CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction

Keisuke Tateno*[1,2], Federico Tombari*[1], Iro Laina[1], Nassir Navab[1,3]
{tateno, tombari, laina, navab}@in.tum.de
[1] CAMP - TU Munich, Munich, Germany
[2] Canon Inc., Tokyo, Japan
[3] Johns Hopkins University, Baltimore, US

Abstract

Given the recent advances in depth prediction from Convolutional Neural Networks (CNNs), this paper investigates how predicted depth maps from a deep neural network can be deployed for accurate and dense monocular reconstruction. We propose a method where CNN-predicted dense depth maps are naturally fused together with depth measurements obtained from direct monocular SLAM. Our fusion scheme privileges depth prediction in image locations where monocular SLAM approaches tend to fail, e.g. along low-textured regions, and vice-versa. We demonstrate the use of depth prediction for estimating the absolute scale of the reconstruction, hence overcoming one of the major limitations of monocular SLAM. Finally, we propose a framework to efficiently fuse semantic labels, obtained from a single frame, with dense SLAM, yielding semantically coherent scene reconstruction from a single view. Evaluation results on two benchmark datasets show the robustness and accuracy of our approach.

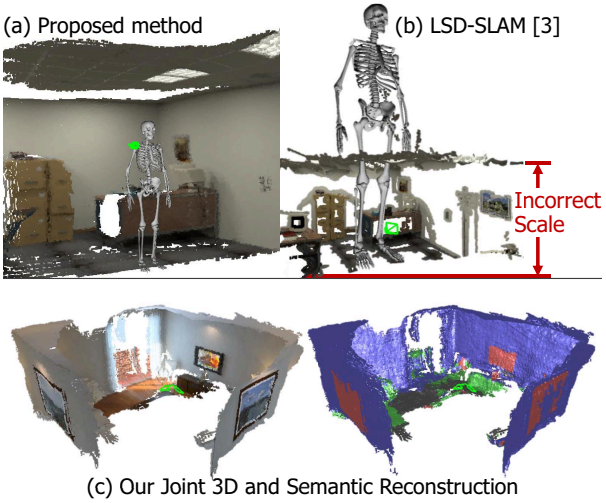

Figure 1. The proposed monocular SLAM approach (a) can estimate a much better absolute scale than the state of the art (b), which is necessary for many SLAM applications such as AR, *e.g.* the skeleton is augmented into the reconstruction. c) our approach can yield joint 3D and semantic reconstruction from a single view.

1. Introduction

Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM) are umbrella names for a highly active research area in the field of computer vision and robotics for the goal of 3D scene reconstruction and camera pose estimation from 3D and imaging sensors. Recently, real-time SLAM methods aimed at fusing together range maps obtained from a moving depth sensor have witnessed an increased popularity, since they can be employed for navigation and mapping of several types of autonomous agents, from mobile robots to drones, as well as for many augmented reality and computer graphics applications. This is the case of volumetric fusion approaches such as Kinect Fusion [21], as well as dense SLAM methods based on RGB-D data [30, 11], which, in addition to navigation and mapping, can also be employed for accurate scene reconstruction. However, a main drawback of such approaches is that depth cameras have several limitations: indeed, most of them have a limited working range, and those based on active sensing cannot work (or perform poorly) under sunlight, thus making reconstruction and mapping less precise if not impossible in outdoor environments.

In general, since depth cameras are not as ubiquitous as color cameras, a lot of research interest has been focused on dense and semi-dense SLAM methods from a single camera [22, 4, 20]. These approaches aim at real-time monocular scene reconstruction by estimating the depth map of the current viewpoint through small-baseline stereo matching over pairs of nearby frames. The working assumption is that the camera translates in space over time, so that pairs of consecutive frames can be treated as composing a stereo rig. Stereo matching is usually carried out through color consistency or by relying on keypoint extraction and matching.

One main limitation of monocular SLAM approaches is the estimation of the absolute scale. Indeed, even if camera

*The first two authors contribute equally to this paper.

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.695

pose estimation and scene reconstruction are carried out accurately, the absolute scale of such reconstruction remains inherently ambiguous, limiting the use of monocular SLAM within most aforementioned applications in the field of augmented reality and robotics (an example is shown in Fig. 1,b). Some approaches suggest solving the issue via object detection by matching the scene with a pre-defined set of 3D models, so to recover the initial scale based on the estimated object size [6], which nevertheless fails in absence of known shapes in the scene. Another main limitation of monocular SLAM is represented by pose estimation under pure rotational camera motion, in which case stereo estimation cannot be applied due to the lack of a stereo baseline, resulting in tracking failures.

Recently, a new avenue of research has emerged that addresses depth prediction from a single image by means of learned approaches. In particular, the use of deep Convolutional Neural Networks (CNNs) [16, 2, 3] in an end-to-end fashion has demonstrated the potential of regressing depth maps at a relatively high resolution and with a good absolute accuracy even under the absence of monocular cues (texture, repetitive patterns) to drive the depth estimation task. One advantage of deep learning approaches is that the absolute scale can be learned from examples and thus predicted from a single image without the need of scene-based assumptions or geometric constraints, unlike [10, 18, 1]. A major limitation of such depth maps is the fact that, although globally accurate, depth borders tend to be locally blurred: hence, if such depths are fused together for scene reconstruction as in [16], the reconstructed scene will overall lack shape details.

Relevantly, despite the few methods proposed for single view depth prediction, the application of depth prediction to higher-level computer vision tasks has been mostly overlooked so far, with just a few examples existing in literature [16]. The main idea behind this work is to exploit the best from both worlds and propose a monocular SLAM approach that fuses together depth prediction via deep networks and direct monocular depth estimation so to yield a dense scene reconstruction that is at the same time unambiguous in terms of absolute scale and robust in terms of tracking. To recover blurred depth borders, the CNN-predicted depth map is used as initial guess for dense reconstruction and successively refined by means of a direct SLAM scheme relying on small-baseline stereo matching similar to the one in [4]. Importantly, small-baseline stereo matching holds the potential to refine edge regions on the predicted depth image, which is where they tend to be more blurred. At the same time, the initial guess obtained from the CNN-predicted depth map can provide absolute scale information to drive pose estimation, so that the estimated pose trajectory and scene reconstruction can be significantly more accurate in terms of absolute scale compared to the state of the art in monocular SLAM. Fig. 1, a) shows an example illustrating the usefulness of carrying out scene reconstruction with a precise absolute scale such as the one proposed in this work. Moreover, tracking can be made more robust, as the CNN-predicted depth does not suffer from the aforementioned problem of pure rotations, as it is estimated on each frame individually. Last but not least, this framework can run in real-time since the two processes of depth prediction from CNNs and depth refinement can be simultaneously carried out on different computational resources of the same architecture - respectively, the GPU and the CPU.

Another relevant aspect of recent CNNs is that the same network architecture can be successfully employed for different high-dimensional regression tasks rather than just depth estimation: one typical example is semantic segmentation [3, 29]. We leverage this aspect to propose an extension of our framework that uses pixel-wise labels to coherently and efficiently fuse semantic labels with dense SLAM, so to attain semantically coherent scene reconstruction from a single view: an example is shown in Fig. 1, c). Notably, to the best of our knowledge semantic reconstruction has been shown only recently and only based on stereo [28] or RGB-D data [15], *i.e.* never in the monocular case.

We validate our method with a comparison on two public SLAM benchmarks against the state of the art in monocular SLAM and depth estimation, focusing on the accuracy of pose estimation and reconstruction. Since the CNN-predicted depth relies on a training procedure, we show experiments where the training set is taken from a completely different environment and a different RGB sensor than those available in the evaluated benchmarks, so to portray the capacity of our approach - particularly relevant for practical uses - to generalize to novel, unseen environments. We also show qualitative results of our joint scene reconstruction and semantic label fusion in a real environment.

2. Related work

In this Section we review related work with respect to the two fields that we integrate within our framework, *i.e.* SLAM and depth prediction.

SLAM There exists a vast literature on SLAM. From the point of view of the type of input data being processed, approaches can be classified into either depth camera-based [21, 30, 11] and monocular camera-based [22, 4, 20]. Instead, from a methodological viewpoint, they are classified as either feature-based [12, 13, 20] and direct [22, 5, 4]. Given the scope of this paper, we will focus here only on monocular SLAM approaches.

As for feature-based monocular SLAM, ORB-SLAM [20] is arguably the state of the art in terms of pose estimation accuracy. This method relies on the extraction of

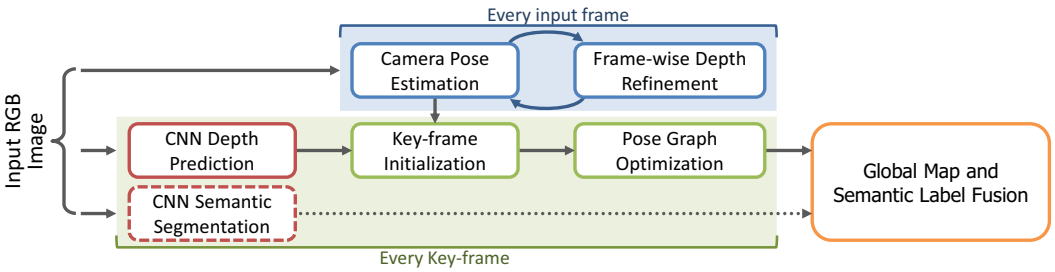

Figure 2. CNN-SLAM Overview.

sparse ORB features from the input image to carry out a sparse reconstruction of the scene as well as to estimate the camera pose, also employing local bundle adjustment and pose graph optimization. As for direct monocular SLAM, the Dense Tracking and Mapping (DTAM) of [22] achieved dense reconstruction in real-time on s GPU by using short-baseline multiple-view stereo matching with a regularization scheme, so that depth estimation is smoother on low-textured regions in the color image. Moreover, the Large-Scale Direct SLAM (LSD-SLAM) algorithm [4] proposed the use of a semi-dense map representation which keeps track of depth values only on gradient areas of the input image, this allowing enough efficiency to enable direct SLAM in real-time on a CPU. An extension of LSD-SLAM is the recent Multi-level mapping (MLM) algorithm [7], which proposed the use of a dense approach on top of LSD-SLAM in order to increase its density and improve the reconstruction accuracy.

Depth prediction from single view Depth prediction from single view has gained increasing attention in the computer vision community thanks to the recent advances in deep learning. Classic depth prediction approaches employ hand-crafted features and probabilistic graphical models [10, 18] to yield regularized depth maps, usually making strong assumptions on the scene geometry. Recently developed deep convolutional architectures significantly outperformed previous methods in terms of depth estimation accuracy [16, 2, 3, 29, 19, 17]. Interestingly, the work of [16] reports qualitative results of employing depth predictions for dense SLAM as an application. In particular, the predicted depth map is used as input for Keller's Point-Based Fusion RGB-D SLAM algorithm [11], showing that SLAM-based scene reconstruction can be obtained using depth prediction, although it lacks shape details, mostly due to the aforementioned blurring artifacts that are associated with the loss of fine spatial information through the contractive part of a CNN.

3. Proposed Monocular Semantic SLAM

In this section, we illustrate the proposed framework for 3D reconstruction, where CNN-predicted dense depth maps are fused together with depth measurements obtained from direct monocular SLAM. Additionally, we show how CNN-predicted semantic segmentation can also be coherently fused with the global reconstruction model. The flow diagram in Fig. 2 sketches the pipeline of our framework. We employ a key-frame based SLAM paradigm [12, 4, 20], in particular we use as baseline the direct semi-dense approach in [4]. Within such approach, a subset of visually distinct frames is collected as key-frames, whose pose is subject to global refinement based on pose graph optimization. At the same time, camera pose estimation is carried out at each input frame, by estimating the transformation between the frame and its nearest key-frame.

To maintain a high frame-rate, we propose to predict a depth map via CNN only on key-frames. In particular, if the currently estimated pose is far from that of existing key-frames, a new key-frame is created out of the current frame and its depth estimated via CNN. Moreover an uncertainty map is constructed by measuring the pixel-wise confidence of each depth prediction. Since in most cases the camera used for SLAM differs from the one used to acquire the dataset on which the CNN is trained, we propose a specific normalization procedure of the depth map designed to gain robustness towards different intrinsic camera parameters. When additionally carrying out semantic label fusion, we employ a second convolutional network to predict a semantic segmentation of the input frame. Finally, a pose graph on key-frames is created so to globally optimize their relative pose.

A particularly important stage of the framework, also representing one main contribution of our proposal, is the scheme employed to refine the CNN-predicted depth map associated to each key-frame via small-baseline stereo matching, by enforcing color consistency minimization between a key-frame and associated input frames. In particular, depth values will be mostly refined around image regions with gradients, i.e. where epipolar matching can provide improved accuracy. This will be outlined in Subsections 3.3 and 3.4. Relevantly, the way refined depths are propagated is driven by the uncertainty associated to each depth value, estimated according to a specifically proposed confidence measure (defined in Subsec. 3.3). Every stage of the framework is now detailed in the following Subsections.

3.1. Camera Pose Estimation

The camera pose estimation is inspired by the key-frame approach in [4]. In particular, the system holds a set of key-frames $k_1, .., k_n \in \mathcal{K}$ as structural elements on which to perform SLAM reconstruction. Each key-frame k_i is associated to a key-frame pose $\boldsymbol{T}_{k_i}$, a depth map $\mathcal{D}_{k_i}$, and a depth uncertainty map $\mathcal{U}_{k_i}$. In contrast to [4], our depth map is dense because it is generated via CNN-based depth prediction (see Subsec. 3.2). The uncertainty map measures the confidence of each depth value. As opposed to [4] that initializes the uncertainty to a large, constant value, our approach initializes it according to the measured confidence of the depth prediction (described in Subsec. 3.3). In the following, we will refer to a generic depth map element as $\boldsymbol{u} = (x, y)$, which ranges in the image domain, i.e. $\boldsymbol{u} \in \Omega \subset \mathbb{R}^2$, with $\dot{\boldsymbol{u}}$ being its homogeneous representation.

At each frame t, we aim to estimate the current camera pose $\boldsymbol{T}_t^{k_i} = [\boldsymbol{R}_t, \boldsymbol{t}_t] \in \mathbb{SE}(3)$, i.e. the transformation between the nearest key-frame k_i and frame t, composed of a 3×3 rotation matrix $\boldsymbol{R}_t \in \mathbb{SO}(3)$ and a 3D translation vector $\boldsymbol{t}_t \in \mathbb{R}^3$. This transformation is estimated by minimizing the photometric residual between the intensity image $\mathcal{I}_t$ of the current frame and the intensity image $\mathcal{I}_{k_i}$ of the nearest key-frame k_i, via weighted Gauss-Newton optimization based on the objective function

$$E(\boldsymbol{T}_t^{k_i}) = \sum_{\tilde{\boldsymbol{u}} \in \Omega} \rho \left(\frac{r\left(\tilde{\boldsymbol{u}}, \boldsymbol{T}_t^{k_i}\right)}{\sigma\left(r\left(\tilde{\boldsymbol{u}}, \boldsymbol{T}_t^{k_i}\right)\right)} \right) \tag{1}$$

where ρ is the Huber norm and σ is a function measuring the residual uncertainty [4]. Here, r is the photometric residual defined as

$$r\left(\tilde{\boldsymbol{u}}, \boldsymbol{T}_t^{k_i}\right) = \mathcal{I}_{k_i}(\tilde{\boldsymbol{u}}) - \mathcal{I}_t\left(\pi\left(\boldsymbol{K}\boldsymbol{T}_t^{k_i}\tilde{\mathcal{V}}_{k_i}(\tilde{\boldsymbol{u}})\right)\right) . \tag{2}$$

Considering that our depth map is dense, for the sake of efficiency, we limit the computation of the photometric residual only on the subset of pixels lying within high color gradient regions, defined by the image domain subset $\tilde{\boldsymbol{u}} \subset \boldsymbol{u} \in \Omega$. Also, in (2), π represents the perspective projection function mapping a 3D point to a 2D image coordinate

$$\pi\left([xyz]^T\right) = (x/z, y/z)^T \tag{3}$$

while $\mathcal{V}_{k_i}(\boldsymbol{u})$ represents a 3D element of the vertex map computed from the key-frame's depth map

$$\mathcal{V}_{k_i}(\boldsymbol{u}) = \boldsymbol{K}^{-1}\dot{\boldsymbol{u}}\mathcal{D}_{k_i}(\boldsymbol{u}) \tag{4}$$

where $\boldsymbol{K}$ is the camera intrinsic matrix.

Once $\boldsymbol{T}_t^{k_i}$ is obtained, the current camera pose in world coordinate system is computed as $\boldsymbol{T}_t = \boldsymbol{T}_t^{k_i}\boldsymbol{T}_{k_i}$.

3.2. CNN-based Depth Prediction and Semantic Segmentation

Every time a new key-frame is created, an associated depth map is predicted via CNN. The depth prediction architecture that we employ is the state-of-the-art approach proposed in [16], based on the extension of the Residual Network (ResNet) architecture [9] to a Fully Convolutional network. In particular, the first part of the architecture is based on ResNet-50 [9] and initialized with pre-trained weights on ImageNet [24]. The second part of the architecture replaces the last pooling and fully connected layers originally presented in ResNet-50 with a sequence of residual up-sampling blocks composed of a combination of unpooling and convolutional layers. After up-sampling, drop-out is applied before a final convolutional layer which outputs a 1-channel output map representing the predicted depth map. The loss function is based on the reverse Huber function [16].

Following the successful paradigm of other approaches that employed the same architecture for both depth prediction and semantic segmentation tasks [3, 29], we also retrained this network for predicting pixel-wise semantic labels from RGB images. To deal with this task, we modified the network so that it has as many output channels as the number of categories and employed a soft-max layer and a cross-entropy loss function to be minimized via back-propagation and Stochastic Gradient Descent (SGD). It is important to point out that, although in principle any semantic segmentation algorithm could be used, the primary objective of this work is to showcase how frame-wise segmentation maps can be successfully fused within our monocular SLAM framework (see Subsec. 3.5).

3.3. Key-frame Creation and Pose Graph Optimization

One limitation of using a pre-trained CNN for depth prediction is that, if the sensor used for SLAM has different intrinsic parameters from those used to capture the training set, the resulting absolute scale of the 3D reconstruction will be inaccurate. To ameliorate this issue, we propose to adjust the depth regressed via CNN with the ratio between the focal length of the current camera, f_{cur} and that of the sensor used for training, f_{tr} as

$$\mathcal{D}_{k_i}(\boldsymbol{u}) = \frac{f_{cur}}{f_{tr}}\tilde{\mathcal{D}_{k_i}}(\boldsymbol{u}) \tag{5}$$

where $\tilde{\mathcal{D}_{k_i}}$ is the depth map directly regressed by the CNN from the current key-frame image $\mathcal{I}_i$.

Fig. 3 shows the usefulness of the adjustment procedure defined in (5), on a sequence of the benchmark *ICL-NUIM* dataset [8] (compare *(a)* with *(b)*). As shown, the performance after the adjustment procedure is significantly

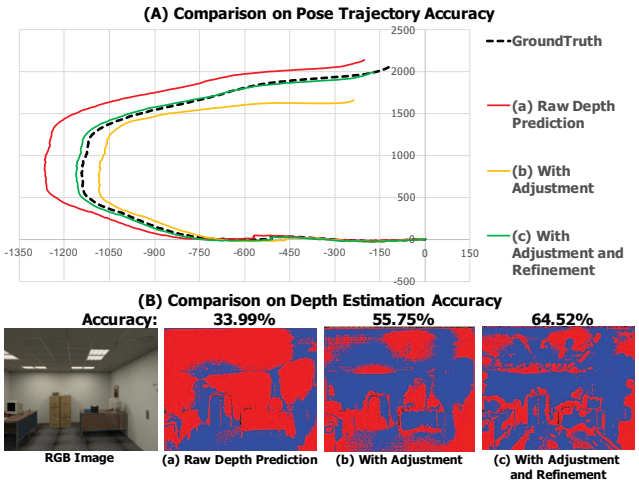

Figure 3. Comparison among (a) direct CNN-depth prediction, (b) after depth adjustment and (c) after depth adjustment and refinement, in terms of (A) pose trajectory accuracy and (B) depth estimation accuracy. Blue pixels depict correctly estimated depths, i.e. within 10 % of ground-truth. The comparison is done on one sequence of the *ICL-NUIM* dataset [8].

improved over that of using the depth map as directly predicted by the CNN. The improvement shows both in terms of depth accuracy as well as pose trajectory accuracy.

In addition, we associate each depth map $\mathcal{D}_{k_i}$ to an uncertainty map $\mathcal{U}_{k_i}$. In [4], this map is initialized by setting each element to a large, constant value. Since the CNN provides us with dense maps at each frame but without relying on any temporal regularization, we propose to instead initialize our uncertainty map by computing a confidence value based on the difference between the current depth map and its respective scene point on the nearest key-frame. Thus, this confidence measures how coherent each predicted depth value is across different frames: for those elements associated to a high confidence, the successive refinement process will be much faster and effective than the one in [4].

Specifically, the uncertainty map $\mathcal{U}_{k_i}$ is defined as the element-wise squared difference between the depth map of the current key-frame k_i and that of the nearest key-frame k_j, warped according to the estimated transformation $\boldsymbol{T}_{k_j}^{k_i}$ from k_i to k_j

$$\mathcal{U}_{k_i}(\boldsymbol{u}) = \left(\mathcal{D}_{k_i}(\boldsymbol{u}) - \mathcal{D}_{k_j}\left(\pi\left(\boldsymbol{K}\boldsymbol{T}_{k_j}^{k_i}\mathcal{V}_{k_i}(\boldsymbol{u})\right)\right)\right)^2. \quad (6)$$

To further improve the accuracy of each newly initialized key-frame, we propose to fuse its depth map and uncertainty map with those propagated from the nearest key-frame (this will obviously not apply to the very first key-frame) after they have been refined with new input frames (the depth refinement process is described in Subsection 3.4). To achieve this goal, we first define a propagated uncertainty map from the nearest key-frame k_j as

$$\tilde{\mathcal{U}}_{k_j}(\boldsymbol{v}) = \frac{D_{k_j}(\boldsymbol{v})}{D_{k_i}(\boldsymbol{u})}\mathcal{U}_{k_j}(\boldsymbol{v}) + \sigma_p^2 \quad (7)$$

where $\boldsymbol{v} = \pi\left(\boldsymbol{K}\boldsymbol{T}_{k_j}^{k_i}\mathcal{V}_{k_i}(\boldsymbol{u})\right)$ while, following [4], σ_p^2 is the white noise variance used to increase the propagated uncertainty. Then, the two depth maps and uncertainty maps are fused together according to the weighted scheme

$$\mathcal{D}_{k_i}(\boldsymbol{u}) = \frac{\tilde{\mathcal{U}}_{k_j}(\boldsymbol{v})\cdot\mathcal{D}_{k_i}(\boldsymbol{u}) + \mathcal{U}_{k_i}(\boldsymbol{u})\cdot\mathcal{D}_{k_j}(\boldsymbol{v})}{\mathcal{U}_{k_i}(\boldsymbol{u}) + \tilde{\mathcal{U}}_{k_j}(\boldsymbol{v})} \quad (8)$$

$$\mathcal{U}_{k_i}(\boldsymbol{u}) = \frac{\tilde{\mathcal{U}}_{k_j}(\boldsymbol{v})\cdot\mathcal{U}_{k_i}(\boldsymbol{u})}{\mathcal{U}_{k_i}(\boldsymbol{u}) + \tilde{\mathcal{U}}_{k_j}(\boldsymbol{v})}. \quad (9)$$

Finally, the pose graph is also updated at each new key-frame, by creating new edges with the key-frames already present in the graph that share a similar field of view (i.e., having a small relative pose) with the newly added key-frame. Moreover, the pose of the key-frames is each time globally refined via pose graph optimization [14].

3.4. Frame-wise Depth Refinement

The goal of this stage is to continuously refine the depth map of the currently active key-frame based on the depth maps estimated at each new frame. To achieve this goal, we use the small baseline stereo matching strategy described in the semi-dense scheme of [5], by computing at each pixel of the current frame t a depth map $\mathcal{D}_t$ and an uncertainty map $\mathcal{U}_t$ based on the 5-pixel matching along the epipolar line. These two maps are aligned with the key-frame k_i based on the estimated camera pose $\boldsymbol{T}_t^{k_i}$.

The estimated depth map and uncertainty map are then directly fused with those of the nearest key-frame k_i as follows:

$$\mathcal{D}_{k_i}(\boldsymbol{u}) = \frac{\mathcal{U}_t(\boldsymbol{u})\cdot\mathcal{D}_{k_i}(\boldsymbol{u}) + \mathcal{U}_{k_i}(\boldsymbol{u})\cdot\mathcal{D}_t(\boldsymbol{u})}{\mathcal{U}_{k_i}(\boldsymbol{u}) + \mathcal{U}_t(\boldsymbol{u})} \quad (10)$$

$$\mathcal{U}_{k_i}(\boldsymbol{u}) = \frac{\mathcal{U}_t(\boldsymbol{u})\cdot\mathcal{U}_{k_i}(\boldsymbol{u})}{\mathcal{U}_{k_i}(\boldsymbol{u}) + \mathcal{U}_t(\boldsymbol{u})} \quad (11)$$

Importantly, since the key-frame is associated to a dense depth map thanks to the proposed CNN-based prediction, this process can be carried out densely, i.e. every element of the key-frame is refined, in contrast to [5] that only refines depth values along high gradient regions. Since the observed depths within low-textured regions tend to have a high-uncertainty (i.e., a high value in $\mathcal{U}_t$), the proposed approach will naturally lead to a refined depth map where elements in proximity of high intensity gradients will be refined by the depth estimated at each frame, while elements within more and more low-textured regions will gradually hold the predicted depth value from the CNN, without being affected from uncertain depth observations.

Fig. 3 demonstrates the effectiveness of the proposed depth map refinement procedure on a sequence of the

benchmark *ICL-NUIM* dataset [8]. The Figure reports, in *(c)*, the performance obtained after both adjustment and depth refinement of the depth map, showing a significant improvement of both depth estimation and pose trajectory with respect to the previous cases.

3.5. Global Model and Semantic Label Fusion

The obtained set of key-frames can be fused together to generate a 3D global model of the reconstructed scene. Since the CNN is trained to provide semantic labels in addition to depth maps, semantic information can be also associated to each element of the 3D global model, through a process that we denote as semantic label fusion.

In our framework, we employ the real-time scheme proposed in [27], which aims at incrementally fusing together the depth map and the connected component map obtained from each frame of a RGB-D sequence. This approach uses a Global Segmentation Model (GSM) to average the assignment of labels to each 3D element over time, so to be robust to noise in the frame-wise segmentation. In our case, the pose estimation is provided as input to the algorithm, since camera poses are estimated via monocular SLAM, while input depth maps are those associated to the set of collected key-frames only. Here, instead of connected component maps as in [27], we use semantic segmentation maps. The result is a 3D reconstruction of the scene, incrementally built over new key-frames, where each 3D element is associated to a semantic class from the set used to train the CNN.

4. Evaluation

We provide here an experimental evaluation to validate the contributions of our method in terms of tracking and reconstruction accuracy, by means of a quantitative comparison against the state of the art on two public benchmark datasets (Subsec. 4.1), as well as a qualitative assessment in terms of robustness against pure rotational camera motions (Subsec. 4.2) and accuracy of semantic label fusion (Subsec. 4.3).

The evaluation is carried out on a desktop PC with an Intel Xeon CPU at 2.4GHz with 16GB of RAM and a Nvidia Quadro K5200 GPU with 8GB of VRAM. As for the implementation of our method, although the CNN network works on an input/output resolution of 304×228 [16], both the input frame and the predicted depth map are converted to 320×240 as input for all other stages. Also, the CNN-based depth prediction and semantic segmentation are run on the GPU, while all other stages are implemented on the CPU, and run on two different CPU threads, one devoted to frame-wise processing stages (camera pose estimation and depth refinement), the other carrying out key-frame related processing stages (key-frame initialization, pose graph optimization and global map and semantic label fusion), so to allow our entire framework to run in real-time.

We use sequences from two public benchmark datasets, i.e. the *ICL-NUIM* dataset [8] and *TUM RGB-D SLAM* dataset [26], the former synthetic, the latter acquired with a Kinect sensor. Both datasets provide ground truth in terms of camera trajectory and depth maps. In all our experiments, we used the CNN model trained on the indoor sequences of the *NYU Depth v2* dataset [25], to test the generalization capability of the network to unseen environments; also because this dataset includes both depth ground-truth (represented by depth maps acquired with a Microsoft Kinect camera) and pixel-wise semantic label annotations, necessary for semantic label fusion. In particular, we train the semantic segmentation network on the official train split of the labeled subset, while the depth network is trained using more frames from the raw NYU dataset, as reported in [16]. Semantic annotations consist of the 4 super-classes *floor*, *vertical structure*, *large structure/furniture*, *small structure*. Noteworthy, the settings of the training dataset are quite different from those on which we evaluate our method, since they encompass different camera sensors, viewpoints and scene layouts. For example, *NYU Depth v2* includes many living rooms, kitchens and bedrooms, which are missing in TUM RGB-D SLAM, being focused on office rooms with desks, objects and people.

4.1. Comparison against SLAM state of the art

We compare our approach against the publicly available implementations of LSD-SLAM[1] [4] and ORB-SLAM[2] [20], two state-of-the-art methods in monocular SLAM representatives of, respectively, direct and feature-based methods. For completeness, we also compare against REMODE [23], state-of-the-art approach focused on dense monocular depth map estimation. The implementation of REMODE has been taken from the author's code[3]. Finally, we also compare our method to the one in [16], that uses the CNN-predicted depth maps as input for a state-of-the-art depth-based SLAM method (point-based fusion[11, 27]), based on the available implementation from the authors of [27][4]. Given the ambiguity of monocular SLAM approaches to estimate absolute scale, we also evaluate LSD-SLAM by bootstrapping its initial scale using the ground-truth depth map, as done in the evaluation in [4, 20]. As for REMODE, since it requires as input the camera pose estimation at each frame, we use the trajectory and key-frames estimated by LSD-SLAM with bootstrapping.

Following the evaluation methodology proposed in [26], Table 1 reports the camera pose accuracy based on the Absolute Trajectory Error (ATE), computed as the root mean

[1] github.com/tum-vision/lsd_slam
[2] github.com/raulmur/ORB_SLAM2
[3] https://www.github.com/uzh-rpg/rpg_open_remode
[4] campar.in.tum.de/view/Chair/ProjectInSeg

Table 1. Comparison in terms of Absolute Trajectory Error [m] and percentage of correctly estimated depth on ICL-NUIM and TUM datasets (TUM/seq1: *fr3/long_office_household*, TUM/seq2: *fr3/nostructure_texture_near_withloop*, TUM/seq3: *fr3/structure_texture_far*.

	Abs. Trajectory Error					**Perc. Correct Depth**					
	Our Method	LSD-BS [4]	LSD [4]	ORB [20]	Laina [16]	Our Method	LSD-BS [4]	LSD [4]	ORB [20]	Laina [16]	Remode [23]
ICL/office0	**0.266**	0.587	0.528	0.430	0.337	**19.410**	0.603	0.335	0.018	17.194	4.479
ICL/office1	**0.157**	0.790	0.768	0.780	0.218	**29.150**	4.759	0.038	0.023	20.838	3.132
ICL/office2	0.213	**0.172**	0.794	0.860	0.509	**37.226**	1.435	0.078	0.040	30.639	16.7081
ICL/living0	**0.196**	0.894	0.516	0.493	0.230	12.840	1.443	0.360	0.027	**15.008**	4.479
ICL/living1	**0.059**	0.540	0.480	0.129	0.060	**13.038**	3.030	0.057	0.021	11.449	2.427
ICL/living2	0.323	**0.211**	0.667	0.663	0.380	26.560	1.807	0.167	0.014	**33.010**	8.681
TUM/seq1	**0.542**	1.717	1.826	1.206	0.809	12.477	3.797	0.086	0.031	**12.982**	9.548
TUM/seq2	0.243	**0.106**	0.436	0.495	1.337	**24.077**	3.966	0.882	0.059	15.412	12.651
TUM/seq3	0.214	**0.037**	0.937	0.733	0.724	**27.396**	6.449	0.035	0.027	9.450	6.739
Avg.	**0.246**	0.562	0.772	0.643	0.512	**22.464**	3.032	0.226	0.029	18.452	7.649

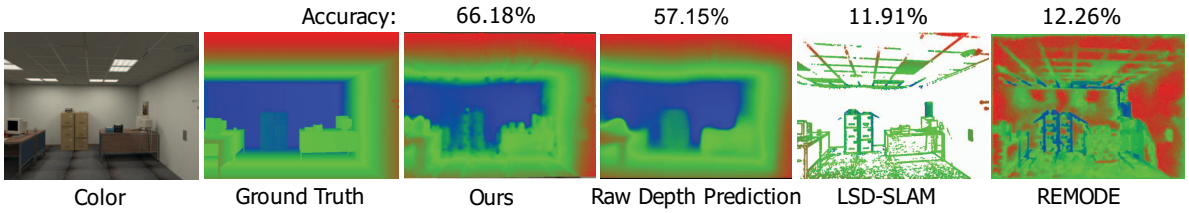

Figure 4. Comparison in terms of depth map accuracy and density among (from the left) the ground-truth, a refined key-frame from our approach, the corresponding raw depth prediction from the CNN, the refined key-frame from LSD-SLAM [4] with bootstrapping and estimated dense depth map from REMODE [23], on the (*office2*) sequence from the *ICL-NUIM* dataset [8]. The accuracy value means correctly estimated depth density on this key-frame.

square error between the estimated camera translation and the ground-truth camera translation for each evaluated sequence. In addition, we assess both reconstruction accuracy and density, by evaluating the percentage of depth values whose difference with the corresponding ground truth depth is less than 10%. Given the observations in the Table, our approach is able to always report a much higher pose trajectory accuracy with respect to monocular methods, due to the their aforementioned absolute scale ambiguity. Interestingly, the pose accuracy of our technique is on average higher than that of LSD-SLAM even after applying bootstrapping, implying an inherent effectiveness of the proposed depth fusion approach rather than just estimating the correct scaling factor. The same benefits are present in terms of reconstruction, being the estimated key-frames not only dramatically more accurate, but also much denser than those reported by LSD-SLAM and ORB-SLAM. Moreover, our approach also reports a better performance in terms of both pose and reconstruction accuracy, also comparing to the technique in [16], where CNN-predicted depths are used as input for SLAM without any refinement, this again demonstrating the effectiveness of the proposed scheme to refine the blurred edges and wrongly estimated depth values predicted by the CNN. Finally, we clearly outperform also REMODE in terms of depth map accuracy.

The increased accuracy with respect to the depth maps estimated by the CNN (as employed in [16]) and by REMODE, as well as the higher density with respect to LSD-SLAM is also shown in Fig. 4. The figure compares the ground-truth with, a refined key-frame using our approach, the corresponding raw depth prediction from the CNN, the refined key-frame from LSD-SLAM [4] using bootstrapping and the estimated dense depth map from REMODE on a sequence of the *ICL-NUIM* dataset. Not only our approach demonstrates a much higher density with respect to LSD-SLAM, but the refinement procedure helps to drastically reduce the blurring artifacts of the CNN-based prediction, increasing the overall depth accuracy. Also, we can note that REMODE tends to fail along low-textured regions, as opposed to our method which can estimate depth densely over such areas by leveraging the CNN-predicted depth values.

4.2. Accuracy under pure rotational motion

As mentioned, one of the advantages of our approach compared to standard monocular SLAM is that, under pure rotational motion, the reconstruction can still be obtained by relying on CNN-predicted depths, while other methods would fail given the absence of a stereo baseline between consecutive frames. To portray this benefit, we evaluate our method on the (*fr1/rpy*) sequence from the *TUM* dataset, mostly consisting of just rotational camera motion. The reconstruction obtained by, respectively, our approach and LSD-SLAM compared to ground-truth are shown in Fig-

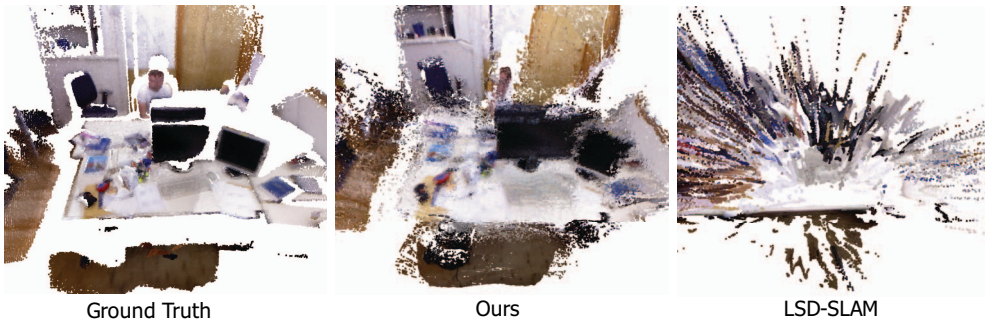

Figure 5. Comparison on a sequence that includes mostly pure rotational camera motion between the reconstruction obtained by ground truth depth (left), proposed method (middle) and LSD-SLAM [4] (right).

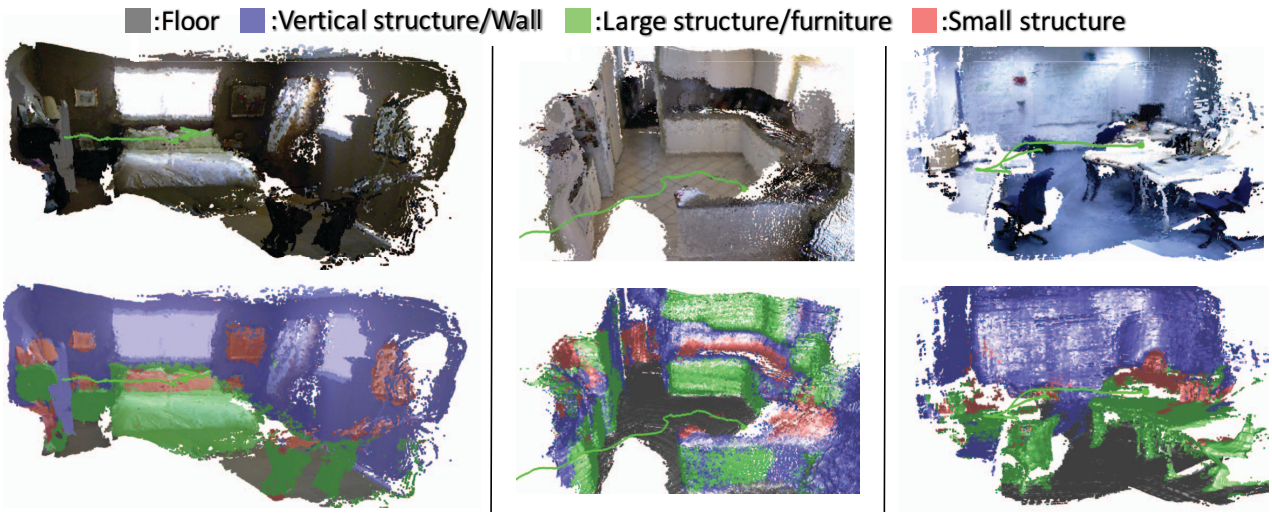

Figure 6. The results of reconstruction and semantic label fusion on the office sequence (top, acquire by our own) and one sequence (*kitchen_0046*) from the *NYU Depth V2* dataset [25] (bottom). Reconstruction is shown with colors (left) and with semantic labels (right).

ure 5. As it can be seen, our method can reconstruct the scene structure even if the camera motion is purely rotational, while the result of LSD-SLAM is significantly noisy, since the stereo baseline required to estimate depth is for most frames not sufficient. We also tried ORB-SLAM on this sequence but it completely fails, given the lack of the necessary baseline to initialize the algorithm.

4.3. Joint 3D and semantic reconstruction

Finally, we show some qualitative results of the joint 3D and semantic reconstruction achieved by our method. Three examples are shown in Fig. 6, which reports an office scene reconstructed from a sequence acquired with our own setup and two sequences from the test set of the *NYU Depth V2* dataset [25]. Another example from the sequence *living0* of the *ICL-NUIM* dataset is shown in Fig.1,c). The Figures also report, in green, the estimated camera trajectory. To the best of our knowledge, this is the first demonstration of joint 3D and semantic reconstruction with a monocular camera. Additional qualitative results in terms of pose and reconstruction quality as well as semantic label fusion are included in the supplementary material.

5. Conclusion

We have shown how the integration of SLAM with depth prediction via a deep neural network is a promising direction to solve inherent limitations of traditional monocular reconstruction, especially with respect to estimating the absolute scale, obtaining dense depths along texture-less regions and dealing with pure rotational motions. The proposed approach to refine CNN-predicted depth maps with small baseline stereo matching naturally overcomes these issues while retaining the robustness and accuracy of direct monocular SLAM in presence of camera translations and high image gradients. The overall framework is capable of jointly reconstructing the scene while fusing semantic segmentation labels with the global 3D model, opening new perspectives towards scene understanding with a monocular camera. A future research avenue is represented by closing the loop with depth prediction, i.e. improving depth estimation by means of geometrically refined depth maps.

References

[1] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006. 2

[2] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *In Proc. Int. Conf. Computer Vision (ICCV)*, 2015. 2, 3

[3] D. Eigen, C. Puhrsch, and R. Fergus. Prediction from a single image using a multi-scale deep network. In *Proc. Conf. Neural Information Processing Systems (NIPS)*, 2014. 2, 3, 4

[4] J. Engel, T. Schps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision (ECCV)*, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[5] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *IEEE International Conference on Computer Vision (ICCV)*, December 2013. 2, 5

[6] D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel. Real-time monocular object slam. *Robot. Auton. Syst.*, 75(PB), jan 2016. 2

[7] W. N. Greene, K. Ok, P. Lommel, and N. Roy. Multi-level mapping: Real-time dense monocular slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016. 3

[8] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014. 4, 5, 6, 7

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[10] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *In Computer Vision and Pattern Recognition (CVPR)*, 2005. 2, 3

[11] 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 *International Conference on 3D Vision (3DV)*, pages 1–8. Ieee, 2013. 1, 2, 3, 6

[12] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *In Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007. 2, 3

[13] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *European Conference on Computer Vision (ECCV)*, 2008. 2

[14] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A General Framework for Graph Optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011. 5

[15] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *Int. Conf. on Robotics and Automation (ICRA)*, 2014. 2

[16] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *IEEE International Conference on 3D Vision (3DV) (arXiv:1606.00373)*, October 2016. 2, 3, 4, 6, 7

[17] B. Li, C. Shen, Y. Dai, A. V. den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, 2015. 3

[18] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *In Computer Vision and Pattern Recognition (CVPR)*, 2010. 2, 3

[19] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 5162–5170, 2015. 3

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards. Orb-slam: A versatile and accurate monocular slam system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015. 1, 2, 3, 6, 7

[21] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, oct 2011. 1, 2

[22] R. A. Newcombe, S. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011. 1, 2, 3

[23] M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 6, 7

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 4

[25] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6, 8

[26] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, oct 2012. 6

[27] K. Tateno, F. Tombari, and N. Navab. Real-time and scalable incremental segmentation on dense slam. 2015. 6

[28] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Perez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 2

[29] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, 2015. 2, 3, 4

[30] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. Mcdonald. Real-time large scale dense

RGB-D SLAM with volumetric fusion. *Intl. J. of Robotics Research, IJRR*, 2014. 1, 2