TANDEM: Tracking and Dense Mapping in Real-time using Deep Multi-view Stereo

Anonymous Author(s)
Affiliation
Address
email

Abstract: In this paper, we present TANDEM a real-time monocular tracking and dense mapping framework. For pose estimation, TANDEM performs photometric bundle adjustment based on a sliding window of keyframes. To increase the robustness, we propose a novel tracking front-end that performs dense direct image alignment using depth maps rendered from a global model that is built incrementally from dense depth predictions. To predict the dense depth maps, we propose Cascade View-Aggregation MVSNet (CVA-MVSNet) that utilizes the entire active keyframe window by hierarchically constructing 3D cost volumes with adaptive view aggregation to balance the different stereo baselines between the keyframes. Finally, the predicted depth maps are fused into a consistent global map represented as a truncated signed distance function (TSDF) voxel grid. Our experimental results show that TANDEM outperforms other state-of-the-art traditional and learning-based monocular visual odometry (VO) methods in terms of camera tracking. Moreover, TANDEM shows state-of-the-art real-time 3D reconstruction performance. Code will be published upon acceptance.

Keywords: SLAM, Dense Mapping, Multi-view Stereo, Deep Learning

1 Introduction

Real-time dense 3D mapping is one of the major challenges in computer vision and robotics. This problem, known as dense SLAM, includes both estimating the 6DoF pose of a sensor and a dense reconstruction of the surroundings. While there exist numerous well-working and robust RGB-D mapping solutions [1, 2, 3], real-time dense reconstruction from monocular cameras is a significantly more difficult challenge as depth values cannot be simply read out from the sensor and fused. Nevertheless, it is a very important problem, as monocular approaches offer significant advantages over RGB-D [1] or LiDAR-based [4] solutions: They are not limited to indoor environments, they are not limited to near-range sensing and they will be more scalable as they can work with standard cameras that are cheap, lightweight and omnipresent - on smartphones, in cars, on robots and drones.

Several deep neural network (DNN) based approaches have been proposed to tackle the problem of monocular tracking and dense mapping by utilizing monocular depth estimation [5], variational auto-encoders [6, 7, 8], or end-to-end neural networks [9, 10]. Unlike the aforementioned works, in this paper, we propose a novel monocular dense SLAM method, TANDEM, which, for the first time, integrates learning-based multi-view stereo (MVS) into a traditional optimization-based VO. This novel design of dense SLAM shows state-of-the-art tracking and dense reconstruction accuracy as well as strong generalization capability on challenging real-world datasets with the model trained only on synthetic data. Figure 1 shows the 3D reconstructions delivered by TANDEM on unseen sequences. We use the marching cubes algorithm [11] to generate meshes from the TSDF grids.

Our contributions. **(1)** a novel real-time monocular dense SLAM framework that seamlessly couples classical direct VO and learning-based MVS reconstruction; **(2)** to our knowledge, the first monocular dense tracking front-end that utilizes depth rendered from a global TSDF model; **(3)** a novel MVS network, CVA-MVSNet, which is able to leverage the entire keyframe window by utilizing view aggregation and multi-stage depth prediction; **(4)** state-of-the-art tracking and reconstruction results on both synthetic and real-world data.

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

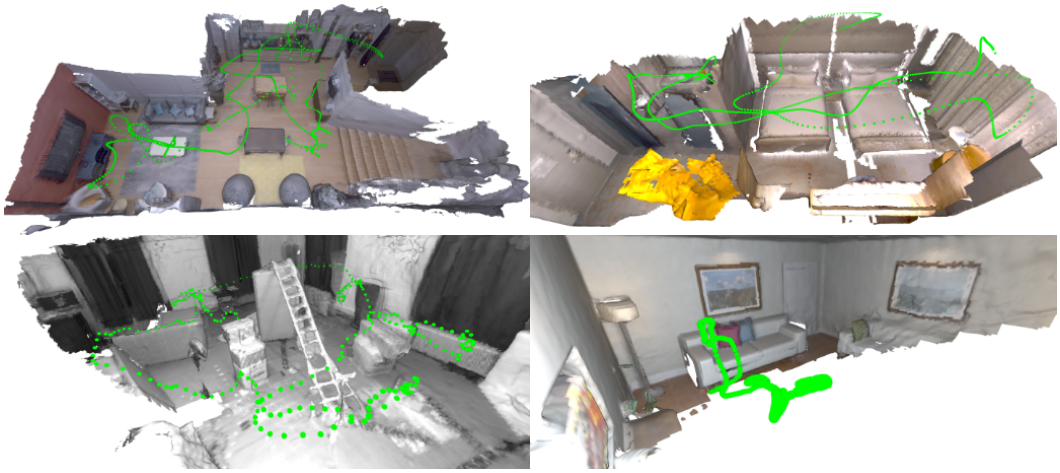


Figure 1: TANDEM is a monocular dense SLAM method that estimates the camera poses and reconstructs the 3D environment in real-time. The figure shows the estimated camera trajectories and the dense 3D models on the sequences of our Replica [12] test split (UL, UR), EuRoC [13] (BL), and ICL-NUIM [14] (BR) using a model trained on the synthetic Replica dataset.

2 Related Work

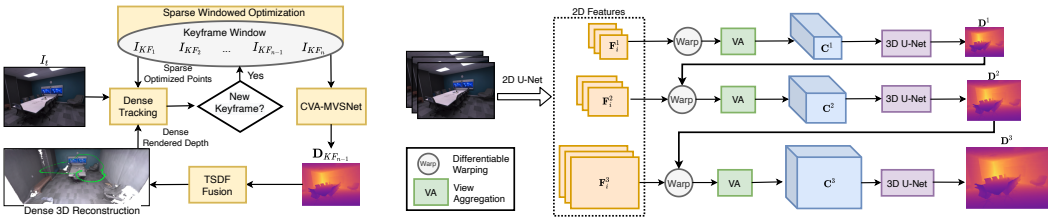
There are two different work streams related to the proposed method. On one side, there is pure 3D reconstruction based on posed images and, on the other side, there are full SLAM or VO frameworks that simultaneously estimate camera poses and a 3D reconstruction of the environment.

3D Reconstruction. Most dense 3D reconstruction approaches consider images and corresponding reference poses as inputs and estimate a dense or partially dense reconstruction of the environment. Over the last decade, several classical methods have been proposed [15, 16, 17, 18].

Recently, deep-learning-based methods have shown superior performance over classical methods. These methods regress a 3D model of the environment utilizing DNNs. This 3D model can be either in the form of a volumetric representation [9, 19, 20, 21], a 3D point cloud [22] or a set of depth maps [23, 24, 25]. Nowadays, most popular are methods which predict the final model from 3D cost volumes. Huang et al. [23] proposed one of the first cost-volume-based approaches. Zhou et al. [25] aggregate multiple image-pair-wise volumes to a single cost volume and use a 3D CNN for depth prediction. Yao et al. [24] propose to directly calculate a single volume based on 2D deep feature maps predicted from the input images. In a follow-up work, Yao et al. [26] replace the depth prediction CNN by a recurrent network. To improve run-time and memory consumption, Gu et al. [27] propose a cascade cost volume. Yi et al. [28] introduce a self-adaptive view aggregation to weight the voxel-wise contribution for each input image. The proposed CVA-MVSNet is built upon the two aforementioned works [27, 28] and largely inspired by them. However, only by their combination and adaption to the SLAM setting we achieve better performance and real-time capability.

While all previous methods are based on a set of selected frames, Murez et al. [9] instead directly predict a TSDF model from a single global 3D cost volume. Weder et al. [19] propose a learning-based alternative to classical TSDF fusion of depth maps. While these volumetric representations, in general, are rather memory intense, Niessner et al. [29] propose voxel-hashing to overcome this limitation and Steinbrücker et al. [30] perform depth map fusion on a CPU using an octree.

RGB-D SLAM. In the area of visual SLAM, RGB-D approaches by nature provide dense depth maps along with the camera trajectory and therefore target to solve a similar problem as our approach. Bylow et al. [31] and Kerl et al. [2, 32] mainly focus on accurate trajectory estimation from RGB-D images. In addition to camera tracking, Newcombe et al. [1] integrate the depth maps into a global TSDF representation. Whelan et al. [3] perform surfel-based fusion and non-rigid surface deformation for globally consistent reconstruction. Kähler et al. [33] use a TSDF map representation which is split into sub maps to model loop closures. While most previous methods optimize only for the frame pose, Schöps et al. [34] propose a full bundle adjustment based direct RGB-D SLAM



(a) TANDEM System Overview. (b) CVA-MVSNet Architecture.

Figure 2: (a) Every new frame is tracked using the optimized sparse points from the visual odometry and the dense depth map rendered from the 3D model. The poses of the keyframes are estimated by sliding-window photometric bundle adjustment and fed into the CVA-MVSNet for dense depth prediction. The depth maps are fused into a globally consistent TSDF volume. (b) CVA-MVSNet builds cascaded cost volumes and hierarchically estimates the depth maps. The view aggregation module effectively aggregates the features of the multi-view images by predicting self-adaptive weights.

which optimizes for both camera pose and 3D structure. Sucar et al. [35] integrate a DNN-based implicit scene representation into an RGB-D SLAM system.

Monocular SLAM. Compared to RGB-D methods, for monocular approaches both tracking and mapping become much more challenging.

Using a single monocular camera, Newcombe et al. [36] perform optimization based on a photometric cost volume to jointly estimate the camera pose and dense depth maps in real-time on a GPU. Engel et al. [37] propose the first large-scale photometric SLAM formulation including loop closure detection and pose graph optimization. By using a sparse representation, Engel et al. [38] were able to formulate the first fully photometric VO framework which jointly estimates pose and depth in real-time. To obtain denser reconstructions, Mur-Artal et al. [39] perform semi-dense probabilistic depth estimation on top of feature-based SLAM [40]. Schöps et al. [41] perform temporal, plane-sweep-based depth estimation using the poses and images obtained from a mobile tracking device.

Tateno et al. [5] and Yang et al. [42, 43] leverage DNNs in a traditional direct SLAM framework to improve tracking performance and overcome the problem of scale ambiguity. While traditional geometric and hybrid approaches still achieve superior tracking performance, there are several fully learned SLAM frameworks [44, 45, 46, 25], which are superior in terms of reconstruction completeness. Jatavallabhula et al. [47] propose a differential optimizer which has the potential to bridge the gap between traditional and learning-based SLAM.

A novel idea for deep-learning-based SLAM is proposed by Bloesch et al. [6]. The authors propose to learn a frame-wise code representation for the scene depth, which is the jointly optimized together with the camera pose. The work of Czarnowski et al. [7] is an extension of [6], where the code representation is integrated into a full SLAM system. Zuo et al. [8] make use of a similar code representation in a visual-inertial setup and furthermore feed sparse, tracked features into a DNN.

3 TANDEM

The proposed TANDEM is comprised of three components: monocular visual odometry (section 3.1), dense depth estimation with CVA-MVSNet (section 3.2), and volumetric mapping (suppl.). Figure 2a shows an overview of the system. The visual odometry utilizes the monocular video stream and the dense depth rendered from the 3D TSDF model to estimate camera poses in a sliding-window manner. Given the keyframes and their estimated poses, the proposed CVA-MVSNet predicts a dense depth map for the reference keyframe. To reconstruct a complete and globally consistent 3D model of the environment, the depth maps are then fused into the TSDF voxel grid [48] with voxel-hashing [29]. By seamlessly integrating these components, the resultant system TANDEM enables real-time tracking and high-quality dense mapping from a monocular camera.

3.1 Visual Odometry

Estimating camera poses by tracking a sparse set of points across multiple frames has shown great performance in recent VO systems [38, 39]. Using more points for the joint optimization, however, does not necessarily further improve the accuracy of the estimated poses while significantly increases the runtime [38]. Therefore, in the proposed VO system we make use of a direct sparse windowed optimization back-end as described in Direct Sparse Odometry (DSO) [38]. However, we utilize dense depth maps rendered from the global TSDF model, which we build up incrementally, in the direct image alignment front-end. In numerous experiments, we confirm that this combination of dense tracking front-end and sparse back-end optimization significantly improves the tracking performance (cf. Table 4) while maintaining a fast runtime.

Dense Front-end Tracking. The front-end tracking provides camera-rate pose estimations and serves as initialization for the windowed optimization back-end. In the original DSO, a new frame is tracked against the last keyframe n by direct image alignment using a sparse depth map $\mathbf{D}_n^{\text{DSO}}[\mathbf{p}]$ generated from all points in the optimization window. This approach, however, lacks robustness (cf. Table 4) due to the sparsity of the depth map. We alleviate the issue by incorporating a dense depth map $\mathbf{D}_n^{\text{TSDF}}$ which is rendered from the constructed TSDF model. For each pixel $\mathbf{p}$ in the current keyframe n, we assign a depth value either based on the sparse VO points $\mathbf{D}_n^{\text{DSO}}[\mathbf{p}]$, if available, or based on the rendered dense depth $\mathbf{D}_n^{\text{TSDF}}[\mathbf{p}]$, otherwise. Due to the incrementally-built TSDF model, the combined depth buffer might not contain valid depth values for all pixels but it is much denser in comparison to using the sparse depth values only. The nearly-dense combined depth map is used for two-frame direct image alignment.

3.2 CVA-MVSNet

Let $\{(I_i, \mathbf{T}_i)\}_{i=1}^n$ be the set of active keyframes where I_i is the image of size (H, W) and $\mathbf{T}_i$ is the corresponding global pose estimated by the VO. CVA-MVSNet is based on the principles of multi-view stereo [49] and further leverages deep neural networks [24] to estimate a depth map for the reference frame I_{n-1}. CVA-MVSNet overcomes the prohibitive memory requirement of deep MVS networks by hierarchically estimating the depth using cascaded cost volumes and aggregates the deep features of all the keyframes effectively with a self-adaptive view aggregation module.

As shown in Figure 2b, the multi-scale deep features $\mathbf{F}_i^s$ of the keyframes are firstly extracted by 2D U-Nets with shared weights, where $i \in [1, n]$ is the frame index and $s \in [1, 3]$ is the scale index. As a result, $\mathbf{F}_i^s$ is of the shape (F^s, H^s, W^s) where F^s is the feature dimension of the scale s, $H^s = H/2^{3-s}$, and $W^s = W/2^{3-s}$. The depth map of the reference frame is estimated hierarchically with 3 stages each of which takes the set of features $\{\mathbf{F}_i^s\}_{i=1}^n$ as the inputs and predicts the reference depth map $\mathbf{D}^s$ of shape (H^s, W^s). For clarity, we first explain how a single stage estimates the depth and then describe how multiple stages are assembled hierarchically.

Single Stage Depth Estimation. For each stage, a cost volume $\mathbf{C}^s$ needs to be constructed using the deep features $\{\mathbf{F}_i^s\}_{i=1}^n$. For each pixel of the reference frame, we define D^s depth hypotheses, which results in a tensor $\mathbf{D}_{hyp}^s$ of shape (D^s, H^s, W^s). The deep features $\mathbf{F}_i^s$ of each frame are geometrically transformed with differentiable warping [50] using the depth hypotheses, the relative pose $\mathbf{T}_j^i = \mathbf{T}_i^{-1}\mathbf{T}_j$ and the camera intrinsics. As a result, a feature volume $\mathbf{V}_i^s$ of shape (F^s, D^s, H^s, W^s) is constructed for every frame.

In order to aggregate the information from multi-view feature volumes into one cost volume $\mathbf{C}^s$, most prior deep MVS methods treat different views equally and use a variance-based cost metric:

$$\mathbf{C}^s = \frac{\sum_{i=1}^n (\mathbf{V}_i^s - \bar{\mathbf{V}}^s)^2}{n}, \quad \text{where} \quad \bar{\mathbf{V}} = \frac{\sum_{i=1}^n \mathbf{V}_i^s}{n}. \tag{1}$$

However, in the sliding-window SLAM setting, the keyframes are not evenly distributed within the optimization window – typically the distance between newer keyframes is much smaller than between older keyframes. This causes considerable occlusion and non-overlapping images. The variance-based cost volume, which weighs different views equally, is thus inappropriate. To alleviate this issue, we employ self-adaptive view aggregation [28] to construct the cost volume:

$$\mathbf{C}^s = \frac{\sum_{i=1, i\neq j}^n (1 + \mathbf{W}_i^s) \odot (\mathbf{V}_i^s - \mathbf{V}_j^s)^2}{n-1}, \tag{2}$$

where the view aggregation weights $\mathbf{W}_i^s$ have the shape $(1, D^s, H^s, W^s)$ and $\odot$ is element-wise multiplication with broadcasting. The view aggregation weights $\mathbf{W}_i^s$ are estimated by a shallow 3D convolutional network for each $\mathbf{V}_i^s$ separately by taking $(\mathbf{V}_i^s - \mathbf{V}_j^s)^2$ as the input. This aggregation module allows the network to adaptively downweight erroneous information.

The cost volume $\mathbf{C}^s$ is then regularized using a 3D U-Net and finally passed through a softmax non-linearity to obtain a probability volume $\mathbf{P}^s$ of shape (D^s, H^s, W^s). Given the per-pixel depth hypotheses $\mathbf{D}_{hyp}^s$ of shape (D^s, H^s, W^s) the estimated depth is given as the expected value

$$\mathbf{D}^s[h, w] = \sum_{d=1}^{D} \mathbf{P}^s[d, h, w] \cdot \mathbf{D}_{hyp}^s[d, h, w]. \tag{3}$$

Hierarchical Depth Estimation. The network leverages the depth estimated from the previous stage $\mathbf{D}^{s-1}(s > 1)$ to define a fine-grained depth hypothesis tensor $\mathbf{D}_{hyp}^s$ with a small D^s. Since no prior stage exists for the first stage, each pixel of $\mathbf{D}_{hyp}^1$ has the same depth range defined by $[d_{min}, d_{max}]$ with $D^1 = 48$ depth values. For the later stages ($s > 1$), the depth $\mathbf{D}^{s-1}$ is upsampled and then used as a prior to define $\mathbf{D}_{hyp}^s$. Specifically, for the pixel location (h, w), $\mathbf{D}_{hyp}^s(\cdot, h, w)$ is defined using the upsampled $\mathbf{D}^{s-1}(h, w)$ as the center and then sampled D^s values around it using a pre-defined offset [27]. In this way, fewer depth planes are needed for the stage with a higher resolution, i.e., $D^1 \geq D^2 \geq D^3$. We train the network using the $L1$ loss applied on all three stages with respect to the ground-truth depth and use the sum as the final loss function.

3.3 Implementation Details

To guarantee real-time execution, TANDEM leverages parallelism on multiple levels. Dense tracking and bundle adjustment are executed in parallel threads on the CPU while, asynchronously and in parallel, TSDF fusion and DNN inference are run on the GPU. We train our CVA-MVSNet in PyTorch [51] and perform inference in C++ using TorchScript.

TANDEM can processes images at ca. 20 FPS while running on a desktop with an Nvidia RTX 2080 with 8 GB VRAM, and an Intel i7-9700K CPU. This includes tracking and dense TSDF mapping but no visualization or mesh generation. We refer the reader to the supplementary for further details.

4 Experimental Results

As TANDEM is a complete dense SLAM system, we evaluate it for both monocular camera tracking and dense 3D reconstruction. Specifically, for the camera tracking, we compare with the state-of-the-art traditional sparse monocular odometry, DSO [38], and learning-based dense SLAM method, DeepFactors [7]. For the 3D reconstruction, we compare with a state-of-the-art deep multi-view stereo method, Cas-MVSNet [27], end-to-end reconstruction method, Atlas [9], learning-based dense SLAM methods, CNN-SLAM [5], DeepFactors [7], and CodeVIO [8], as well as iMAP[1] [35], a recently proposed RGB-D dense SLAM method using a deep implicit map representation [52, 53].

In the following, we will first introduce the datasets we use for training and evaluation. Note that only CVA-MVSNet needs to be trained while the dense tracking part of TANDEM is purely optimization-based and does not require training on specific datasets. Then, the ablation study for TANDEM demonstrates different design choices. In the end, we show quantitative and qualitative comparisons with other state-of-the-art methods. Due to the limited space, we demonstrate only part of the conducted evaluation and refer to the supplementary material for additional experiments.

4.1 Datasets

Training sets. We train two models for CVA-MVSNet: One on the real-world **ScanNet** [54] dataset and the other one on the synthetic **Replica** dataset [12]. The ScanNet dataset consists of 1513 indoor scenes and we use the official train and test split for training CVA-MVSNet to give a fair comparisons with DeepFactors and Atlas. However, the geometry and texture in ScanNet show noticeable artifacts and incompleteness [12], which limits its potential to training high-quality dense

[1] We thank the authors of iMAP for providing us with evaluation data and scripts from their paper.

Table 1: Ablation study of CVA-MVSNet on Replica [12]. Using all keyframes within the VO window (*Win*) does not improve the baseline. However, combining *Win* with view aggregation (*VA*) yields more accurate results at the cost of increased inference runtime and memory. By reducing the number of depth planes (*S*) from $(48, 32, 8)$ to $(48, 4, 4)$ we retain high quality and guarantee the real-time performance of TANDEM. Best shown in **bold** and second best shown underlined.

	Win	VA	S	Abs↓ [cm]	a_1 ↑ [%]	a_2 ↑ [%]	a_3 ↑ [%]	Time↓ [ms]	Mem.↓ [MiB]
baseline [27]				2.64	98.64	82.33	20.12	**142**	3447
+ VO window	✓			2.64	98.39	83.55	20.56	215	4117
+ View aggregation	✓	✓		**1.92**	**99.00**	**88.59**	26.03	288	4117
+ Fewer depth planes	✓	✓	✓	2.33	98.51	86.93	**26.09**	158	**2917**

Table 2: Depth evaluation on ICL-NUIM [14]. We show the percentage of pixels for which the estimated depth falls within 10% of the groundtruth value.

Sequence	CNN-SLAM [5]	DeepFactors [7]	Atlas [9]	Cas-MVSNet [27]	Ours (ScanNet)	Ours (Replica)
icl/office0	19.41	30.17	28.79	77.73	52.34	**84.04**
icl/office1	29.15	20.16	62.89	88.87	61.83	**91.18**
icl/living0	12.84	20.44	83.16	**97.06**	86.42	97.00
icl/living1	13.04	20.86	36.93	84.11	71.35	**90.62**
Average	19.77	30.17	66.93	86.94	67.99	**90.71**

reconstruction methods. To complement the ScanNet dataset, we build upon the recently proposed Replica dataset [12], which consists of 18 photorealistic scenes. These scenes were captured from real-world rooms using a state-of-the-art 3D reconstruction pipeline and show very high-quality texture and geometry. Because the authors did not release any sequences, we extend the dataset by manually creating realistic camera trajectories that yield 55 thousand poses and images.

Evaluation sets. We use the **ICL-NUIM** [14] dataset and the Vicon Room sequences of the **EuRoC** dataset [13] for evaluating both the camera tracking and the dense 3D reconstruction. Note that TANDEM is not trained on either of the datasets. ICL-NUIM is a synthetic indoor dataset with both pose and dense depth ground-truth. It contains low-texture scenes which are challenging for both monocular visual odometry and dense depth estimation. EuRoC is a real-world dataset recorded by a micro aerial vehicle (MAV).

4.2 Camera Pose Estimation

TANDEM is a dense SLAM system that estimates camera poses in real-time. We evaluate our method against the monocular SLAM methods DSO [38] and DeepFactors [7] on both ICL-NUIM and EuRoC. We add CodeVIO [8] which uses a camera and an IMU sensor as a reference. We also implement a variant of DSO (DSO + Dense Depth) which uses all the pixels of the dense depth maps estimated by CVA-MVSNet for the front-end direct image alignment. Note that the difference between TANDEM and DSO + Dense Depth is that TANDEM tracks against the *global 3D model* by rendering the depth maps from the TSDF grid. TANDEM achieves overall better tracking accuracy and robustness than the other monocular methods on both ICL-NUIM and EurRoC. Due to the limited space, we show the results on EuRoC in Table 4 of the main paper and kindly refer to the supplementary material for the results on ICL-NUIM.

All the methods except for CodeVIO [8] are run five times for each sequence and reported with the mean RMSE error and standard deviation in terms of absolute pose estimations after Sim(3) alignment with ground-truth. For CodeVIO, we directly take the numbers reported in their paper. The comparison with DSO and DSO + Dense Depth indicates that the proposed dense tracking against the global 3D model improves the camera pose estimations, especially on the more challenging sequences (V102 and V202). However, we should admit that TANDEM still cannot compete with CodeVIO which uses an IMU sensor for pose estimations.

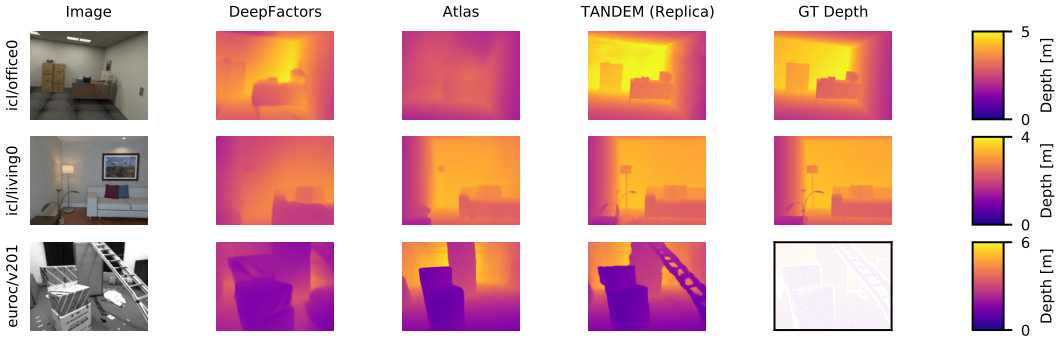


Figure 3: Depth comparison for DeepFactors [7], Atlas [9], and TANDEM on unseen sequences. TANDEM produces finer-scale details, e.g. the plant in the second row, or the ladder in the third row. For EuRoC only sparse ground-truth depth is available. A high-resolution version of this figure can be found in the supplementary material.

Table 3: Depth evaluation on EuRoC [13]. We show the percentage of correct pixels d_1 as in [55].

Sequence	CodeVIO[8]	DeepFactors[7]	Atlas[9]	Cas-MVSNet[27]	Ours (ScanNet)	Ours (Replica)
EuRoC/V101	86.99	71.82	57.16	93.05	93.69	**94.25**
EuRoC/V102	78.65	X	**92.29**	88.03	89.62	90.50
EuRoC/V201	77.32	71.85	93.28	95.43	96.84	**97.17**
EuRoC/V202	71.98	68.26	64.33	93.16	92.65	**95.68**
Average	78.74		76.77	92.42	93.20	**94.40**

4.3 Ablation Study

We conduct the ablation study of CVA-MVSNet on the test split of Replica and show the results in Table 1. Specifically, we evaluate the effectiveness of using the full VO window with 7 keyframes (Win), the view aggregation module (VA), and fewer depth planes (S) with $(48, 4, 4)$ for (D^1, D^2, D^3). The baseline method is the original Cas-MVSNet using 3 multi-view images as the inputs, no view aggregation module, and more depth planes $(48, 32, 8)$. We use the absolute difference (Abs), and the percentage of inliers with different thresholds (a_1, a_2, a_3) as the metrics for the depth map evaluation. Please refer to the supplementary material for the formulas of the metrics. In addition, we also show the inference time and the memory usage of different models. From the table, we can see that using the entire keyframe window with more frames does not improve the accuracy over the baseline model while increasing the runtime and memory usage. With the view aggregation module, the accuracy is significantly improved, but the runtime further increases. Using fewer depth planes does not show a significant drop in accuracy but improves the runtime and memory usage a lot. Therefore, to guarantee the real-time performance of TANDEM, we use the fewer-plane model as the final CVA-MVSNet and all other experiments in the paper are conducted with this model.

4.4 3D Reconstruction

We evaluate the reconstruction accuracy on both ICL-NUIM and EuRoC. On ICL-NUIM we compare with DeepFactors [7], CNN-SLAM [5], Atlas [9], and Cas-MVSNet [27]. Table 2 shows the evaluation results. Since Atlas, a pure 3D reconstruction method, does not estimate poses, we provide ground-truth poses as the input. Note that Atlas estimates a TSDF volume directly from a 3D CNN, so we render the depth maps for evaluation against other methods. CNN-SLAM, DeepFactors, and Atlas are trained on ScanNet. For Cas-MVSNet, we re-train it on the Replica dataset and use the same poses as for our CVA-MVSNet to show a fair comparison. The depth maps of monocular methods are aligned in scale based on the trajectory and further details are given in the supplementary. For our methods, we show the results of the evaluations for the ScanNet model and the Replica model. We use the a_1 metric as the major measurement for accuracy. From Table 2, we can see that our method shows a notable improvement in comparison to the other methods and delivers the best result on average. Note that CVA-MVSNet achieves better results with the model trained on the synthetic Replica than the model trained on ScanNet.

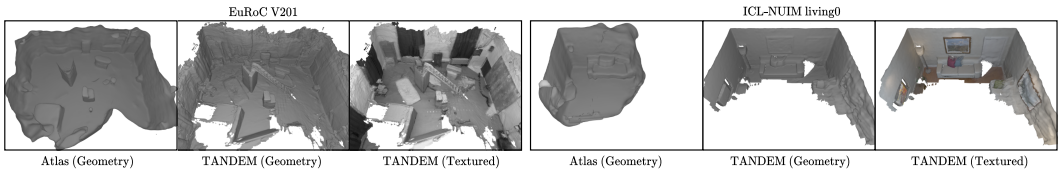


Figure 4: Qualitative comparison of Atlas [9] and TANDEM on unseen sequences. Atlas does not construct textured meshes, so we also render the pure geometry from TANDEM for comparison.

Table 4: Pose evaluation on EuRoC [13]. All the methods are Sim(3) aligned w.r.t. the ground-truth trajectories. The mean absolute pose errors and the standard deviations over five runs are shown.

Sequence	CodeVIO[8]	DeepFactors[7]	DSO [38]	DSO+Dense Depth	Ours
EuRoC/V101	**0.06** (-)	1.48 (±0.134)	0.10 (±0.006)	0.09 (±0.002)	0.09 (±0.001)
EuRoC/V102	**0.07** (-)	Lost	0.27 (±0.017)	0.28 (±0.015)	0.17 (±0.006)
EuRoC/V201	**0.10** (-)	1.06 (±0.441)	0.09 (±0.005)	0.09 (±0.003)	0.09 (±0.002)
EuRoC/V202	**0.06** (-)	1.89 (±0.019)	0.21 (±0.020)	0.19 (±0.022)	0.12 (±0.009)

On EuRoC, we cannot compare with CNN-SLAM since the numbers on EuRoC are not provided on the paper and code is not publicly available. We further add CodeVIO [8] into the evaluation on EuRoC as it is a recent dense SLAM system and it was also evaluated on EuRoC. Please note that CodeVIO uses a monocular camera and an inertial sensor for tracking, while TANDEM and other SLAM methods rely only on monocular cameras. Table 3 shows the evaluation results.

We further evaluate TANDEM against iMAP [35], an RGB-D dense SLAM system that leverages deep implicit map representation. Please note two major differences between iMAP and TANDEM: on one hand, iMAP uses an RGB-D sensor while TANDEM needs only a monocular camera; on the other hand, the DNN of iMAP does not require any pre-training and it is trained purely online with the RGB-D inputs, while the depth estimation network of TANDEM requires offline training. Since iMAP was also evaluated on Replica, we, therefore, compare TANDEM with iMAP on the two sequences of their dataset which are not included in the training set of our Replica split. We use the evaluation metrics from iMAP and show the results in Table 5. In general, TANDEM achieves similar results to iMAP while using a monocular camera.

In Figure 3 we show qualitative depth maps estimated by DeepFactors, Atlas, and TANDEM. Both DeepFactors and Atlas can recover the geometry of the underlying scene well, but our method generally manages to capture more fine-scale details. We further show the complete scene reconstruction as meshes in Figure 4. As DeepFactors does not generate a complete 3D model by itself, we only compare TANDEM with Atlas for this experiment. From the figure we can see that, similarly to the depth maps, TANDEM is able to reconstruct more fine-scale details than Atlas.

5 Conclusion

We presented TANDEM, a real-time dense monocular SLAM system with a novel design that couples direct photometric visual odometry and deep multi-view stereo. In particular, we propose CVA-MVSNet which leverages the whole keyframe window effectively and predicts high-quality depth maps. Further, the proposed dense tracking scheme bridges camera pose estimation and dense 3D reconstruction by tracking against the global 3D model created with TSDF fusion. The quantitative and qualitative experiments show that TANDEM achieves better results than other state-of-the-art methods for both 3D reconstruction and visual odometry on synthetic and real-world data. We believe that TANDEM further bridges the gap between RGB-D mapping and monocular mapping.

Table 5: Comparison to iMAP [35]. TANDEM shows comparable performance to iMAP, which uses RGB-D data, but no training prior to scanning. The mesh-based 3D metrics are as in [35].

	room-1			office-4		
	Acc [cm]	Comp [cm]	CR [%]	Acc [cm]	Comp [cm]	CR [%]
iMAP	**3.69**	4.87	**83.45**	4.83	6.59	**77.63**
TANDEM	4.26	**4.71**	81.95	**3.76**	**6.11**	74.66

References

[1] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, and S. Hodges. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[2] C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[3] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison. ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, 2015.

[4] J. Zhang and S. Singh. Visual-lidar odometry and mapping: low-drift, robust, and fast. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

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

[6] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM - learning a compact, optimisable representation for dense visual SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison. DeepFactors: Real-time probabilistic dense monocular SLAM. *IEEE Robotics and Automation Letters*, 2020.

[8] X. Zuo, N. Merrill, W. Li, Y. Liu, M. Pollefeys, and G. Huang. CodeVIO: Visual-inertial odometry with learned optimizable dense depth. *arXiv preprint arXiv:2012.10133*, 2020.

[9] Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich. Atlas: End-to-end 3D scene reconstruction from posed images. In *European Conference on Computer Vision (ECCV)*, 2020.

[10] C. Tang and P. Tan. BA-Net: Dense bundle adjustment networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[11] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1987.

[12] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[13] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[14] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[15] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[16] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition - 32nd DAGM Symposium*, 2010.

[17] 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.

[18] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[19] S. Weder, J. L. Schönberger, M. Pollefeys, and M. R. Oswald. RoutedFusion: Learning real-time depth map fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang. SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[21] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[22] R. Chen, S. Han, J. Xu, and H. Su. Point-based multi-view stereo network. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[23] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang. DeepMVS: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018.

[25] H. Zhou, B. Ummenhofer, and T. Brox. DeepTAM: Deep tracking and mapping. In *European Conference on Computer Vision (ECCV)*, 2018.

[26] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan. Recurrent MVSNet for high-resolution multi-view stereo depth inference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[28] H. Yi, Z. Wei, M. Ding, R. Zhang, Y. Chen, G. Wang, and Y. Tai. Pyramid multi-view stereo net with self-adaptive view aggregation. In *European Conference on Computer Vision (ECCV)*, 2020.

[29] M. Niessner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013.

[30] F. Steinbrücker, J. Sturm, and D. Cremers. Volumetric 3D mapping in real-time on a CPU. In *ICRA*, pages 2021–2028. IEEE, 2014.

[31] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers. Real-time camera tracking and 3D reconstruction using signed distance functions. In *Robotics: Science and Systems (RSS)*, 2013.

[32] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[33] O. Kähler, V. A. Prisacariu, and D. W. Murray. Real-time large-scale dense 3D reconstruction with loop closure. In *European Conference on Computer Vision (ECCV)*, 2016.

[34] T. Schöps, T. Sattler, and M. Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] E. Sucar, S. Liu, J. Ortiz, and A. Davison. iMAP: Implicit mapping and positioning in real-time. In *arXiv preprint arXiv:2103.12352*, 2021.

[36] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[37] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, 2014.

[38] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[39] R. Mur-Artal and J. D. Tardos. Probabilistic semi-dense mapping from highly accurate feature-based monocular SLAM. In *Robotics: Science and Systems (RSS)*, 2015.

[40] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 2015.

[41] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys. Large-scale outdoor 3D reconstruction on a mobile device. *Computer Vision and Image Understanding*, 2017.

[42] N. Yang, R. Wang, J. Stuckler, and D. Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision (ECCV)*, 2018.

[43] N. Yang, L. von Stumberg, R. Wang, and D. Cremers. D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[44] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[45] Z. Yin and J. Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] K. M. Jatavallabhula, G. Iyer, and L. Paull. ∇SLAM: Dense SLAM meets automatic differentiation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2130–2137, 2020.

[48] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.

[49] Y. Furukawa and C. Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[50] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Conference on Neural Information Processing Systems (NeurIPS)*. 2019.

[52] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[53] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[54] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[55] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Conference on Neural Information Processing Systems (NeurIPS)*, pages 2366–2374. Curran Associates, 2014.