

Deep Single and Direct Multi-View Depth Fusion

José M. Fácil, Alejo Concha, Luis Montesano and Javier Civera

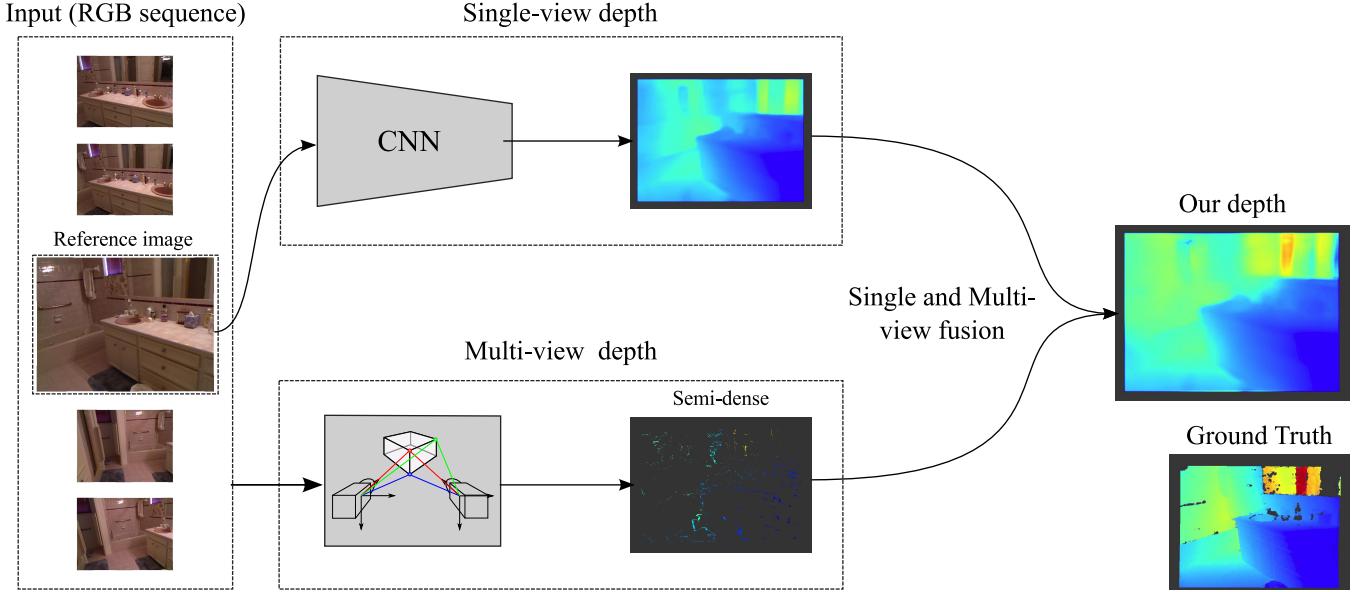


Fig. 1: Overview of our proposal. Our input is a set of overlapping monocular views. We estimate the learning-based single-view and geometry-based multi-view depth and fuse both information sources, outperforming both of them. All the depth images are color-normalized for better comparison. This figure is best viewed in color.

Abstract—Dense 3D mapping from a monocular sequence is a key technology for several applications and still a research problem. This paper leverages recent results on single-view CNN-based depth estimation and fuses them with direct multi-view depth estimation. Both approaches present complementary strengths. Multi-view depth estimation is highly accurate but only in high-texture and high-parallax cases. Single-view depth captures the local structure of mid-level regions, including textureless areas, but the estimated depth lacks global coherence.

The single and multi-view fusion we propose has several challenges. First, both depths are related by a non-rigid deformation that depends on the image content. And second, the selection of multi-view points of high accuracy might be difficult for low-parallax configurations. We present contributions for both problems. Our results in the public datasets of NYU and TUM shows that our algorithm outperforms the individual single and multi-view approaches.

I. INTRODUCTION

Estimating an online, accurate and dense 3D scene reconstruction from a general monocular sequence is one of the fundamental research problems in computer vision. The problem has nowadays a high relevance, as it is a key tech-

The authors are with the I3A, University of Zaragoza, Spain
`{jmfacil, alejocb, montesano, jcivera}@unizar.es`

nology in several emerging application markets (augmented and virtual reality, autonomous cars and robotics in general).

The state of the art are the so-called direct mapping methods [1], that estimate an image depth by minimizing a regularized cost function based on the photometric error between corresponding pixels in several views. The accuracy of the multi-view depth estimation depends mainly on three factors: 1) The geometric configuration, with lower accuracies for low-parallax configurations; 2) the quality of the correspondences among views, that can only be reliably estimated for high-gradient pixels; and 3) the regularization function, typically the Total Variation norm, that is inaccurate for large textureless areas. Due to this poor performance on large low-gradient areas, semi-dense maps are sometimes estimated only in high-gradient image pixels for visual direct SLAM (e.g., [2]). Such semi-dense maps are accurate in high-parallax configurations but not a complete model of the viewed scene.

An alternative method is single-view depth estimation, which has recently experienced a qualitative improvement in its accuracy thanks to the use of deep convolutional networks [3]. Their accuracy is still lower than that of multi-view methods for high-texture and high-parallax points. But, as

we will argue in this paper, they improve the accuracy of multi-view methods in low-texture areas due to the high-level feature extraction done by the deep networks –opposed to the low-level high-gradient pixels used by the multi-view methods. Interestingly, the errors in the estimated depth are locally and not globally correlated since they are obtained from different image features.

The main idea of this paper is to exploit the information of single and multi-view depth maps to obtain an improved depth even in low-parallax sequences and in low-gradient areas. Our contribution is an algorithm that fuses these complementary depth estimations. There are two main challenges in this task. First, the error distribution of the single-view estimation has several local modes, as it depends on the image content and not on the geometric configuration. Single and multi-view depth are hence related by a content-dependent non-rigid deformation. Secondly, as we aim to a general method for high and low-parallax cases, modeling the multi-view accuracy is not trivial. We propose a method based on a weighted interpolation of the single-view local structure based on the quality and influence area of the multi-view semi-dense depth and evaluate its performance in two public datasets –NYU and TUM. The results show that our fusion algorithm improves over both individual single and multi-view approaches.

The rest of the paper is organized as follows. Section II describes the most relevant related work. Section III motivates and details the proposed algorithm for single and multi-view fusion. Section IV presents our experimental results and, finally, section V contains the conclusions of this work.

II. RELATED WORK

A. Multi-View Depth

[1], [4], [5] are the first works that achieved dense and real-time reconstructions from monocular sequences. Some of the most relevant aspects are the direct minimization of the photometric error –instead of the traditional geometric error of sparse reconstructions– and the regularization of the multiview estimation by adding the total variation (TV) norm to the cost function.

[6] showed that a TV regularization has low accuracy for large textureless areas and proposed a piecewise-planar regularization; the plane parameters coming from multi-view superpixel triangulation [7] or layout estimation [8]. [9] proposes higher-order regularization terms that enforce piecewise affine constraints even in separated pixels. [10] selects the best regularization function among a set using sparse laser data. [11] adds the sparse data-driven 3D primitives of [12] as a regularization prior. Compared to these works, our fusion is the first one where the information added to the multi-view depth is fully dense, data-driven and single-view; and hence it does not rely on additional sensors, parallax or Manhattan and piecewise-planar assumptions.

Due to the difficulty of estimating an accurate and fully dense map from monocular views there are several approaches that estimate only the depth for the highest-gradient

	High-Gradient	Low-Gradient
Multi-View	0.25	0.79
Single-View	0.60	0.18

TABLE I: Median depth error, in meters, for single and multi-view algorithms, and high and low-gradient pixels.

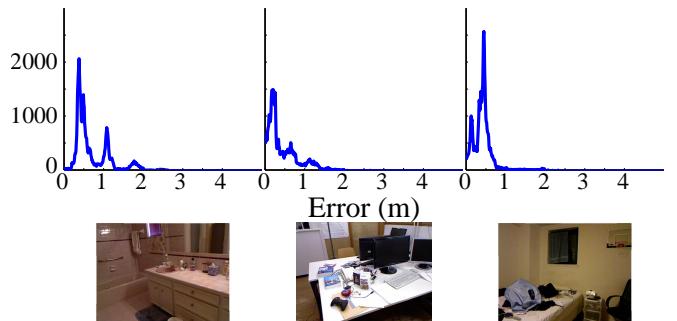


Fig. 2: Histogram of single-view depth error [m] for three sample sequences. Notice the multiple modes, each one corresponding to a local image structure.

pixels (e.g., [2]). While this approach produces maps of higher density than the more traditional feature-based ones (e.g., [13]), they are still uncompleted models of the scene and hence their applicability might be more limited.

B. Single-View Depth

Depth can be estimated from a single view using different image cues, for example from focus (e.g., [14]) or perspective cues (e.g., [15]). Learning-based approaches, as the one we use, basically discover the RGB patterns that are relevant for accurate depth regression.

The pioneering work of Saxena *et al.* [16] trained a MRF to model depth from a set of global and local image features. Very recently, Eigen *et al.* [17] presented a two deep convolutional neural network (CNN) stacked, one to predict global depth prediction and the second one refines it locally. Build upon this method [3], recently presented a three scale convolutional network to estimate depth, surface normals and semantic labeling. Liu *et al.* [18] use a unified continuous CRF-and-CNN framework to estimate depth. The CNN is used to learn the unary and pairwise potentials that the CRF uses for depth prediction.

III. SINGLE AND MULTI-VIEW DEPTH FUSION

State-of-the-art multi-view techniques have a strong dependency on high-parallax sequences and heterogeneous-texture scenes. Only a reduced set of salient pixels that hold both constraints has a small error, and the error for the majority of the points is large and uncorrelated. In contrast, single-view methods based on CNN networks achieve reasonable errors in all the image but they are locally correlated. Our proposal exploits the best properties of these two methods. Specifically, it uses a deep convolutional network (CNN) to produce rough depth maps and fuses their structure with the results of a semi-dense multi-view depth method (Fig. 1).

Before delving into the technical aspects, we will motivate our proposal with some illustrative results. Table I shows the median depth error of the high-gradient and low-gradient pixels for a multi-view and single view reconstruction using a high-parallax sequence of the TUM dataset. For the multi-view reconstruction, the error for the low-gradient pixels increases by a factor of 3. Notice that the opposite happens for the single-view reconstruction: the error of high-gradient pixels is the one increasing by a factor of 3.

Furthermore, the single-view depth error usually has a structure that indicates the presence of local correlations. For instance, Fig. 2 shows the histogram of the single-view depth estimation error for three different sequences (two of the NYU dataset and one of the TUM dataset). Notice that the error distribution is grouped in different modes, each one corresponding to an image segment. This effect is caused by the use of the high-level image features of the latest layers of the CNN network, that extend over dozens of pixels in the original image and hence over homogeneous texture areas. The different nature of the errors can be exploited to outperform both individual estimations. However, fusion cannot be implemented based on a single global model but requires non-rigid deformations.

In the next subsections we detail the specific multi and single-view methods that we use in this work and our fusion algorithm.

A. Multi-view Depth

For the estimation of the multi-view depth we adopt a direct approach [2], that allows us to estimate a dense or semi-dense map in contrast to the more sparse maps of the feature-based approaches. In order to estimate the depth of a keyframe \mathcal{I}_k we first select a set of n overlapping frames $\{\mathcal{I}_1, \dots, \mathcal{I}_o, \dots, \mathcal{I}_n\}$ from the monocular sequence. After that, every pixel x_l^k of the reference image \mathcal{I}_k is first backprojected at an inverse depth ρ and projected again in every overlapping image \mathcal{I}_o .

$$x_l^o = T_{ko}(x_l^k, \rho_l) = KR_{ko}^\top \left(\begin{pmatrix} K^{-1}x_l^k \\ ||K^{-1}x_l^k|| \\ \rho_l \end{pmatrix} - t_{ko} \right), \quad (1)$$

where T_{ko}, R_{ko} and t_{ko} are respectively the relative transformation, rotation and translation between the keyframe \mathcal{I}_k and every overlapping frame \mathcal{I}_o . K is the camera internal calibration matrix.

We define the total photometric error $C(\rho)$ as the summation of every photometric error ϵ_l between every pixel (or every high-gradient pixel if we want a semi-dense map) x_l^k in the reference image \mathcal{I}_k and its corresponding one x_l^o in every other overlapping image \mathcal{I}_o at an hypothesized inverse depth ρ_l ,

$$C(\rho) = \frac{1}{n} \sum_{o=1, o \neq k}^n \sum_{l=1}^t \epsilon_l(\mathcal{I}_k, \mathcal{I}_o, x_l^k, \rho_l). \quad (2)$$

The error $\epsilon_l(\mathcal{I}_k, \mathcal{I}_o, x_l^k, \rho_l)$ for each individual pixel x_l^k is the difference between the photometric values of the pixel

and its corresponding one

$$\epsilon_l(\mathcal{I}_k, \mathcal{I}_o, x_l^k, \rho_l) = \mathcal{I}_k(x_l^k) - \mathcal{I}_o(x_l^o). \quad (3)$$

The estimated depth for every pixel $\hat{\rho} = (\hat{\rho}_1 \dots \hat{\rho}_l \dots \hat{\rho}_t)^\top$ is obtained by the minimization of the total photometric error $C(\rho)$:

$$\hat{\rho} = \arg \min_{\rho} C(\rho) \quad (4)$$

B. Single-view Depth

For single-view depth estimation we use the Deep Convolutional Neural Network presented by Eigen *et al.*, [3]. This network uses three stacked CNN to process the images in three different scales. The input to the network is the RGB keyframe \mathcal{I}_k . As we use the network structure and parameters released by the authors without further training, our input image size is 320×240 . The output of the network is the predicted depth, that we will denote as s . The size of the output is 147×109 , that we upsample in our pipeline in order to fuse it with the multi-view depth.

The first scale CNN extract high-level features tuned for depth estimation. This CNN produces 64 feature maps of size 19×14 that are the input, along with the RGB image, of the second scale CNN. This second stacked CNN refines the output of the first one with mid-level features to produce a first coarse depth map of size 74×55 . This depth map is upsampled and feeds a third stacked CNN that does a local refinement of the depth. This final step is necessary, as the convolution and pooling steps of the previous layers filter out the high-frequency details.

The first scale was initialized with two different pre-trained networks: the AlexNet [19] and the Oxford VGG [20]. We use the VGG version, the most accurate one as reported by the authors. This network has been trained in indoor scenes with the NYUDepth v2 dataset [21]. As they used the official train/test splits of the dataset, so do we. We decided to use this neural network because it is the best-performing dense single-view method. We refer the reader to the original work [3] for more details on this part of our pipeline.

C. Non-rigid Fusion

As we mentioned before, the objective is to fuse the output of each previous method while keeping the best properties of each of them: the single-view reliable local structure and the accurate, but semi-dense multi-view depth estimation. Let denote s and m to the single-view depth and the multi-view semi-dense depth estimation, respectively. s is predicted as detailed in section III-B and $m = \frac{1}{\rho}$ is the inverse of the inverse depth estimated in section III-A.

The fused depth estimation f_{ij} for the pixel (i, j) is computed as a weighted interpolation of depths over the set of pixels in the multi-view depth image

$$f_{ij} = \sum_{(u,v) \in \Omega} W_{s_{ij}}^{m_{uv}} (m_{uv} + (s_{ij} - s_{uv})), \quad (5)$$

where Ω is the set of pixels estimated by the multi-view algorithm. The interpolation weights $W_{s_{ij}}^{m_{uv}}$ model the likelihood for each pixel $(u, v) \in \Omega$ belonging to the same local

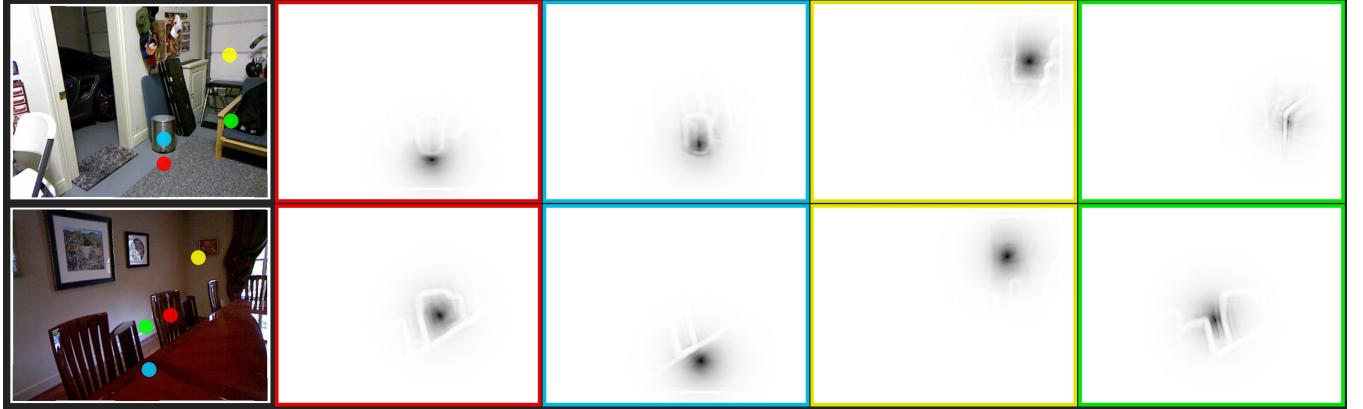


Fig. 3: Non-normalized influence of the weight for each one of the points in the image. *First column:* RGB input image with four different situated points in the image. *Second, third, fourth and fifth columns:* influence of each point highlighted in the RGB image. Figure best viewed in color.

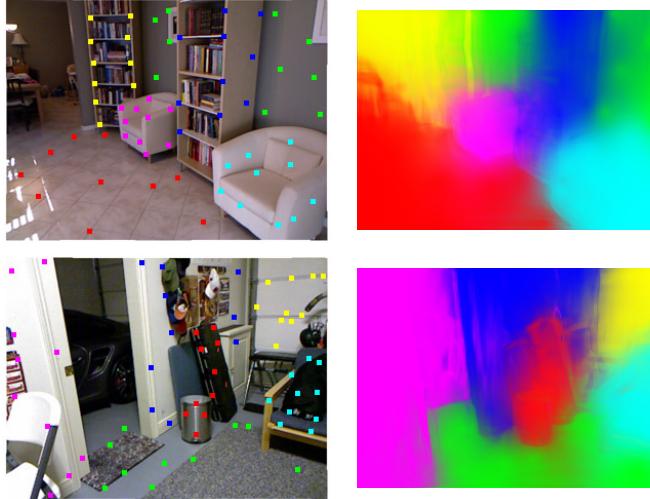


Fig. 4: Normalized influence area of the weights. Notice how it expands around local structure areas given a set of points in Ω . *First column:* RGB image with the points of Ω labeled with different colors. *Second column:* influence areas computed by our method. Figure best viewed in color.

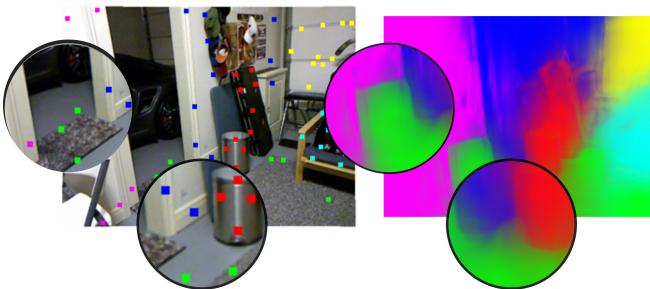


Fig. 5: Detail of the influence area. Notice how it expands mainly in the areas with same local structure. Figure best viewed in color.

structure as pixel (i, j) . The interpolation can be interpreted in two ways. First, the depth gradient $(s_{ij} - s_{uv})$ is added

to each multi-view depth m_{uv} , i.e. we create depth map for each m_{uv} with the structure of s and then weigh them with pixel based weights. Second, for each depth s_{ij} we modify it according to the weighed discrepancy between $(m_{uv} - s_{uv})$.

The key ingredient of this interpolation are the weights $W_{s_{ij}}^{m_{uv}}$ that need to achieve the non-rigid transformation mentioned above based on the local structure of the image. Each weight is computed as the product of four different factors. The first factor

$$\tilde{W}_{1s_{ij}}^{m_{uv}} = e^{-\frac{\sqrt{(i-u)^2 + (j-v)^2}}{\sigma_1}}, \quad (6)$$

simply measures proximity based on the distance of the pixels (i, j) and (u, v) . The parameter σ_1 controls the radius of proximity for each point. The remainder three factors depend on the structure of the single-view prediction s . The second factor

$$\begin{aligned} \tilde{W}_{2s_{ij}}^{m_{uv}} &= \frac{1}{|\nabla_x s_{uv} - \nabla_x s_{ij}| + \sigma_2} \\ &\cdot \frac{1}{|\nabla_y s_{uv} - \nabla_y s_{ij}| + \sigma_2} \end{aligned} \quad (7)$$

measures the similarity of depth gradients and assigns larger weights to similar ones. $\nabla_x s_{ij}$ and $\nabla_y s_{ij}$ represent the depth gradient in the x and y direction respectively at the pixel (i, j) . σ_2 limits the influence of a point to avoid extremely high weights for very similar or identical gradients. We set it to 0.1 in the experiments. The last two factors strengthen the influence between the points lying in the same plane and are defined as

$$\tilde{W}_{3s_{ij}}^{m_{uv}} = e^{-|(s_{ij} + \nabla_x s_{ij} \cdot (i-u)) - s_{uv}|} + \sigma_3 \quad (8)$$

and

$$\tilde{W}_{4s_{ij}}^{m_{uv}} = e^{-|(s_{ij} + \nabla_y s_{ij} \cdot (j-v)) - s_{uv}|} + \sigma_3, \quad (9)$$

where σ_3 sets a minimum weight to any point in Ω . This is required to avoid vanishing weights when they are combined with $\tilde{W}_{1s_{ij}}^{m_{uv}}$ and $\tilde{W}_{2s_{ij}}^{m_{uv}}$.

Sequence	MEAN ERROR (m)			GAIN w.r.t TV	GAIN w.r.t [3]	MEAN ERROR (m)
	TV	Eigen[3]	Ours(auto)	Ours(auto)	Ours(auto)	
NYUDepth v2	bathroom_0018	1.612	0.723	0.627	61.12%	13.33%
	bedroom_0013	0.918	0.421	0.342	62.73%	18.80%
	dining_room_0032	0.991	0.342	0.399	59.86%	-16.39%
	kitchen_0032	0.941	0.583	0.581	38.29%	0.37%
	living_room_0025	0.828	0.440	0.394	52.47%	10.55%
	living_room_0030a	1.472	0.572	0.487	66.91%	14.87%
TUM	fr1_desk	0.605	0.437	0.358	40.82%	18.07%
	fr1_room	0.613	0.216	0.202	67.04%	6.48%
						0.229
						0.131

TABLE II: *Left table*: Mean error for the TV-regularized multi-view depth, the single-view depth [3] and our fused depth. The last two columns show our improvement with respect to the two individual methods respectively. *Right table*: Mean error for the fused depth with manual multi-view point selection.

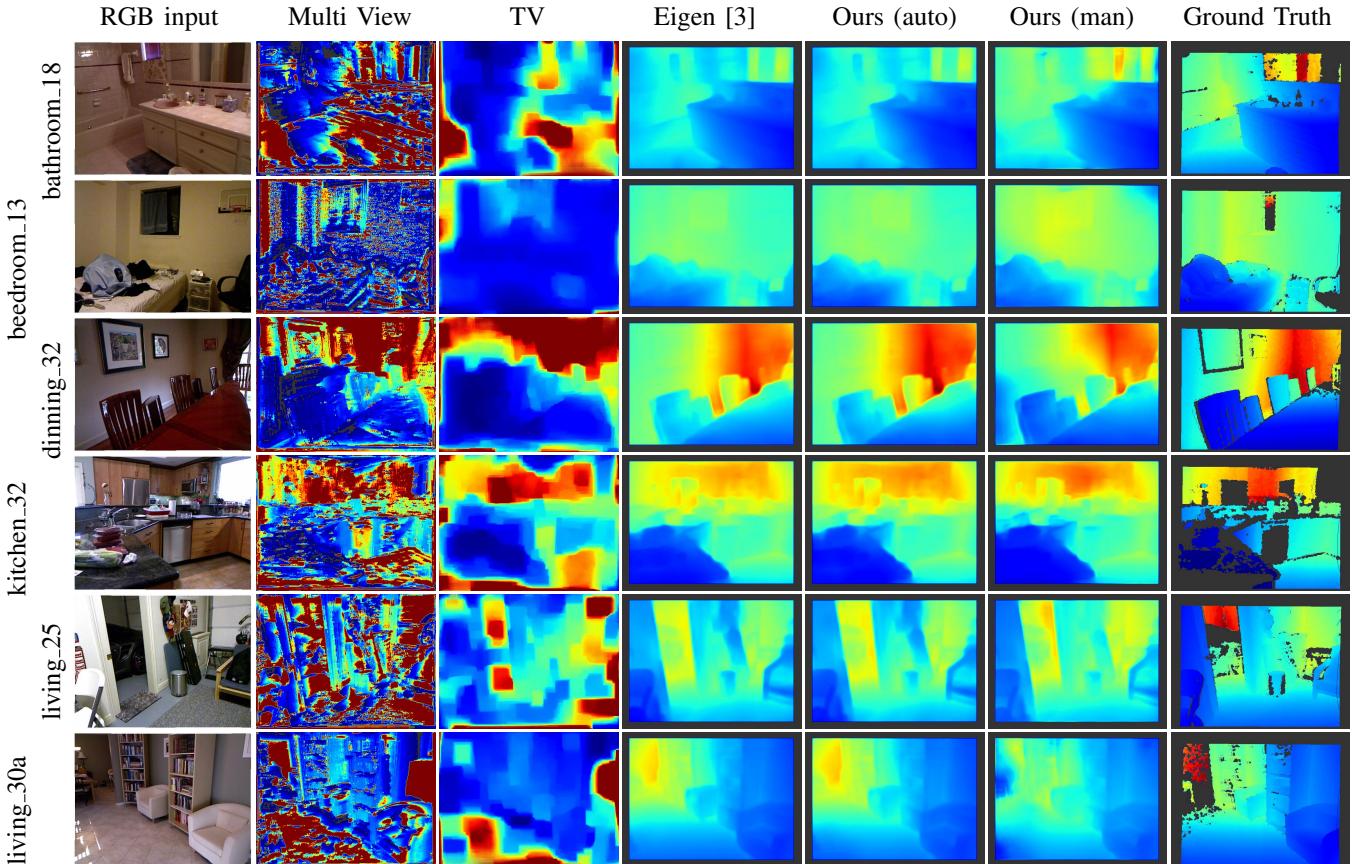


Fig. 6: Depth images for the NYU Depth v2 dataset [21]. Color ranges are row-normalized to facilitate the comparison between different methods. *First column* RGB keyframe, *second column* multi-view depth, *third column* TV-regularized multi-view depth, *fourth column* single-view depth, *fifth column* our depth fusion with automatic multi-view point selection, *sixth column* our depth fusion with manual multi-view point selection, and *seventh column* ground truth. Figure best viewed in color.

The product of this four factor makes an non-normalized weight for each pixel in Ω

$$\tilde{W}_{m_{uv}}^{s_{ij}} = \prod_{n=1}^4 \tilde{W}_{n m_{uv}}^{s_{ij}} \quad (10)$$

and represents its area of influence. The parameters σ_1 , σ_2 and σ_3 shape the area of influence and have to be selected to balance proximity, gradient and planarity and to avoid discontinuities in the result of the fusion. This was done empirically on small set of three images. The values of the

parameters were 15, 0.1 and $1e-3$, respectively. These values were fixed for all the experiments suggesting that they do not require image or sequence specific tuning.

Fig. 3 shows this area for some points on two different images. Notice how the influence expands around the point but is kept inside the same local structure. Once all the factors has been computed for all the pixels in Ω , we normalize the weights for each single-view pixel so all the weights over a

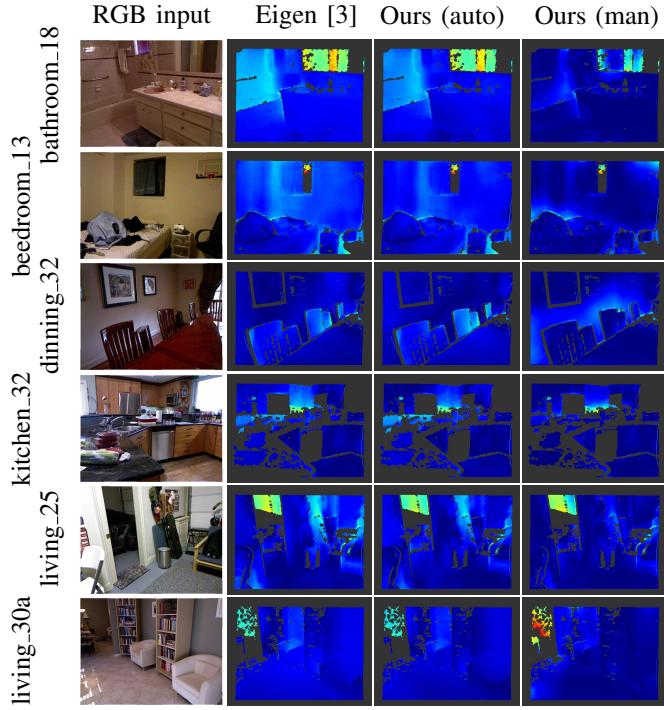


Fig. 7: Error images on sequences of the NYU Depth v2 dataset [21]. Color ranges are row-normalized to facilitate the comparison between different methods. Darker blue is better. *First column* RGB keyframe, *second column* single-view depth, *third column* our depth fusion with automatic multi-view point selection, *sixth column* our depth fusion with manual multi-view point selection.

pixel (i, j) sum 1.

$$W_{sij}^{m_{uv}} = \frac{\tilde{W}_{sij}^{m_{uv}} - \min_{(g,h) \in \Omega} \tilde{W}_{sij}^{m_{gh}}}{\sum_{(p,k) \in \Omega} \tilde{W}_{sij}^{m_{pk}} - \min_{(g,h) \in \Omega} \tilde{W}_{sij}^{m_{gh}}} \quad (11)$$

The normalized weights expand the local influence to the whole image (see Fig. 4 and Fig. 5 for a more detailed view). Note how the influence get expanded along planes even if the points in Ω do not reach the end of the plane and get cut when local structure changes. Once these influence weights have been calculated and normalized, the fusion depth estimation, f , for each point is a combination of all the selected points in Ω , as presented in Eq. 5.

D. Multi-view Low-Error Point Selection

Up to now we have assumed that all the points in the multi-view semi-dense depth have low error. This is easily achievable in high-parallax sequences by using robust estimators –robust cost functions or RANSAC schemes. However, it is problematic for the degenerate or quasi-degenerate low-parallax geometries that we also target in this paper. In this case, multi-view depths may contain large errors that will propagate to the fused depth map and it is necessary to filter them out. Unexpectedly, selecting high gradient pixels was not robust enough to remove points with large depth errors and we have developed a two step algorithm that takes into

account photometric and geometric information in the first step and the single-view depth map in the second one.

The first step selects a fixed percentage of the best correspondence candidates –the best 25% in our experiments– based on the product of a photometric score and geometric one. On one hand, the photometric criterion focuses on the quality of the correspondences using image information. We apply a modified version of the *second best ratio*. We compute the two closest matches for a pixel (smallest photometric errors according to Eq. 3), and we compute the score as a function of the ratio between the distance of the two descriptors (a low ratio meaning a clear match) and the gradient of the distance function along the epipolar (i.e. the function presents a clear *V-shape* around this match indicating that the match is spatially accurate). On the other hand, the geometric score simply backpropagates the image correspondence error to the depth estimation, resulting in low scores for low-parallax correspondences.

In a second stage we also use the structure of the single-view reconstruction and apply RANSAC to estimate a spurious-free linear transformation between the multi and single-view points using only the points pre-filtered in the first stage. This reduces further the number of spurious depth values from the multi-view algorithm. The result is a small set of low-error points that we use for the non-rigid deformation of the previous section. As mentioned before, in our experiments this algorithm behaves better than a geometric compatibility test, specially in the low-parallax sequences of the NYU dataset.

IV. EXPERIMENTAL RESULTS

In this section we evaluate the algorithm and compare its performance against two state-of-the-art methods: multi-view direct mapping using TV regularization (implemented following [1], [23]) and the single-view depth estimation using the network of [3]. We have selected two datasets with different properties. The first one is the NYUv2 Depth Dataset [21], a general dataset aimed at image segmentation evaluation and hence likely to contain low-parallax and low-texture sequences. We analyze results in six sequences from the test set (i.e. the single-view net had not been trained on these sequences) selected just to include different types of rooms. The second one is the TUM RGB-D SLAM Dataset [22], a dataset oriented to visual SLAM and then likely to present a bias benefitting multi-view depth. In this case, we evaluated two sequences selected randomly.

We run our algorithm in a 320×240 subsampled version of the images, as this is the size of the single-view neural network given by the authors. We also run our multi-view depth estimation at this image size, and upsample the fused depth to 640×480 in order to compare it against the ground truth D channel from the Kinect camera.

As our aim is to evaluate the mapping accuracy, we will assume that camera poses are known for the multiview estimation. In the TUM RGB-D SLAM Dataset [22] we use the ground truth camera poses provided by the dataset. In the NYUv2 Depth Dataset sequences we estimate them using

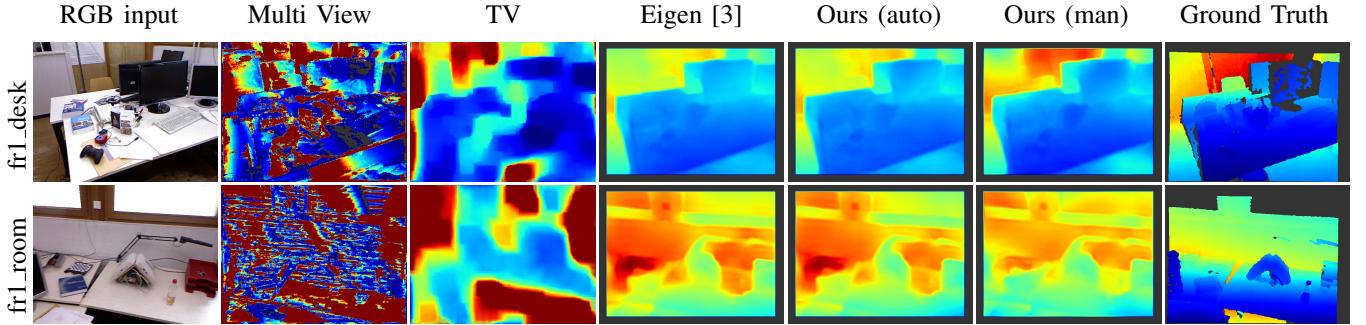


Fig. 8: Depth images on sequences of the TUM Dataset [22]. Color ranges are row-normalized to facilitate the comparison between different methods. *First column* RGB keyframe, *second column* multi-view depth, *third column* TV-regularized multi-view depth, *fourth column* single-view depth, *fifth column* our depth fusion with automatic multi-view point selection, *sixth column* our depth fusion with manual multi-view point selection, and *seventh column* ground truth. Figure best viewed in color.

the RGB-D Dense Visual Odometry by Gutierrez-Gomez *et al.* [24]. These camera poses will remain fixed and used to create the multi-view depth maps. As mentioned before, the parameters of the fusion algorithm were experimentally set prior to the evaluation on a small separate set of images.

To evaluate the methods, we computed the mean depth error across the whole image for each method. The results are summarized in Table II. Our method outperforms the TV regularization in both datasets obtaining an average improvement over 50%. As expected, the TV regularization performs better in the TUM sequences and achieves lower errors, but in terms of improvement there seems not to be big differences between both datasets. Our fusion of depths also outperforms the single-view depth reconstruction. However, in this case the gain is not as large and is close to 10% on average. Both methods perform similarly in both datasets, but except in one sequence, our method is always better or as good as the deep single-view reconstruction.

The right-most column of Table II shows the depth errors when the set of multi-view points does not contain outliers. We selected them using the ground-truth data from the D channel, and keeping only those points whose depth error was lower than 10cm. The results are for all sequences better than any method attaining improvements around 70% and 38% with respect to TV and [3], respectively. Although expected, this result highlights the impact of outliers and the need for good point selection. It also provides an upper bound and shows that there is still room for improvement in this latest part of our algorithm.

Finally, we present the results of some randomly picked images for each sequence of each dataset. Fig. 6 and Fig. 8 show the obtained depth images for the NYUDepth v2 and the TUM datasets, respectively. The improvement with respect to the regularized multi-view approach is clear visually since the depth structure is much more consistent. Improvements with respect to single-view images are more subtle and are best viewed by looking at the corresponding depth error images of Fig. 7 and Fig. 9. Usually, the improvement comes from a better relative placement of some

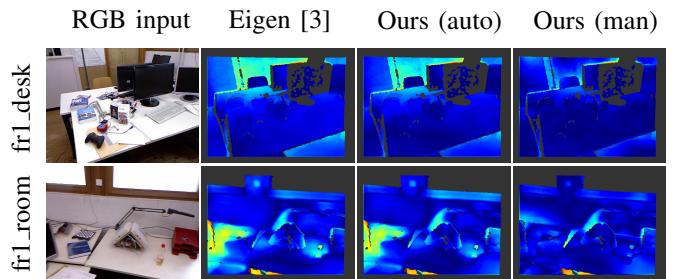


Fig. 9: Error images for sample sequences of the TUM Dataset [22]. Color ranges are row-normalized to facilitate the comparison between different methods. Darker blue is better. *First column* RGB keyframe, *second column* single-view depth, *third column* our depth fusion with automatic multi-view point selection, *sixth column* our depth fusion with manual multi-view point selection.

local structure. For instance, the walls are darker in the error images (see the bathroom_18 or dining_32 in Fig. 7). The effect is more evident when the multi-view points were selected based on the ground truth, but is the same effect as the one obtained with the automatic selection. This better local structure alignment reduces the error as can be seen in the boxplots of each sequence shown in Fig. 10.

V. CONCLUSIONS

In this paper we have presented an algorithm for dense depth estimation by fusing 1) the multi-view depth estimation from a direct mapping method, and 2) the single-view depth that comes from a deep convolutional network trained on RGB-D images. Our approach selects a set of the most accurate points from the multi-view reconstruction and fuses them with the dense single-view estimation. It is worth remarking that the single-view depth errors do not depend on the geometric configuration but on the image content and hence the transformation is non-rigid and varies locally. The estimation of this alignment is our main contribution and the most challenging aspect of this research.

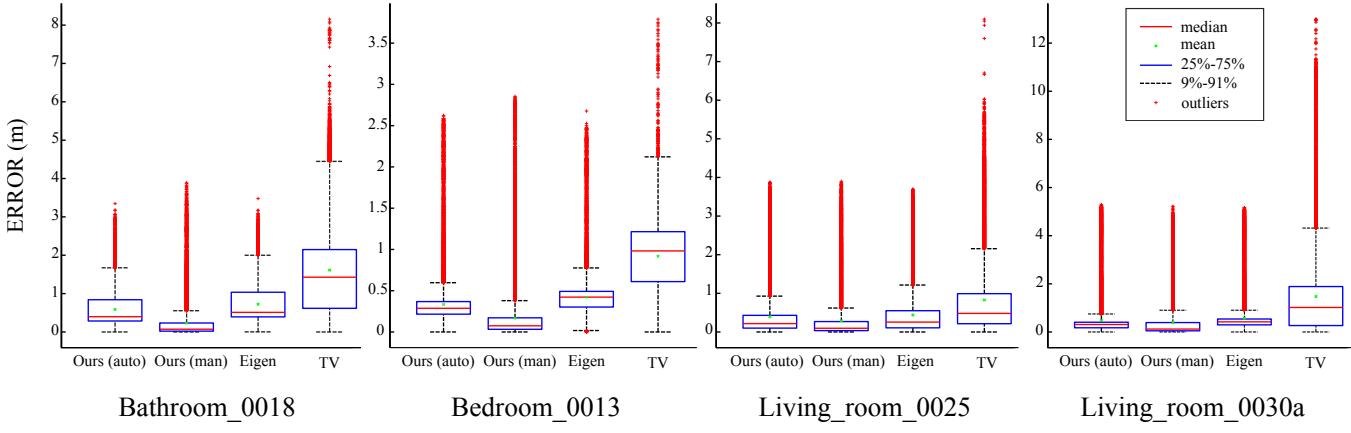


Fig. 10: Box and Whiskers plots of the pixel error distribution for four of the scenes we have tested on. From left to right we compare our method with automatic point selection, our method with manual point selection, single-view method proposed by Eigen *et al.* [3] and TV regularization for multi-view estimation.

Our experiments show that our proposal improves over the state of the art (Eigen *et al.* [17] for single-view depth and direct mapping plus TV regularization for multi-view depth). Contrary to other approaches, the single-view depth we use is entirely data-driven and hence does not rely on any scene assumption. Future work will, as suggested by the results, try to improve the multi-view points selection and the fusion of both images using, for instance, iterative procedures.

ACKNOWLEDGMENT

We gratefully acknowledge the support of NVIDIA Corporation for the donation of a Titan X GPU, the Spanish government (projects DPI2012-32168 and DPI2015-67275), the Aragon regional government (Grupo DGA T04-FSE) and the University of Zaragoza (JIUZ-2015-TEC-03).

REFERENCES

- [1] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2320–2327, IEEE, 2011.
- [2] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.
- [3] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658, 2015.
- [4] G. Graber, T. Pock, and H. Bischof, “Online 3d reconstruction using convex optimization,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 708–711, IEEE, 2011.
- [5] J. Stühmer, S. Gumhold, and D. Cremers, “Real-time dense geometry from a handheld camera,” in *Joint Pattern Recognition Symposium*, pp. 11–20, Springer, 2010.
- [6] A. Concha, W. Hussain, L. Montano, and J. Civera, “Manhattan and Piecewise-Planar Constraints for Dense Monocular Mapping,”
- [7] A. Concha and J. Civera, “Using superpixels in monocular SLAM,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 365–372, IEEE, 2014.
- [8] V. Hedau, D. Hoiem, and D. Forsyth, “Recovering the spatial layout of cluttered rooms,” in *2009 IEEE 12th international conference on computer vision*, pp. 1849–1856, IEEE, 2009.
- [9] P. Piniés, L. M. Paz, and P. Newman, “Dense mono reconstruction: Living with the pain of the plain plane,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 5226–5231, IEEE, 2015.
- [10] P. Piniés, L. M. Paz, and P. Newman, “Too much TV is bad: Dense reconstruction from sparse laser with non-convex regularisation,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 135–142, IEEE, 2015.
- [11] A. Concha, W. Hussain, L. Montano, and J. Civera, “Incorporating scene priors to dense monocular mapping,” *Autonomous Robots*, vol. 39, no. 3, pp. 279–292, 2015.
- [12] D. F. Fouhey, A. Gupta, and M. Hebert, “Data-driven 3d primitives for single image understanding,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3392–3399, 2013.
- [13] R. Mur-Artal, J. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *Robotics, IEEE Transactions on*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [14] J. Ens and P. Lawrence, “An investigation of methods for determining depth from focus,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 2, pp. 97–108, 1993.
- [15] P. Sturm and S. Maybank, “A method for interactive 3d reconstruction of piecewise planar objects from single images,” in *The 10th British machine vision conference (BMVC’99)*, pp. 265–274, The British Machine Vision Association (BMVA), 1999.
- [16] A. Saxena, M. Sun, and A. Y. Ng, “Make3D: Learning 3D scene structure from a single still image,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2009.
- [17] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Advances in Neural Information Processing Systems*, pp. 2366–2374, 2014.
- [18] F. Liu, C. Shen, and G. Lin, “Deep convolutional neural fields for depth estimation from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170, 2015.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [21] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from RGBD Images,” in *ECCV*, 2012.
- [22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
- [23] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison, “Applications of legendre-fenchel transformation to computer vision problems,” *Department of Computing at Imperial College London. DTR11-7*, vol. 45, 2011.
- [24] D. Gutiérrez-Gómez, W. Mayol-Cuevas, and J. Guerrero, “Inverse Depth for Accurate Photometric and Geometric Error Minimisation in RGB-D Dense Visual Odometry,”