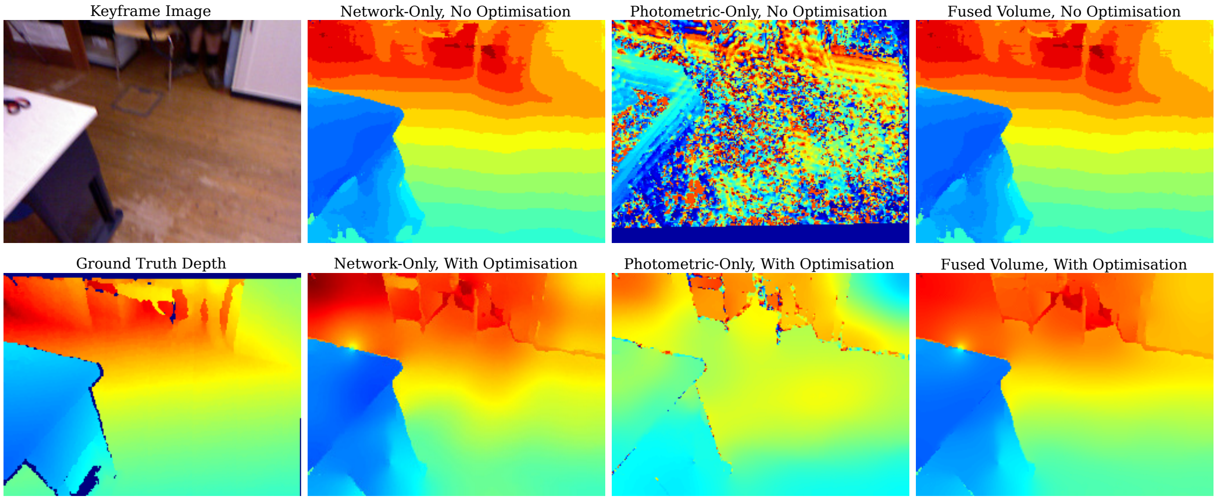
Keyframe Image
Network-Only, No Optimisation
Photometric-Only, No Optimisation
Fused Volume, No Optimisation
Ground Truth Depth
Network-Only, With Optimisation
Photometric-Only, With Optimisation
Fused Volume, With Optimisation

Fig. 6: Qualitative results from an example keyframe and 6 additional reference frames in the TUM RGB-D *fr1/360* sequence. The top left image is the keyframe image, and the bottom left is the ground truth depth. The remaining images on the top row are the depth estimates obtained by taking the maximum probability depth from each corresponding probability volume. The bands of colour show the quantisation that results from using this method. The remaining images in the bottom row are the depth estimates that result after performing the optimisation step. Note that the photometric error is only capable of estimating the depth at pixels with a high image gradient (the repeated edges are the result of pose error). While using only the network prediction results in a good reconstruction, the best reconstruction is obtained by fusing the network and photometric volumes together.

IV. Conclusion

We have presented a method for fusing learned monocular depth priors into a standard pipeline for 3D reconstruction. By training a DNN to predict nonparametric probability distributions, we allow the network to express uncertainty and make multi-hypothesis depth predictions.

Through a series of experiments, we demonstrated that by fusing the discrete probability volume predicted by the network with a probability volume computed from the photometric error, we achieve better performance than either on its own. Further experiments showed the value of our regularisation scheme and warping method.

References

[1] R. A. Newcombe, S. Lovegrove, and A. J. Davison, “DTAM: Dense Tracking and Mapping in Real-Time,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[2] M. Pizzoli, C. Forster, and D. Scaramuzza, “REMODE: Probabilistic, monocular dense reconstruction in real time,” in *Proceedings of the*

IEEE International Conference on Robotics and Automation (ICRA), 2014.

[3] A. Concha, W. Hussain, L. Montano, and J. Civera, "Manhattan and piecewise-planar constraints for dense monocular mapping," in *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[4] A. Concha and J. Civera, "DPPTAM: Dense Piecewise Planar Tracking and Mapping from a monocular sequence," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[5] A. Concha, G. Loianna, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[6] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," in *Neural Information Processing Systems (NIPS)*, 2014.

[7] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2016.

[8] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep Ordinal Regression Network for Monocular Depth Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and Motion Network for Learning Monocular Stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] H. Zhou, B. Ummenhofer, and T. Brox, "DeepTAM: Deep Tracking and Mapping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[14] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth Inference for Unstructured Multi-view Stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[15] J.-R. Chang and Y.-S. Chen, "Pyramid Stereo Matching Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] C. S. Weerasekera, Y. Latif, R. Garg, and I. Reid, "Dense Monocular Reconstruction using Surface Normals," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[17] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] N. Yang, R. Wang, J. Stückler, and D. Cremers, "Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[19] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "CodeSLAM – Learning a Compact, Optimisable Representation for Dense Visual SLAM," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning Depth from Monocular Videos using Direct Methods," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] T. Laidlow, J. Czarnowski, and S. Leutenegger, "DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM using Single-View Depth and Gradient Predictions," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[22] C. Tang and P. Tan, "BA-Net: Dense Bundle Adjustment Network," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[23] J. M. Fácil, A. Concha, L. Montesano, and J. Civera, "Single-view and multi-view depth fusion," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1994–2001, 2017.

[24] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[25] C. M. Bishop, "Mixture density networks," 1994.

[26] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" in *Neural Information Processing Systems (NIPS)*, 2017.

[27] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Using Multiple Hypotheses to Improve Depth-Maps for Multi-View Stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[28] V. Peretroukhin, B. Wagstaff, and J. Kelly, "Deep Probabilistic Regression of Elements of SO(3) using Quaternion Averaging and Uncertainty Injection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[29] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural RGB-D Sensing: Depth and Uncertainty From a Video Camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] A. Concha and J. Civera, "Using superpixels in monocular SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 365–372.

[31] C. S. Weerasekera, T. Dharmasiri, R. Garg, T. Drummond, and I. Reid, "Just-in-Time Reconstruction: Inpainting Sparse Maps using Single View Depth Predictors as Priors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3d reconstructions of indoor scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[34] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2012.

[35] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics (T-RO)*, vol. 33, no. 5, pp. 1255–1262, 2017.

[36] M. Ramamonjisoa and V. Lepetit, "SharpNet: Fast and Accurate Recovery of Occluding Contours in Monocular Depth Estimation," in *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2019.

[37] C. Loop, Q. Cai, P. Chou, and S. Orts-Escolano, "A closed-form bayesian fusion equation using occupancy probabilities," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2016.