3D Scene Mesh From CNN Depth Predictions And Sparse Monocular SLAM

Tomoyuki Mukasa Jiu Xu Björn Stenger

Rakuten Institute of Technology

Abstract

In this paper, we propose a novel framework for integrating geometrical measurements of monocular visual simultaneous localization and mapping (SLAM) and depth prediction using a convolutional neural network (CNN). In our framework, SLAM-measured sparse features and CNN-predicted dense depth maps are fused to obtain a more accurate dense 3D reconstruction including scale. We continuously update an initial 3D mesh by integrating accurately tracked sparse features points. Compared to prior work on integrating SLAM and CNN estimates [26], there are two main differences: Using a 3D mesh representation allows as-rigid-as-possible update transformations. We further propose a system architecture suitable for mobile devices, where feature tracking and CNN-based depth prediction modules are separated, and only the former is run on the device. We evaluate the framework by comparing the 3D reconstruction result with 3D measurements obtained using an RGBD sensor, showing a reduction in the mean residual error of 38% compared to CNN-based depth map prediction alone.

1. Introduction

Computer vision has long been successfully employed to track camera motion and reconstruct 3D structure from image sequences. These methods have been applied *e.g.* in the visual effects industry [6], the robot vision community [4], or various types of 3D reconstruction from a large scale [1] to a small scale on a mobile device [14]. In the past decade, mobile applications for augmented reality (AR) and mixed reality (MR) have become ubiquitous. Since most current mobile devices come with a single (back-facing) camera, these applications rely on monocular visual SLAM to recover camera pose [5, 20]. Visual SLAM estimates depth from small-baseline stereo matching over pairs of nearby frames. This assumes that the camera translates in space over time, so that pairs of consecutive frames are equivalent to the pairs of frames captured using a stereo rig. Traditionally, visual SLAM has relied on 2D feature matching due to its efficiency and robustness in scenes with sufficient texture [5, 20]. Recent methods that employ contour information or the whole image have been shown to add robustness, but come with higher computational cost [9, 11, 22]. A known limitation of monocular SLAM is that it cannot estimate the scale of the scene. This can be estimated by including additional information, such as data from an inertial measurement unit (IMU) [19], or prior knowledge about object sizes [29]. Another shortcoming is that monocular visual SLAM is ill-conditioned for certain camera motions like rotation without translation.

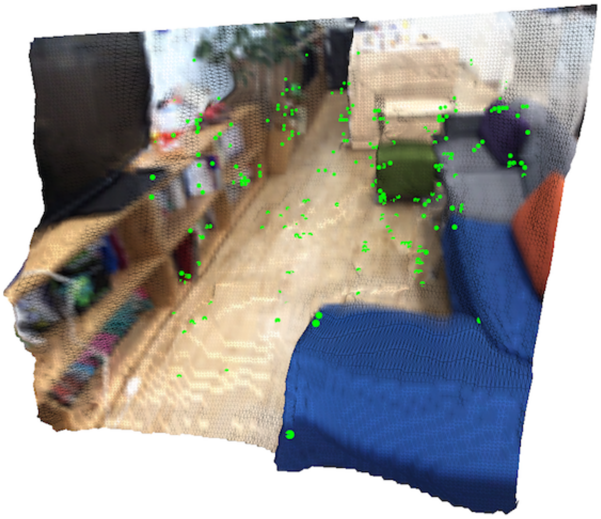

Figure 1. **3D Scene Mesh.** *Result of our method, reconstructed from a CNN-predicted mesh, deformed using 3D points obtained by SLAM, which are indicated as light green dots.*

Recently, neural networks have been shown to provide good predictions of geometry, *i.e*. depth and normals, from a given input image [7, 8, 15]. An end-to-end trained CNN is able to predict geometry densely, even for less textured areas. Unlike the geometry from monocular SLAM, the depth map includes an absolute scale, as learned from the training examples. One drawback of current methods is that occlud-

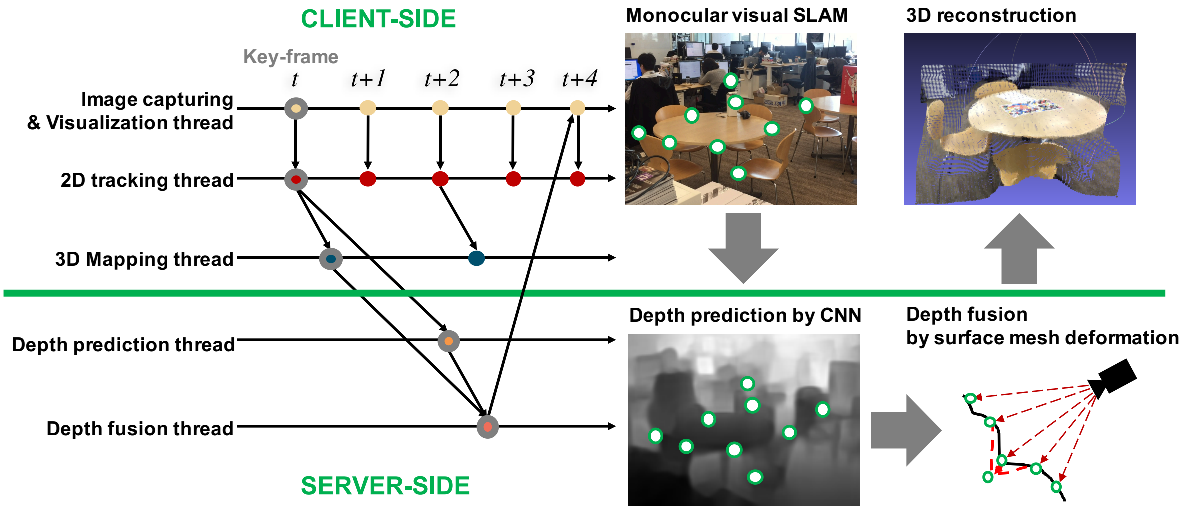

Figure 2. **System Overview.** *Monocular SLAM is run on the mobile device. Images and 3D points are sent to the server side, where CNN depth prediction and surface mesh deformation is carried out and is sent back to the client for visualization.*

ing boundary regions tend to be overly smooth and shape details are lost.

Recent work on combining 3D SLAM measurements and depth predictions from a CNN has shown that both sources can complement each other [26], see Table 1. In this paper we propose a framework to fuse monocular SLAM with CNN-based depth predictions in a new way: Feature point-based ORB-SLAM is run on the mobile device, yielding sparse but accurate 3D points. Depth maps are asynchronously predicted on a server and converted to a mesh representation. The mesh is then deformed, in an as-rigid-as-possible manner, using the sparse, but accurate feature points, and the updated mesh sent to the device. This approach corrects both coarse global geometric errors and reintroduces some shape details, see Figure 1. We evaluate the method on challenging office scenes, comparing the result with depth-sensor ground truth.

2. Related work

We review three areas of related work, monocular visual SLAM, CNN-based depth prediction, and surface mesh deformation.

Monocular visual SLAM can be classified into two categories [30], feature-based [13, 20] and direct approaches [9, 22]. One of the state-of-the-art methods in the feature-based category is ORB-SLAM[20]. The method extracts sparse ORB features and reconstructs them in 3D using bundle adjustment and pose graph optimization. In contrast, direct methods carry out pose estimation using all image pixels. The first real-time method in this category was Dense Tracking and Mapping (DTAM) [22]. Since processing every pixel is computationally more expensive, DTAM achieved real-time performance using a GPU. Engel *et al.* [9] proposed Large-Scale Direct Monocular SLAM (LSD-SLAM) which runs in real-time on a CPU. The method estimates depth at pixels near image boundaries and recovers a semi-dense map. Apart from higher computational complexity, direct matching tends to work better for short baselines even with motion blur, while the invariance property of feature-based approaches allows large viewpoint changes. Engel *et al.* [10] proposed Semi-dense Visual Odometry (SVO), a hybrid between feature-based and direct SLAM methods, using a combination of direct methods to establish feature correspondences and feature-based methods to refine the camera pose estimates.

Depth prediction from single images has been a long-standing research problem, and deep learning methods have been shown to exceed methods using hand-crafted features in terms of the accuracy [7, 8, 15, 17, 18, 27]. Recent methods combine CNN-based depth predictions with visual SLAM. Laina *et al.* [15] proposed a fully convolutional architecture and residual learning to predict depth maps from images. In their evaluation, the predicted depth maps were input to Kellers Point-Based Fusion RGB-D SLAM algorithm [12]. The estimated 3D geometry lacks some shape detail because of blurred regions in the predicted depth maps. Recently, Tateno *et al.* [26] proposed CNN-SLAM in which predicted depth and normal maps are fused with direct monocular SLAM inspired by LSD-SLAM.

Depth fusion is an important process for reconstructing accurate and complete 3D shape from depth maps. Curless *et al.* proposed to use averaging truncated signed distance functions (TSDF) for depth susion [3] which is simple

Method	3D Reconstruction	Computational complexity	Accuracy	Scale
Monocular visual SLAM (feature based)	Sparse (scene complexity dependent)	Low (runs on mobile device)	High	None
CNN-based depth prediction	Dense (estimated for each pixel)	High (a few seconds for each frame)	Medium (training-data dependent)	Available
Proposed framework	Dense (estimated for each pixel)	High (but only visual SLAM runs on mobile device)	High	Available

Table 1. *Properties of individual reconstruction methods and of their combination, which retains desirable properties of each.*

yet effective and used in a large number of reconstruction pipelines including KinectFusion [21].

Mesh deformation techniques are widely used in graphics and vision. Especially, linear variational mesh deformation techniques were developed for editing detailed high-resolution meshes, like those produced by scanning real-world objects [2]. For local detail preservation mesh deformations that are locally as-rigid-as-possible (ARAP) have been proposed. The ARAP method by Sorkine *et al.* [25] optimizes rigid transformations in 1-ring neighborhoods ("cells"), maintaining consistency between adjacent pairs of rigid transformations by single overlapping edges. Levi *et al.* [16] introduced SR-ARAP energy formulation in which rotation of local neighborhood on mesh are constrained to be similar to neighbors, and enhance the smoothness of the ARAP method. In this work we convert depth maps to surface meshes and employ ARAP deformations using geometric constraints.

3. Depth fusion by geometric constraints

We designed our framework consisting of three parts, monocular visual SLAM, CNN-based depth prediction, and surface mesh deformation for fusing depth maps constrained by geometric constraints generated by the SLAM process. Figure 2 shows the pipeline of our framework. For the implementation we use a client-server design, where feature-based monocular SLAM runs on the mobile device and distinctive key-frames together with camera poses and 2D and 3D feature coordinates are sent to the server. On the server side, a depth map is predicted for each key frame and converted to a surface mesh. Finally, the surface meshes are deformed using 3D features as geometric constraints, and fused to a 3D reconstruction. Updates of the 3D reconstruction are returned to the client and where it is visualized for the current camera position. In the following subsections, we will detail each stage of the framework, SLAM process in Section3.1, depth prediction in Section 3.2, and surface mesh deformation in 3.3.

3.1. Monocular visual SLAM

Although our framework is compatible with any type of feature-based monocular visual SLAM methods, we employ ORB-SLAM [20] because of its robustness and accuracy. ORB-SLAM incorporates three parallel threads: tracking, mapping and loop closing. The tracking is in charge of localizing the camera in every frame and deciding when to insert a new key-frame. The mapping processes new key-frames and performs local bundle adjustment for reconstruction. The loop closing searches for loops with every new key-frame.

Each key-frame K_t is associated with camera pose T_{k_t} at time t, locations of ORB features $\mathbf{p^{2D}(t)}$ and corresponding 3D map points $\mathbf{p^{3D}(t)}$. Note that T_{k_t} and $\mathbf{p^{3D}(t)}$ are defined in the map coordinates, which lacks absolute scale.

3.2. CNN-based depth prediction

For depth prediction, we use the state-of-the-art architecture proposed in [7]. When a new key-frame is created and sent to the server-side, a depth map is predicted by the CNN. The CNN of [7] is a three-step multi-scale network that predicts the structure of the scene taking context into account by including pooling and convolution layers with different stride and kernel sizes. The network is trained using an element-wise L2 loss that explicitly accounts for depth relations between pixel locations, in addition to the point-wise error. The loss is defined as:

$$\begin{aligned} L_{depth}(D, D^*) = \frac{1}{n}\sum_i d_i^2 - \frac{1}{2n^2}\left(\sum_i d_i\right)^2 \\ +\frac{1}{n}\sum_i[(\bigtriangledown_x d_i^2) + (\bigtriangledown_y d_i^2)], \end{aligned} \quad (1)$$

where D and D^* are predicted depth and ground truth depth, respectively, the loss equals $D - D^*$.

After computing the depth map D_t of key-frame K_t, we convert it to a point cloud in which points correspond to pixels in the map, and a surface mesh S_t^{cam} defined in camera coordinates at time t. Surface mesh S_t^{cam} is fused with other meshes to form a unique 3D reconstruction in the next step. In this deformation process, the mesh needs to be wa-

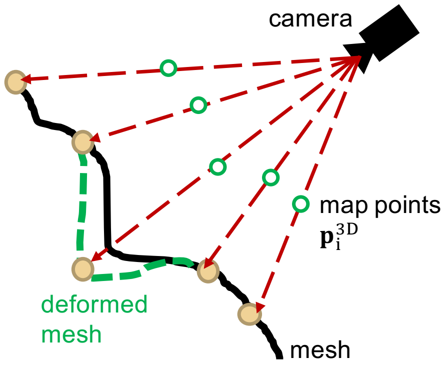

Figure 3. *Correspondences between 3D map points and mesh vertices.*

tertight to avoid mesh corruption because of the local disparities of deformation force. We simply define edges between vertices based on the pixel connectivities on the map.

3.3. Mesh deformation for depth fusion

We fuse surface mesh S_t^{cam} into a unique 3D reconstruction in world coordinates based on map points $\mathbf{p^{3D}(t)}$ defined in map coordinates. We first convert S_t^{cam} to S_t^{map} defined in map coordinates by reprojecting each vertex using the associated camera pose T_{k_t} recorded in map coordinates. Secondly, we scale S_t^{cam} by minimizing the distance between map points $\mathbf{p_i^{3D}(t)}$ and corresponding vertices $\mathbf{v_i}$ in S_t^{cam} as follows:

$$s_t^{map} = \operatorname*{argmin}_s \sum_i \parallel s\mathbf{v_i} - \mathbf{p_i^{3D}(t)} \parallel^2, \tag{2}$$

where s_t^{map} is the scale factor for S_t^{cam}. The correspondences $f : \mathbf{v_i} \mapsto \mathbf{p_i^{3D}(t)}$ can be easily found by projecting a ray from the camera center to map points and find the vertex nearest to its intersection with the mesh (see Figure 3).

Our mesh deformation is inspired by as-rigid-as-possible (ARAP) transformations proposed in [25]. We use the set of map points $\mathbf{p_i^{3D}(t)}$ as the geometric constraint of the deformation and define the one-ring neighborhood of each vertex. Ideally the deformation seeks to keep the transformation for the surface in each local neighborhood as rigid as possible. Overlap of local neighborhoods is necessary to avoid surface stretching or shearing at the boundary of the local neighborhoods. By using the local neighborhood concept, we can define the following energy function for the local neighborhood C_i, corresponding to vertex $\mathbf{v_i}$, and its deformed version C_i':

$$E(C_i') = \sum_{\mathbf{v_j} \in \mathbf{C_i}} w_{ij} \parallel (\mathbf{v_i'} - \mathbf{v_j'}) - \mathbf{R_i}(\mathbf{v_i} - \mathbf{v_j}) \parallel^2, \tag{3}$$

where $\mathbf{R_i}$ is a 3×3 rotation matrix and w_{ij} denotes the cotangent weight:

$$w_{ij} = \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}), \tag{4}$$

where α_{ij}, β_{ij} are the angles opposite of the mesh edge (i, j). We define the energy function of the whole mesh by summing over the deviations from rigidity per local neighborhood as follows:

$$E(S_t') = \sum_{\mathbf{v_i} \in \mathbf{S_t}} w_i E(C_i'), \tag{5}$$

where w_i is a weight for the local neighborhood C_i.

We expand the ARAP method by defining w_i based on the normal vector corresponding to the local neighborhood of each vertex. As we define S_t as a watertight surface mesh, and its corresponding depth map is generated from a single camera viewpoint, there are areas in which their normals are nearly perpendicular to the ray from the camera center, *i.e.*, the observability from the camera is low. These areas tend to be boundary areas between objects in the scene, and do not correspond to objects in the actual scene. To maintain shape details of the objects in the scene, we selectively deform these areas as much as possible by defining the weight w_i by a sigmoid function as follows:

$$w_i = \frac{1}{1 + e^{-a(x + b\pi)}}, \tag{6}$$

where x denotes the angle between the normal of the local neighborhood and the ray from the camera center to the vertex, a and b are empirically defined parameters. Figure 4 shows the distribution of w_i on a surface mesh.

We further introduce a bending factor B_{ij}, as suggested in [16], as follows:

$$B_{ij} = \alpha A \parallel \mathbf{R_i} - \mathbf{R_j} \parallel, \tag{7}$$

where α is a weighting coefficient, A is the surface area for scaling invariance, and finally update Equation 5 as follows:

$$E(S_t') = \sum_{\mathbf{v_i} \in \mathbf{S_t}} w_i E(C_i') + B_{ij}. \tag{8}$$

After this deformation, we scale the deformed mesh by the absolute scale s_t^{world} estimated for time t as follows:

$$s_t^{world} = \frac{t}{\sum_t s_t^{cam}}. \tag{9}$$

The scaled 3D mesh is sent to the client, and rendered from the current camera position or any other viewpoints specified by the user.

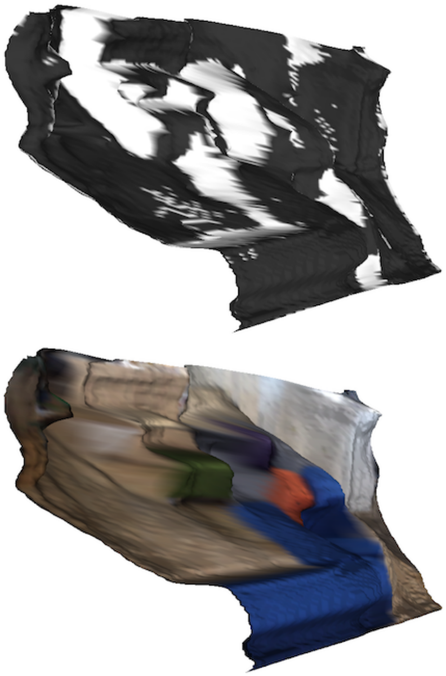

Figure 4. *(Top) Distribution of weights w_i for the deformation and (bottom) the corresponding textured mesh. Larger intensity values in the top figure indicate the higher weights.*

4. Experiments

We evaluate our framework quantitatively by comparing the 3D reconstruction result with ground truth data. For the acquisition of the ground truth data and the input images, we used a tablet equipped with an RGB-D *StructureSensor* [23]. The sensor captures RGB images at VGA resolution and depth images at QVGA resolution, and is able to capture within a range of 40 to 350 cm distance to the camera (see Figure 5).

Our CNN architecture is trained on the *NYU Depth Dataset v2* [24], and thus performs well on typical indoor scenes. Our framework is designed for the case in which the CNN-predicted depth map is inaccurate. We captured a new challenging office dataset with many poorly textured surfaces.

We fed the image sequences into the proposed framework. ORB-SLAM runs in real-time (5 to 10 fps) on the client, specifically, an iPhone 6 with an A8 processor and a 8 mega-pixel camera. The other components, CNN depth estimation and mesh adaptation is carried out on the server, a PC with Intel Xeon dual core CPU, 2.4GHz, 96GB of RAM and an Nvidia GeForce GTX Titan X GPU with 12GB of VRAM. To adjust to the original implementation of the CNN and increase speed, both input RGB images and estimated depth images are resized to 320 × 240. The average processing time of the depth prediction for each key-frame is 2.6 seconds. This is longer than the duration between key-frames detected by ORB-SLAM because these are selected based on visual changes. We filter out those key-frames using a spatio-temporal distance criterion similar to the other feature-based approaches, *e.g.*, PTAM, and send them to the server.

The key-frames are processed on the server and the depth image for each frame is estimated by the CNN architecture. In the fusion process, we convert the depth images to a refined mesh sequence as shown at the bottom of Figure 5.We also make the ground truth mesh sequence correspond to the refined one from the raw depth maps captured by the depth sensor on the other hand. We compute residual errors between the refined mesh and the ground truth as shown in Table 2 and Figure 6. We can observe that our framework efficiently reduces the residual errors for all sequences. Both the average and the median of the residual errors fall within the range from about two thirds to a half.

We also evaluate the absolute scale estimated from depth prediction as shown in the rightmost column in the Table 2. The average error of the estimated scales for our six office scenes is 20% of the ground truth scale.

5. Conclusion

In this paper, we proposed a framework fusing the result of geometric measurement, *i.e.*, feature-based monocular visual SLAM and CNN-based depth prediction. We have shown its efficiency andpotential for applications which can run on standard mobile devices only equipped with a single camera. Thanks to the capability of CNN for depth prediction, some of the main limitations of feature-based monocular visual SLAM, such as lack the absolute scale, sparse 3D reconstruction, were overcome. The 3D map acquired by monocular visual SLAM also compensate the limitation of CNN-based depth maps by refining it with surface mesh deformation to maintain shape details. There are several possible directions of future work. The first is global mesh refinement and integration based on the photometric and geometric consistency between meshes to obtain a unified reconstruction result similar to [28]. As the current deformation is constrained only by sparse 3D features, refined meshes are not fully registered with each other as show at the bottom of Figure 5. Second direction is IMU-based scale estimation. As the absolute scale in the current framework is predicted by CNN and highly depend on the training data, we expect the its accuracy can be enhanced if we fuse it with IMU measurement. Another direction is full use of CNN-based prediction, *e.g.*, semantic labeling. By utilizing semantic labeling, we can selectively manipulate the 3D reconstruction result. For instance, we can recognize real furniture in the reconstruction and replace it with a virtual one.

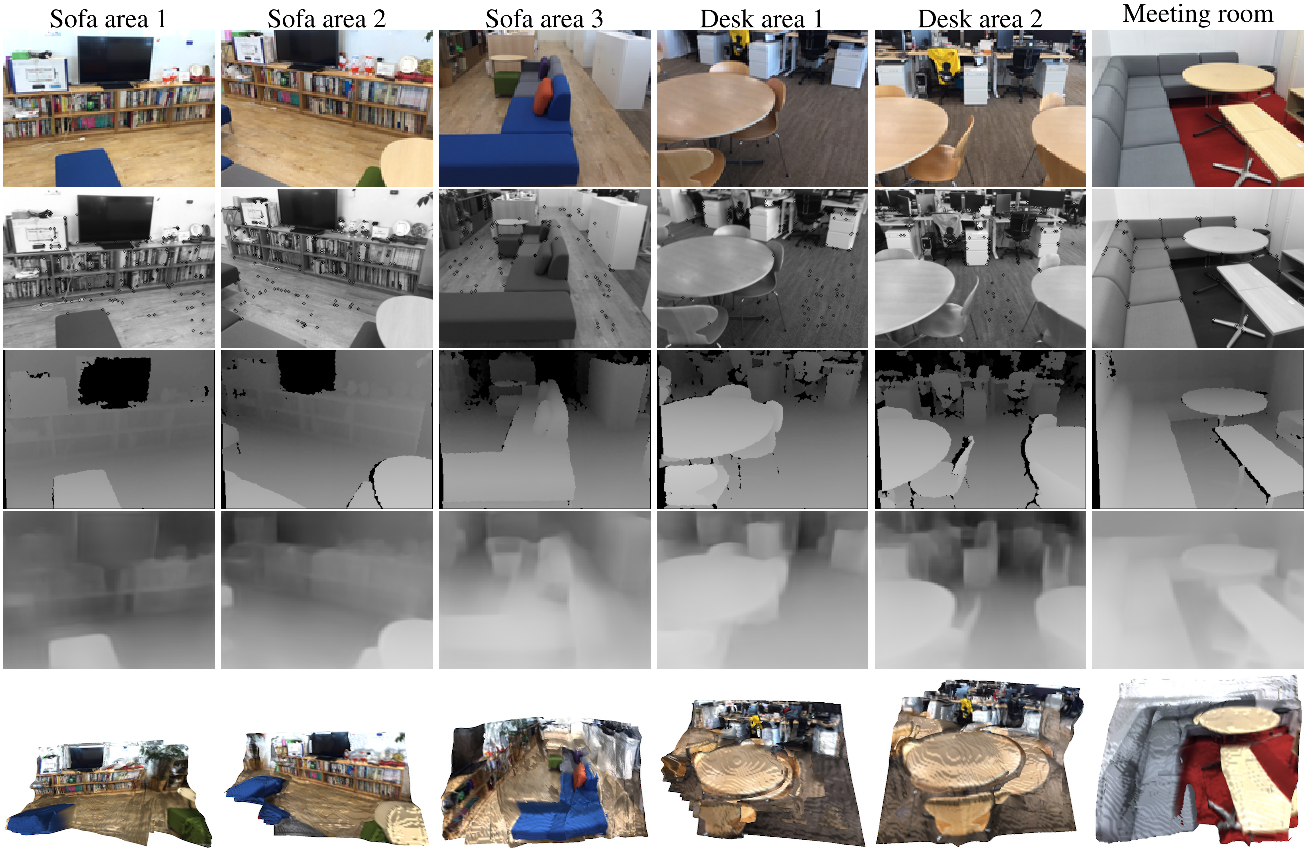

Figure 5. **Input data for our depth fusion and the reconstructed scenes.** *From top to bottom row: color images, feature tracking result of SLAM, corresponding ground truth depth images, depth images estimated by DNN, and 3D reconstruction results on six office scenes, respectively.*

Scene	**Mesh from CNN depth map**			**Refined mesh by our method**			
	Mean	Median	Std dev	Mean	Median	Std dev	Scale
Sofa area 1	112.2	110.8	39.5	81.4	77.0	36.7	0.73
Sofa area 2	87.2	80.0	40.3	53.0	39.0	40.7	0.83
Sofa area 3	26.6	20.6	25.9	16.8	10.2	23.2	1.18
Desk area 1	48.0	34.8	40.0	35.1	22.3	37.4	0.96
Desk area 2	41.1	27.8	38.5	37.1	20.7	52.5	0.98
Meeting room	31.4	21.6	31.4	21.3	11.7	27.7	1.54

Table 2. **Accuracy improvement results.** *Comparison of residual errors [cm] from the ground truth obtained using a depth sensor, and the absolute scale estimated based on depth prediction.*

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, Oct. 2011. 1

[2] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, Jan 2008. 3

[3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM. 2

[4] A. J. Davison and D. W. Murray. *Mobile robot localisation using active vision*, pages 809–825. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. 1

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. 29(6):1052–1067, 2007. 1

[6] T. Dobbert. *Matchmoving: The Invisible Art of Camera Tracking*. John Wiley and Sons, 2012. 1

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolu-

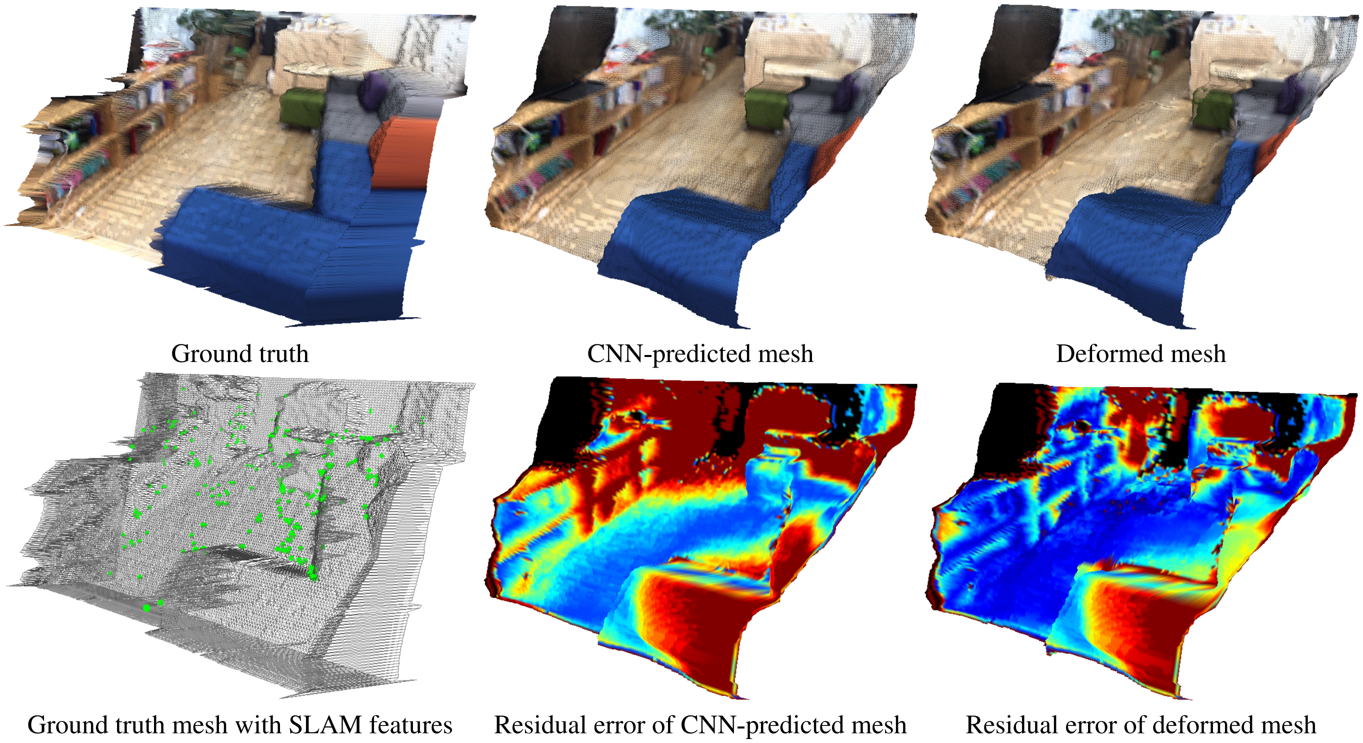
Ground truth
CNN-predicted mesh
Deformed mesh
Ground truth mesh with SLAM features
Residual error of CNN-predicted mesh
Residual error of deformed mesh

Figure 6. **Surface mesh deformation result.** *(Top left) ground truth surface mesh obtained from depth-sensor measurements, (top center) surface mesh converted from the CNN-predicted depth map, (top right) deformation result of center mesh using 3D SLAM features. (Bottom left) 3D SLAM features together with the ground truth mesh, (bottom center and right) residual errors of CNN-prediction and deformed mesh. Warm colors indicates higher residual error. Black regions indicate areas beyond the maximum depth sensor range.*

tional architecture. *CoRR*, abs/1411.4734, 2014. 1, 2, 3

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. 1, 2

[9] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, September 2014. 1, 2

[10] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *ICCV*, Sydney, Australia, December 2013. 2

[11] M. Hofer, M. Maurer, and H. Bischof. Efficient 3D scene abstraction using line segments. *CVIU*, 2016. 1

[12] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3D reconstruction in dynamic scenes using point-based fusion. In *3DV*, pages 1–8, Washington, DC, USA, 2013. IEEE Computer Society. 2

[13] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society. 2

[14] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. Turning mobile phones into 3d scanner. In *CVPR*, 2014. 1

[15] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016. 1, 2

[16] Z. Levi and C. Gotsman. Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE Transactions on Visualization and Computer Graphics*, 21:264–277, 2015. 3, 4

[17] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, June 2015. 2

[18] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. *CoRR*, abs/1411.6387, 2014. 2

[19] S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2013. 1

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015. 1, 2, 3

[21] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011. 3

[22] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, pages 2320–2327, Nov 2011. 1, 2

[23] Occipital. https://structure.io. v0.6.2, 2016. 5

[24] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5

[25] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proc. Fifth Eurographics Symp. Geometry Processing*, SGP '07, pages 109–116, 2007. 3, 4

[26] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. *CoRR*, abs/1704.03489, 2017. 1, 2

[27] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, June 2015. 2

[28] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *Int. J. Robotics Research*, 35(14):1697–1716, 2016. 5

[29] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 1

[30] G. Younes, D. C. Asmar, and E. A. Shammas. A survey on non-filter-based monocular visual SLAM systems. *CoRR*, abs/1607.00470, 2016. 2