DeepRelativeFusion: Dense Monocular SLAM using Single-Image Relative Depth Prediction

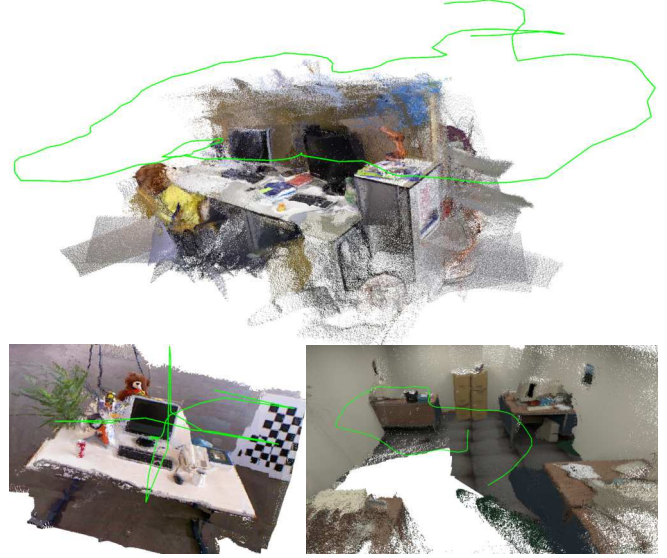Shing Yan Loo[1,2], Syamsiah Mashohor[2], Sai Hong Tang[2] and Hong Zhang[1]

Abstract— Traditional monocular visual simultaneous localization and mapping (SLAM) algorithms have been extensively studied and proven to reliably recover a sparse structure and camera motion. Nevertheless, the sparse structure is still insufficient for scene interaction, e.g., visual navigation and augmented reality applications. To densify the scene reconstruction, the use of single-image absolute depth prediction from convolutional neural networks (CNNs) for *filling in* the missing structure has been proposed. However, the prediction accuracy tends to not generalize well on scenes that are different from the training datasets.

In this paper, we propose a dense monocular SLAM system, named DeepRelativeFusion, that is capable to recover a globally consistent 3D structure. To this end, we use a visual SLAM algorithm to reliably recover the camera poses and semi-dense depth maps of the keyframes, and then combine the keyframe pose-graph with the densified keyframe depth maps to reconstruct the scene. To perform the densification, we introduce two incremental improvements upon the energy minimization framework proposed by DeepFusion: (1) an additional image gradient term in the cost function, and (2) the use of single-image relative depth prediction. Despite the absence of absolute scale and depth range, the relative depth maps can be corrected using their respective semi-dense depth maps from the SLAM algorithm. We show that the corrected relative depth maps are sufficiently accurate to be used as priors for the densification. To demonstrate the generalizability of relative depth prediction, we illustrate qualitatively the dense reconstruction on two outdoor sequences. Our system also outperforms the state-of-the-art dense SLAM systems quantitatively in dense reconstruction accuracy by a large margin.

I. INTRODUCTION

Recovering dense structure from images can lead to many applications, ranging from augmented reality to self-driving. Visual SLAM uses only cameras to recover structure and motion, which provides cheaper solutions to the SLAM problems in comparison to light detection and ranging (LiDAR). Traditional monocular visual SLAM algorithms have shown promising sparse [1]–[3] and semi-dense [4] reconstruction accuracy by reliably matching the texture-rich image regions such as corners and edges. While the sparse structure suffices for localizing the camera, it is not sufficient for the interaction between a moving robot and the environment, e.g., avoid hitting a pedestrian.

Thanks to the ubiquity of graphics processing units (GPUs), computation of a dense structure from an image sequence in real-time has become possible by aggregating the photometric information in bundles of frames [5]. In

[1]The authors are with Department of Computing Science, University of Alberta, Canada.
[2]The authors are with Faculty of Engineering, Universiti Putra Malaysia, Malaysia.

Fig. 1: Qualitative reconstruction of our dense SLAM system on (top) TUM RGB-D [8] fr3_long_office_household, (bottom left) TUM RGB-D fr2_xyz, and (bottom right) ICL-NUIM [9] of_kt2. The green line represents the camera trajectory. Best viewed digitally.

general, the photometric information aggregation seeks to optimize the map by reducing the photometric re-projection errors between the bundles of frames, which is a necessary but not sufficient condition to obtain a globally optimized solution. One inherent limitation is the minimization of photometric re-projection errors in textureless image regions in a bundle of frames as no distinct local minima can be found [5]. Nevertheless, one common practice in recovering depth information in texture-poor regions is to enforce a *smoothness* constraint [6], [7], i.e., the adjacent depth values in the texture-poor image regions change gradually.

Alternatively, the use of constraints from CNN depth [10]–[12] and surface normals [13], [14] predictions has been proposed to recover the 3D structure in texture-poor image regions. Both depth and surface normals provide 3D geometry information, the difference being that surface normals contain local surface orientation (i.e., the relative locations between local space points) while a depth value contains the absolute location of a space point. Therefore, the incorporation of learned 3D geometry into traditional SLAM algorithms have been proposed to solve the monocular dense reconstruction problem.

In this paper, we present a dense SLAM system that augments a monocular SLAM algorithm [4] with a dense

mapping optimization framework. The optimization framework exploits the accurate depth and depth gradient information from single-image relative depth prediction as priors to densify the semi-dense structure provided by the SLAM algorithm. Next, our system combines the optimized dense depth maps with the pose-graph maintained by the SLAM algorithm to produce a globally consistent dense structure (see Figure 1). The experimental results show that our system achieves state-of-the-art dense reconstruction accuracy. Our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose the use of single-image relative depth prediction, as opposed to absolute depth prediction, for providing depth and depth gradient as priors to solve the dense monocular SLAM problem. In the end, we demonstrate a potential use case of using relative depth prediction for achieving a *general* dense monocular SLAM system, overcoming the poor generalizability of absolute depth prediction to diverse scene types.
- We provide empirical evidence that the cost function used in our optimization framework—which incorporates image gradient information with depth and depth gradient priors from relative depth prediction— improves the 3D reconstruction accuracy.
- We show, quantitatively and qualitatively, that relative depth maps are sufficiently accurate to be used for dense scene reconstruction, resulting in the state-of-the-art reconstruction accuracy.

II. RELATED WORK

Traditional monocular SLAM algorithms are capable of producing sparse, semi-dense, and dense structures. Conceptually, sparse refers to the sparsity of the structure as well as the independence of each space point from one another during the structure and motion optimization. During the optimization, each image point (usually a corner) is being matched across frames and mapped, and collectively, the whole structure and camera motion are being optimized in the form photometric [2] or geometric [1], [3] reprojection error minimization. On the other hand, instead of processing the sparse points independently, semi-dense and dense methods employ the notion of the neighbourhood *connectedness* of the points. Dense methods regularize the neighbouring depth pixels using image gradient [5]–[7], typically formulated as a *smoothness* term in an energy minimization framework; whereas the semi-dense method, LSD-SLAM [4], estimates the depth values of the high gradient image regions, thus semi-dense, and regularizes the semi-dense depth map by computing each depth value the weighted average of the neighbouring depth values with the estimated variances as their weight. In this work, rather than only relying on image gradient—as used in dense methods— as a prior to recovering a dense structure, we use LSD-SLAM to reliably recover a semi-dense structure, and then perform densification through regularization of the structure using image gradient, and depth and depth gradient information from single-image depth prediction.

There are two types of single-image depth predictions: absolute depth prediction and relative depth prediction. Absolute depth prediction problem is to train a CNN to predict the metric depth maps from single images [15]–[18]. Because of the CNN prediction range, the CNN training is commonly limited to one scene type, e.g., indoor or outdoor dataset. On the other hand, relative depth prediction is concerned with the estimation of the distance of one space point with respect to the others, i.e., their order in depth, rather than the absolute depth. Early work on relative depth prediction learns from ordinal depth annotations (closer/farther relationship between two points), which contain fairly accurate sparse depth relationships covering a wide range of scene types (e.g., mixing indoor and outdoor scenes in a combined training dataset) [19], [20]. The training results demonstrate accurate ordinal depth prediction quantitatively on different datasets and qualitatively on unconstrained photos taken from the internet, albeit the absence of absolute depth values. To train on larger and diverse datasets, Lasinger et al. propose to train a relative depth prediction CNN, named MiDaS [21], using a scale- and shift-invariant loss, which handles unknown depth scale and global shift in different datasets. By eliminating the absolute scale and shift, the MiDaS's relative depth prediction is essentially constrained to disparity space, and is akin to having surface normals prediction [22] for regularization of neighbouring space points [13], [14], and therefore are particularly suitable as priors for our semi-dense structure densification framework.

Fusions of single-image depth prediction to visual SLAM algorithms have been proposed to solve dense reconstruction problems. One approach to performing depth fusion from multiple viewpoints is through the accumulation of probabilistic distribution of depth observations from the single-image depth prediction [10], [23]. Recently, Czarnowski et al. propose a factor-graph optimization framework named DeepFactors [12], which jointly optimizes the camera motion and the code-based depth maps. Each depth map is parameterized in an n-dimensional code to avoid costly per-pixel depth map optimization. Another dense SLAM system proposed by Laidlow et al., named DeepFusion [11], uses the depth and depth gradient predictions from a CNN to constrain the optimized depth maps. Our proposed system is similar to DeepFusion, except for two key differences: (1) we use depth and depth gradient from relative depth prediction as priors in depth map optimization and (2) we also include an additional image gradient smoothness term in the cost function. As will be discussed later, with the scale- and shift-correction using the semi-dense structure recovered by a SLAM algorithm, the MiDaS's relative depth prediction accuracy is comparable to that of the ground truth depth map, and is qualitatively better than VNLNet's [18] absolute depth prediction (see Figure 4). Furthermore, the image gradient smoothness term has also been used to condition on the depth prediction outputs using the image gradient magnitude in the single-image depth prediction CNN training [15], [16]. Table I shows a side-by-side comparison of the state-of-the-art systems and our proposed system. As expected, the use

TABLE I: Comparison of the state-of-the-art dense monocular SLAM systems and our system

	Depth prediction CNN		Optimization							
	Absolute/ relative	Pre-trained datasets	Scheme	Two-view depth map consistency	Photometric	Geometric (feature) reprojection error	CNN depth consistency	CNN depth gradient consistency	SLAM depth consistency	Image gradient consistency
CNN-SLAM [10]	Absolute	NYU Depth V2	Bayesian filtering	✓					✓	
DeepFusion [11]	Absolute	SceneNet RGB-D	Structure energy minimization				✓	✓	✓	
DeepFactors [12]	Absolute	ScanNet	Structure and motion factor graph optimization	✓	✓	✓				
Ours	Relative	Multiple*	Structure energy minimization	✗	✗		✓	✓	✓	✓

* We use the pre-trained relative depth prediction CNN, MiDaS [21], which has been trained on DIML Indoor, Megadepth, ReDWeb, WSVD and 3D Movies datasets.
Note: We tested the error terms marked with '✗', but did not improve the reconstruction accuracy. We used the geometric consistency loss term proposed in [24] and the standard combination of photometric re-projection error and structural similarity index (SSIM) in [16].
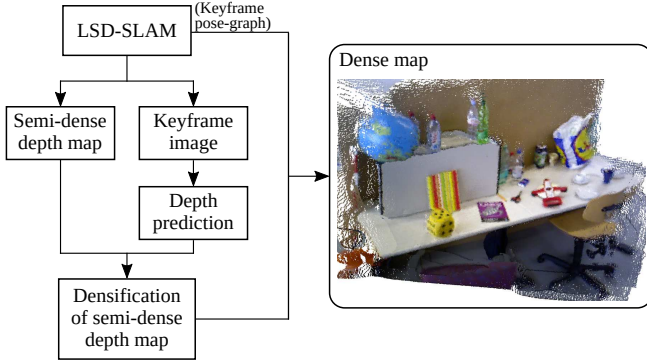
Fig. 2: Our dense monocular SLAM pipeline. We augment the LSD-SLAM [4] with a depth prediction module and a dense mapping module. The optimized keyframe poses from the keyframe pose-graph maintained by LSD-SLAM is combined with the densified depth maps to generate a globally consistent 3D structure.

of absolute depth prediction in CNN-SLAM and DeepFusion helps with the absolute map scale recovery; however, the absolute scale of the depth prediction is lost in DeepFactors' factor graph optimization. And since relative depth prediction does not contain absolute scale, our map is scaled to that of the SLAM map.

III. METHOD

Our proposed dense SLAM system is shown in Figure 2. The system pipeline contains an optimization framework, which uses the predicted depth maps of the keyframe images (see Section III-B) to perform the densification of the semi-dense structure from LSD-SLAM [4] (see Section III-C). To reconstruct the scene, we obtain the latest keyframe poses from the keyframe pose-graph, and then back-project the densified depth maps of the keyframes from their respective poses, assuming the camera intrinsics are known.

A. Notation

In LSD-SLAM, the trajectory of the camera poses and the 3D location of the map points are stored in a list of keyframes. Each keyframe $\mathcal{K}_i$ contains an image $I_i : \Omega \to \mathbb{R}$, a semi-dense inverse depth map $D_{i,\text{semi-dense}} : \Omega_i \to \mathbb{R}^+$, a

semi-dense inverse depth variance map $V_{i,\text{semi-dense}} : \Omega_i \to \mathbb{R}^+$, and a camera pose $S_i \in \text{Sim}(3)$. Note that $\Omega_i \subset \Omega$ is a subset of pixels extracted from the texture-rich image regions for the structure and camera motion estimation, and a Sim(3) camera pose S_i is defined by:

$$S_i = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}, \tag{1}$$

where $R \in SO(3)$ is the rotation matrix, $t \in \mathbb{R}^3$ the translation vector and $s \in \mathbb{R}^+$ the scaling factor.

B. Depth prediction

For every new keyframe $\mathcal{K}_i$, we obtain a relative inverse depth map, hereinafter referred to as relative depth map, from MiDaS [21] for the densification of the semi-dense depth map. Because the depth prediction $D_{i,\text{CNN}}$ is a relative depth map, the predicted depth map needs to be scale- and shift-corrected before it can be used in the densification step. The scale- and shift-correction can be performed as follows:

$$D'_{i,\text{CNN}} = a * D_{i,\text{CNN}} + b, \tag{2}$$

where $a \in \mathbb{R}^+$ and $b \in \mathbb{R}$ are the scale and shift parameters, respectively. Let $\vec{d_n} = \begin{pmatrix} d_n & 1 \end{pmatrix}^T$ and $h^{\text{opt}} = \begin{pmatrix} a & b \end{pmatrix}^T$, and the parameters a and b can be solved in closed-form as follows [21]:

$$h^{\text{opt}} = \left(\sum_{n \in \Omega_i} \vec{d_n} \vec{d_n}^T \right)^{-1} \left(\sum_{n \in \Omega_i} \vec{d_n} d'_n \right), \tag{3}$$

where $d_n \in D_{i,\text{semi-dense}}$ and $d'_n \in D_{i,\text{CNN}}$ are the inverse depth values of the semi-dense depth map and relative depth map, respectively.

C. Densification of the semi-dense structure

Consider the densification of $D_{i,\text{semi-dense}}$ of $\mathcal{K}_i$ using $D'_{i,\text{CNN}}$ and I_i: the estimated inverse dense depth map $D_{i,\text{opt}}$ can be obtained through the minimization of the cost function given by

$$E_{\text{total}} = E_{\text{CNN\_grad}} + \lambda_1 E_{\text{semi-dense}} + \lambda_2 E_{\text{photo\_grad}} + \lambda_3 E_{\text{CNN}}. \tag{4}$$

The first term, CNN depth gradient regularization $E_{\text{CNN\_grad}}$, enforces depth gradient consistency between $D_{i,\text{CNN}}$ and $D_{i,\text{opt}}$:

$$E_{\text{CNN\_grad}} = \frac{1}{|\Omega|} \sum_{n \in \Omega} \frac{(E_{\text{CNN\_grad},x}(n))^2 + (E_{\text{CNN\_grad},y}(n))^2}{\left(1/D'_{i,\text{CNN}}(n)\right)^2}, \tag{5}$$

with

$$\begin{aligned} E_{\text{CNN\_grad},x} &= \partial_x \ln D_{i,\text{opt}} - \partial_x \ln D'_{i,\text{CNN}} \\ E_{\text{CNN\_grad},y} &= \partial_y \ln D_{i,\text{opt}} - \partial_y \ln D'_{i,\text{CNN}}, \end{aligned} \tag{6}$$

where $|\Omega|$ is the cardinality of Ω, and ∂ the gradient operator. This error term is similar to the *scale-invariant mean squared error in log space* used in [25]. The denominator $\left(1/D'_{i,\text{CNN}}\right)^2$ in Equation (5) simulates the variance of the depth prediction, which provides stronger depth gradient regularization to closer objects than farther objects.

The second term, semi-dense depth consistency $E_{\text{semi-dense}}$, minimizes the difference between the optimized depth map and the semi-dense depth map from LSD-SLAM (similar to [11]):

$$E_{\text{semi-dense}} = \frac{1}{|\Omega_i|} \sum_{n \in \Omega_i} \rho \left(\frac{(D_{i,\text{opt}}(n) - D_{i,\text{semi-dense}}(n))^2}{V_{i,\text{semi-dense}}(n)} \right), \tag{7}$$

where $|\Omega_i|$ is the cardinality of Ω_i. We add the generalized Charbonnier penalty function [26], $\rho(.)$, to improve reconstruction accuracy.

The third term, image gradient regularization $E_{\text{photo\_grad}}$ (same as [16]), promotes depth smoothness in texture-poor image regions by matching the image gradient ∂I with the depth gradient ∂D_{opt}:

$$\begin{aligned} E_{\text{photo\_grad}} = \frac{1}{|\Omega|} \sum_{n \in \Omega} \big(&|\partial_x D_{\text{i, opt}}(n)| e^{-|\partial_x I_i(n)|} + \\ &|\partial_y D_{\text{i, opt}}(n)| e^{-|\partial_y I_i(n)|} \big). \end{aligned} \tag{8}$$

The last term, CNN depth consistency E_{CNN} (similar to [11]), provides an unary constraint over the estimated depth map:

$$E_{\text{CNN}} = \frac{1}{|\Omega|} \sum_{n \in \Omega} \frac{\left(\ln D_{i,\text{opt}}(n) - \ln D'_{i,\text{CNN}}(n)\right)^2}{\left(1/D'_{i,\text{CNN}}(n)\right)^2}. \tag{9}$$

IV. IMPLEMENTATION

Our dense SLAM pipeline is implemented using PyTorch [27] Multiprocessing[1], which allows for parallel processing of the depth prediction module and the dense mapping module.

For the energy minimization, we use PyTorch Autograd [28] with Adam optimizer [29], where the learning rate is set to 0.05. To compute the cost function, we set the weighting of different error terms to $\lambda_1 = 0.003$, $\lambda_2 = 0.05$ and $\lambda_3 = 0.001$, and the generalized Charbonnier function [26] parameters are set to $\epsilon = 0.001$ and $\alpha = 0.45$. The number of optimization iterations is set to 30. The

images have been resized to 320×240 before the depth prediction and densification steps.

In LSD-SLAM, we use the original parameter settings with the exception of setting the `minUseGrad` parameter to 1 (instead of 5) for the following sequences: ICL/office0, ICL/living1, and TUM/seq2 (see Table II). The decrease of the parameter allows more image points to be extracted for camera tracking, which helps prevent camera tracking lost in the aforementioned sequences. The frame-rate of all image sequences is set to 5 to allow for better synchronization between the camera tracking and the visualization of the dense map; the increase in frame-rate theoretically should not affect the dense reconstruction performance except for the delayed visualization of the dense map, thanks to the Multiprocessing implementation.

V. EVALUATION

In this section, we present experimental results that validate the effectiveness of our proposed method, namely (1) the cost function in our optimization framework, and (2) the use of relative depth prediction for providing depth and depth gradient priors.

A. Reconstruction accuracy

To evaluate our system, we use ICL-NUIM [9] and TUM RGB-D [8] datasets, which contain ground truth depth maps and trajectories to measure the reconstruction accuracy. We use the reconstruction accuracy metric proposed in [10], which is defined as the percentage of the depth values with relative error less than 10%. Since we do not interfere with the LSD-SLAM camera tracking, readers can expect similar from our system, if not identical, absolute trajectory errors (ATEs) produced from LSD-SLAM [4]. Also, our system does not produce absolute scale scene reconstruction, and therefore each depth map needs to be scaled using the optimal trajectory scale (calculated with the TUM benchmark script[2]) and its corresponding Sim(3) scale for depth correctness evaluation.

We compare our reconstruction accuracy against the state-of-the-art dense SLAM systems, namely CNN-SLAM [10], DeepFusion [11], and DeepFactors [12].[3] Table II shows a comparison of the reconstruction accuracy: the first three columns show the reconstruction accuracy of the state-of-the-art systems and the last two columns show a comparison between using VNLNet (an absolute depth prediction CNN) and MiDaS (a relative depth prediction CNN) in our optimization framework (see Section V-D). Owing to the similarity of the optimization frameworks between our system and DeepFusion (see Table I), we include the dense reconstruction results by running our optimization without the image gradient consistency error term, dubbed simulated

TABLE II: Comparison of overall reconstruction accuracy on the ICL-NUIM dataset [9] and the TUM RGB-D dataset [8]. (TUM/seq1: fr3_long_office_household, TUM/seq2: fr3_nostructure_texture_near_withloop, TUM/seq3: fr3_structure_texture_far, Sim. DeepFusion: Simulated DeepFusion.)

Sequence	Percentage of correct depth (%)					
	CNN-SLAM	DeepFactors*	DeepFusion	Sim. DeepFusion(MiDaS)*	Ours (VNLNet)*	Ours (MiDaS)*
ICL/office0	19.410	**30.17**	21.090	13.008	13.934	16.574
ICL/office1	29.150	20.16	37.420	52.860	**55.816**	51.758
ICL/office2	37.226	-	30.180	**69.382**	63.425	65.925
ICL/living0	12.840	20.44	24.223	**63.171**	53.328	62.085
ICL/living1	13.038	20.86	14.001	75.479	65.435	**78.226**
ICL/living2	26.560	-	25.235	71.631	68.454	**73.313**
TUM/seq1	12.477	29.33	8.069	62.872	60.356	**65.722**
TUM/seq2	24.077	16.92	14.774	48.548	41.236	**58.090**
TUM/seq3	27.396	51.85	27.200	74.613	70.690	**74.812**
Average	22.464	27.10	22.466	59.063	54.742	**60.723**

*After aligned with ground truth scale

TABLE III: Effect of the error terms on the reconstruction accuracy. The last two rows show the simulated DeepFusion [11] cost functions.

Energy term	Percentage of correct depth (%)		
	ICL/living2	ICL/office2	TUM/seq1
1	62.620	57.563	55.031
1(c)	65.611	57.644	55.042
1(c)+2	64.016	61.853	57.689
1(c)+2+3	72.594	65.098	64.820
1(c)+2+3+4	72.508	64.853	64.745
1(c)+2+3+5	67.285	61.596	63.678
1(c)+2+3+6	**72.609**	65.090	**64.822**
1(c)+3	69.967	**69.905**	64.650
1(c)+3+6	70.167	69.863	64.730

1. SLAM depth consistency
2. Image gradient consistency
3. CNN gradient consistency
4. Photometric error
5. Two-view depth map consistency
6. CNN depth consistency
(c). Generalized Charbonnier function

DeepFusion[4]. Note that the reconstruction accuracy of our method is taken with an average of 5 runs to take into account the non-deterministic nature of multi-threading in LSD-SLAM. Our method outperforms the competitors except for the ICL/office0 sequence, as LSD-SLAM is unable to generate a good semi-dense structure under rotational motion, hence the degraded reconstruction performance in the densification of the semi-dense structure. The reconstruction results demonstrate the superiority of our system by comparing the last column with all other columns in Table II. To understand the performance difference between our system and the rest of the systems, we identify the significance of each error term in the cost function being used in our optimization framework.

B. Cost function analysis

Table III shows the reconstruction results using different combinations of error terms in the cost function. The

last row is the reconstruction accuracy using a simulated DeepFusion cost function. To ensure consistent measurement of the reconstruction accuracy using different cost functions, the keyframes—i.e., the semi-dense depth and depth variance maps, and the camera poses—are pre-saved so that the densification process is not influenced by the non-deterministic nature from LSD-SLAM. Consistent with the finding in DeepFusion, incorporation of CNN depth gradient consistency improves the reconstruction accuracy dramatically, although our CNN does not explicitly predict depth gradient and depth gradient variance maps. By adding the image gradient prior (the third last row in Table III), our dense reconstruction is more accurate than the simulated DeepFusion cost function (see also the third last column and last column of Table II). Figure 3 shows the use of our cost function to obtain more accurate densified depth maps from less accurate relative depth maps. However, contrary to the findings in CNN-SLAM and DeepFactors, the inclusion of two-view depth map consistency and photometric error terms does not improve the reconstruction accuracy. One can see that having only the SLAM depth consistency in the cost function (the first row in Table III) is already performing better than the state-of-the-art systems (compare with Table II), which leads us to the following question: Can our system still perform well with a sparse structure[5]?

### C. Effect of map density on reconstruction accuracy

To measure the impact of SLAM map density on the reconstruction accuracy, we artificially reduce the LSD-SLAM structure density and measure the correctness of the keyframe depth maps. To this end, we perform densification on 1000, 2000, 4000, 6000, 8000, 10000, 15000 and 20000 randomly sampled depth values from the LSD-SLAM semi-dense depth maps, and then measure the correctness of the densified depth maps. We found that the reconstruction errors increase noticeably when the number of points being used in

---

[4]Despite the similarity in the cost function, there are two main differences between our simulated DeepFusion and the original implementation [11]: (1) their CNN explicitly predicts depth, depth uncertainty, depth gradient and depth gradient uncertainty maps, and (2) they use Opt [30] to perform energy minimization.

[5]Depending on the amount of visual texture in an image, LSD-SLAM can provide about $12k$ to $30k$ semi-dense structure points (after resizing the images to 320×240) for the densification, while a typical sparse SLAM algorithm generates about $1k$ to $2k$ sparse structure points [2], [3].
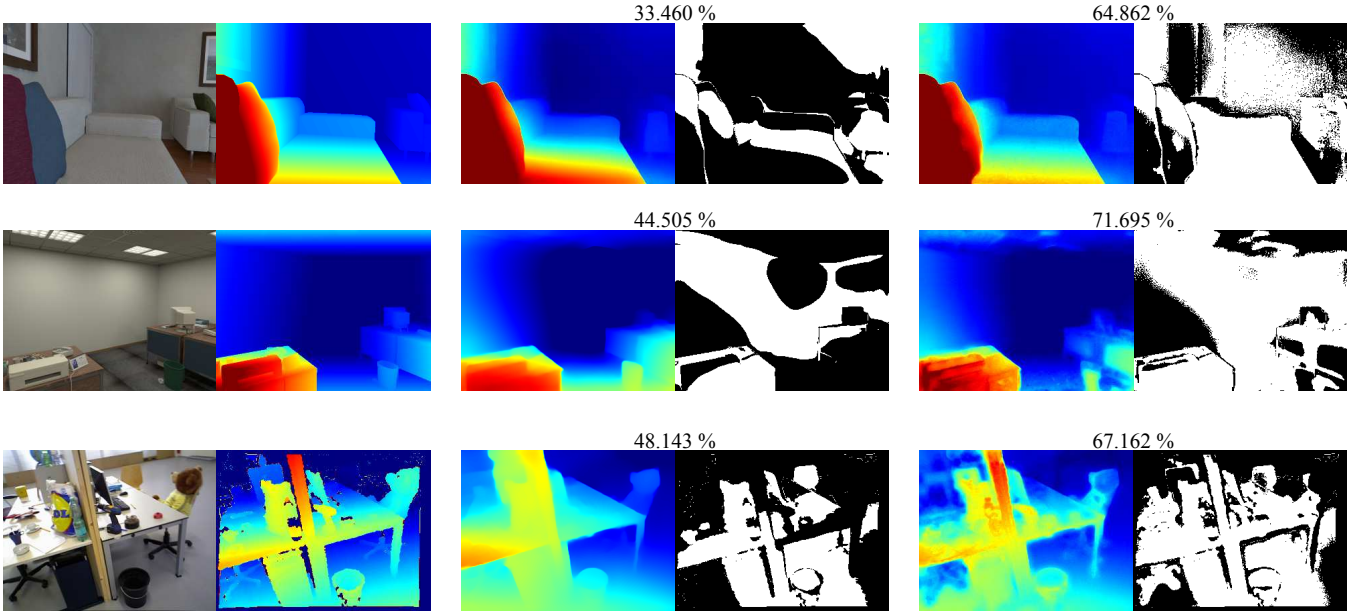
Fig. 3: Demonstration of the effectiveness of our optimization framework by comparing the relative depth prediction accuracy from MiDaS before the densification with the densified depth map. (Left column) image and ground truth depth map. (Middle column) scale- and shift-corrected relative depth map and depth correctness mask. (Right column) densified depth map and depth correctness mask. The precentage of correct depth of the depth correctness mask is shown above.

(a)   (b)
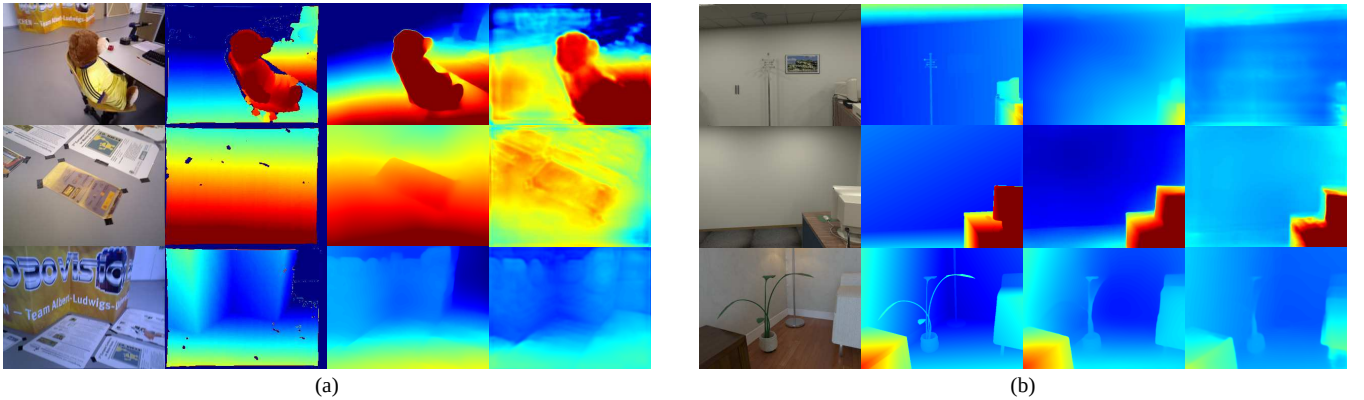
Fig. 4: Qualitative comparison of relative depth maps from MiDaS and absolute depth maps from VNLNet on the (a) TUM RGB-D dataset and the (b) ICL-NUIM dataset. From left to right: image, ground truth depth map, depth prediction from MiDaS, and depth prediction from VNLNet. Both relative and absolute depth maps have been scale- and shift-corrected using their corresponding semi-dense depth maps from LSD-SLAM. Also, the depth maps have been aligned with their ground truth scale before applying the heat map in order to reflect the same depth range.

the densification drops below 10000[6], thereby demonstrating the advantage of using LSD-SLAM, instead of a sparse SLAM algorithm [2], [3], for dense reconstruction. Additionally, the densified depth maps using a sparse structure also appear to be less smooth.

D. Relative depth prediction vs. absolute depth prediction

One key factor for accurate dense reconstruction is the meaningfulness of the priors—in our case, the depth and the depth gradient—being provided to the densification process. To illustrate the advantage of using relative depth prediction CNNs (e.g., MiDaS) for providing meaningful priors,

we perform the same densification step with an absolute depth prediction CNN, VNLNet[7] [18], and then compare the reconstruction accuracy between them. To promote a fair comparison, neither MiDaS nor VNLNet has been trained on the TUM RGB-D and ICL-NUIM datasets. In Table II, we show that, in general, using scale- and shift-corrected relative depth prediction (labelled with MiDaS) instead of absolute depth prediction (other columns) has superior dense reconstruction performance, as a result of more accurate depth prediction from MiDaS (last column of Table IV)

---

[6]Quantitatively, using 10000 depth values retains about 98% of the overall reconstruction accuracy while using 2000 depth values retains about 93%.

[7]One important consideration in selecting a competing absolute depth prediction CNN is the runtime memory requirements. VNLNet is considered state-of-the-art at the time of experimental setup with a reasonable memory footprint.

TABLE IV: Comparison of depth prediction CNNs accuracy being used in CNN-SLAM (Laina [31]) and our system (VNLNet [18] and MiDaS [21]) on the ICL-NUIM dataset [9] and the TUM RGB-D dataset [8]. (TUM/seq1: fr3_long_office_household, TUM/seq2: fr3_nostructure_texture_near_withloop, TUM/seq3: fr3_structure_texture_far, abs: absolute depth prediction CNN, rel: relative depth prediction CNN.)

	Percentage of correct depth (%)		
Sequence	Laina (abs)	VNLNet (abs)*	MiDaS (rel)*
ICL/office0	**17.194**	11.791	13.059
ICL/office1	20.838	**45.866**	42.980
ICL/office2	30.639	**55.180**	55.136
ICL/living0	15.008	40.294	**54.287**
ICL/living1	11.449	55.806	**72.139**
ICL/living2	33.010	59.367	**67.130**
TUM/seq1	12.982	47.552	**54.860**
TUM/seq2	15.412	33.143	**55.136**
TUM/seq3	9.450	52.144	**57.255**
Average	18.452	44.571	**52.442**

*After scale- and shift-correction

Fig. 5: Dense reconstruction from two outdoor sequences: (top) Oxford Robotcar 2014-06-22-15 [33] and (bottom) TUM MonoVO Seq29 [34].

than depth prediction from VNLNet (second last column of Table IV); Laina (second column of Table IV), another absolute depth prediction CNN being used in CNN-SLAM, is significantly less accurate than MiDaS, which indicates that the outperformance of our system may just simply be due to the fact that MiDaS provides more accurate depth prediction for the densification. The depth prediction CNN used by DeepFusion is not available for accuracy evaluation, but, given the similar reconstruction accuracy between CNN-SLAM and DeepFusion, we assume that the depth prediction CNN performance is similar to Laina. Not only are the scale- and shift-corrected relative depth maps from MiDaS metrically more accurate than the absolute depth maps from VNLNet, but the relative depth maps also appear to be smoother (see Figure 4).

Another advantage of using relative depth prediction CNNs is the generalizability to different testing datasets [20], [21], [32]. To illustrate the generalizability, Figure 5 shows dense reconstructions on two outdoor sequences, hinting at the potential use of relative depth prediction CNNs in dense reconstruction problems in diverse scene types.

E. Timing evaluation

On average, the CNN depth prediction and optimization require 0.15 s and 0.35 s, respectively, to complete. The measurements are taken on a laptop computer equipped with an Intel 7820HK CPU and an Nvidia GTX 1070 GPU.

VI. DISCUSSIONS

This study illustrates the potential capability of combining a relative depth prediction CNN with a visual SLAM algorithm in solving the dense monocular reconstruction problem. One of the major bottlenecks of the state-of-the-art dense SLAM systems is the accurate depth prediction requirement in the testing scene. While the use of absolute depth prediction may help produce absolute scale reconstruction, it mostly makes sense in the context narrow application domain, such as dense scene reconstruction for self-driving cars. With the proposed use of relative depth prediction, we improve the versatility of our system by forgoing absolute scale reconstruction, which can be easily recovered using fiducial markers or objects with known scales. With accurate relative depth prediction as well as continuous expansion in single-image relative depth CNN training datasets, we are getting closer to solving dense monocular SLAM *in the wild*—dense scene reconstruction on arbitrary image sequences.

VII. CONCLUSIONS

In this paper, we have presented a real-time dense SLAM system, named DeepRelativeFusion, that leverages the depth and depth gradient priors provided by a relative depth prediction CNN. Our system densifies the semi-dense structure provided by LSD-SLAM through a GPU-based energy minimization framework. Experimental results show that our system significantly outperforms the state-of-the-art dense SLAM systems and the performance gap can be attributed to (1) the formulation of a cost function that effectively combines the LSD-SLAM semi-dense structure, image gradient, CNN depth and CNN depth gradients, and (2) the meaningfulness of depth and depth gradient from relative depth maps as priors. We have validated the effectiveness of the cost function through the ablation studies, which examines the contribution of the error terms to the dense reconstruction accuracy. We have also demonstrated, quantitatively and qualitatively, that the relative depth prediction CNN— MiDaS—provide sufficiently accurate depth maps for dense scene reconstruction. With the accurate relative depth prediction on diverse scene types, the use of a relative depth prediction CNN is a promising step toward dense scene reconstruction in unconstrained environments.

However, our system is not without its flaws. Since our system is built upon LSD-SLAM, it automatically inherits the weaknesses of LSD-SLAM. These include wide baseline camera tracking, photometric constancy assumption and rotation motion to name a few. To further improve the dense reconstruction accuracy, we plan to improve the robustness of LSD-SLAM or another visual SLAM algorithm.

Robustification of a visual SLAM algorithm can be realized through the tight integration of accurate CNN predictions into the structure and motion optimization. Specifically, the integration of pose, depth and uncertainty predictions into front-end camera tracking and back-end bundle adjustment have been proposed [35], [36]. In the future, we will look into the ways to integrate relative depth prediction information into the optimization. Another possible extension is to use the optimized keyframe depth maps for camera motion refinement.

REFERENCES

[1] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'14)*, (Hong Kong, China), pp. 15–22, IEEE, May 2014.

[2] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2018.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] J. Engel, T. Schps, and D. Cremers, "LSD-SLAM: Large-scale Direct Monocular SLAM," in *Proc. European Conference on Computer Vision (ECCV'14)*, (Zurich, Switzerland), pp. 834–849, Springer, Sept. 2014.

[5] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*, pp. 2320–2327, IEEE, 2011.

[6] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1498–1505, IEEE, 2010.

[7] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Joint Pattern Recognition Symposium*, pp. 11–20, Springer, 2010.

[8] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[9] 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.

[10] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6243–6252, 2017.

[11] T. Laidlow, J. Czarnowski, and S. Leutenegger, "Deepfusion: Real-time dense 3d reconstruction for monocular slam using single-view depth and gradient predictions," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4068–4074, IEEE, 2019.

[12] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "Deepfactors: Real-time probabilistic dense monocular slam," *arXiv preprint arXiv:2001.05049*, 2020.

[13] J. Tang, J. Folkesson, and P. Jensfelt, "Sparse2dense: From direct sparse odometry to dense 3-d reconstruction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 530–537, 2019.

[14] C. S. Weerasekera, Y. Latif, R. Garg, and I. Reid, "Dense monocular reconstruction using surface normals," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2524–2531, IEEE, 2017.

[15] A. J. Amiri, S. Y. Loo, and H. Zhang, "Semi-supervised monocular depth estimation with left-right consistency using deep neural network," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 602–607, 2019.

[16] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, (Honolulu, Hawaii), IEEE, July 2017.

[17] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 283–291, 2018.

[18] W. Yin, Y. Liu, C. Shen, and Y. Yan, "Enforcing geometric constraints of virtual normal for depth prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5684–5693, 2019.

[19] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman, "Learning ordinal relationships for mid-level vision," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 388–396, 2015.

[20] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Advances in neural information processing systems*, pp. 730–738, 2016.

[21] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *arXiv:1907.01341*, 2019.

[22] W. Yin, X. Wang, C. Shen, Y. Liu, Z. Tian, S. Xu, and C. Sun, "Diversedepth: Affine-invariant depth prediction using diverse data," in *arxiv: 2002.00569*, 2020.

[23] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, "Neural rgb (r) d sensing: Depth and uncertainty from a video camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10986–10995, 2019.

[24] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Advances in Neural Information Processing Systems*, pp. 35–45, 2019.

[25] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, pp. 2366–2374, 2014.

[26] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

[27] 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 *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NeurIPS Workshop*, 2017.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] Z. DeVito, M. Mara, M. Zollöfer, G. Bernstein, C. Theobalt, P. Hanrahan, M. Fisher, and M. Nießner, "Opt: A domain specific language for non-linear least squares optimization in graphics and imaging," *ACM Transactions on Graphics 2017 (TOG)*, 2017.

[31] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*, pp. 239–248, IEEE, 2016.

[32] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, 2018.

[33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[34] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," in *arXiv:1607.02555*, July 2016.

[35] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," *arXiv preprint arXiv:2003.01060*, 2020.

[36] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang, "Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5218–5223, IEEE, 2019.