ACCEPTED VERSION

Chamara Saroj Weerasekera, Yasir Latif, Ravi Garg, Ian Reid
Dense monocular reconstruction using surface normals
2017 IEEE International Conference on Robotics and Automation (ICRA), 2017 / pp.2524-2531

Copyright © 2017 IEEE.

Published version at: http://dx.doi.org/10.1109/ICRA.2017.7989293

PERMISSIONS

https://www.ieee.org/publications/rights/author-posting-policy.html

Author Posting of IEEE Copyrighted Papers Online

The IEEE Publication Services & Products Board (PSPB) last revised its Operations Manual Section 8.1.9 on Electronic Information Dissemination (known familiarly as "author posting policy") on 7 December 2012.

PSPB accepted the recommendations of an ad hoc committee, which reviewed the policy that had previously been revised in November 2010. The highlights of the current policy are as follows:

- The policy reaffirms the principle that authors are free to post their own version of their IEEE periodical or conference articles on their personal Web sites, those of their employers, or their funding agencies for the purpose of meeting public availability requirements prescribed by their funding agencies. Authors may post their version of an article as accepted for publication in an IEEE periodical or conference proceedings. Posting of the final PDF, as published by IEEE *Xplore*®, continues to be prohibited, except for open-access journal articles supported by payment of an article processing charge (APC), whose authors may freely post the final version.
- The policy provides that IEEE periodicals will make available to each author a preprint version of that person's article that includes the Digital Object Identifier, IEEE's copyright notice, and a notice showing the article has been accepted for publication.
- The policy states that authors are allowed to post versions of their articles on approved third-party servers that are operated by not-for-profit organizations. Because IEEE policy provides that authors are free to follow public access mandates of government funding agencies, IEEE authors may follow requirements to deposit their accepted manuscripts in those government repositories.

IEEE distributes accepted versions of journal articles for author posting through the Author Gateway, now used by all journals produced by IEEE Publishing Operations. (Some journals use services from external vendors, and these journals are encouraged to adopt similar services for the convenience of authors.) Authors' versions distributed through the Author Gateway include a live link to articles in IEEE *Xplore*. Most conferences do not use the Author Gateway; authors of conference articles should feel free to post their own version of their articles as accepted for publication by an IEEE conference, with the addition of a copyright notice and a Digital Object Identifier to the version of record in IEEE *Xplore*.

28 April 2021

http://hdl.handle.net/2440/117918

Dense Monocular Reconstruction using Surface Normals

Chamara Saroj Weerasekera, Yasir Latif, Ravi Garg, Ian Reid

***Abstract*— This paper presents an efficient framework for dense 3D scene reconstruction using input from a moving monocular camera. Visual SLAM (Simultaneous Localisation and Mapping) approaches based solely on geometric methods have proven to be quite capable of accurately tracking the pose of a moving camera and simultaneously building a map of the environment in real-time. However, most of them suffer from the 3D map being too sparse for practical use. The missing points in the generated map correspond mainly to areas lacking texture in the input images, and dense mapping systems often rely on hand-crafted priors like piecewise-planarity or piecewise-smooth depth. These priors do not always provide the required level of scene understanding to accurately fill the map. On the other hand, Convolutional Neural Networks (CNNs) have had great success in extracting high-level information from images and regressing pixel-wise surface normals, semantics, and even depth. In this work we leverage this high-level scene context learned by a deep CNN in the form of a surface normal prior. We show, in particular, that using the surface normal prior leads to better reconstructions than the weaker smoothness prior.**

I. Introduction

The ability to carry out dense and accurate 3D mapping of the environment is desirable in applications such as autonomous navigation, robotic manipulation, augmented reality, etc. Doing so merely using input from a moving monocular/stereo camera is attractive as cameras are ubiquitous, compact, power efficient, and not limited by range. Visual SLAM (Simultaneous Localisation and Mapping) systems have evolved to the point where they are capable of many feats such as accurate and real-time camera tracking and 3D mapping [1], [2], [3], with some capable of fully dense live 3D reconstruction [4], [5].

Less attention has been paid to the application of high-level scene understanding to aid the mapping process. Some work has introduced constraints such as Manhattan world assumptions, or piecewise planar priors [6], [7], [8]. Other stronger priors have also been leveraged, such as known objects [9], [10], while other works have made use of smoothness assumptions to "fill in" regions where there is insufficient photometric variation [4], [11]. For example, [4] requires hand crafted priors on depth (piecewise constant disparity) to fill in the regions of low texture. High-level scene context (e.g. offices having a desk, on top of which a monitor, keyboard, etc. often lie in a specific configuration) is usually ignored in pure-geometry based SLAM systems.

In our work we recognise that recent advances in deep learning techniques mean that priors about surface orientation can be captured within a multi-layer Convolutional Neural Network (CNN) which regresses image patches to local surface normal values [12], [13]. The predictions are likely to be backed by the capacity of neural networks to grasp high-level concepts such as object type and scene layout, and relative orientation to one another, in addition to low-level cues such as shading [14]. The improvements they can bring to traditional reconstruction techniques are also demonstrated in recent work like [15]. Motivated by this, we here present the first framework for real-time monocular dense mapping that efficiently combines both the benefits of deep CNNs and well-established geometry-based methods.

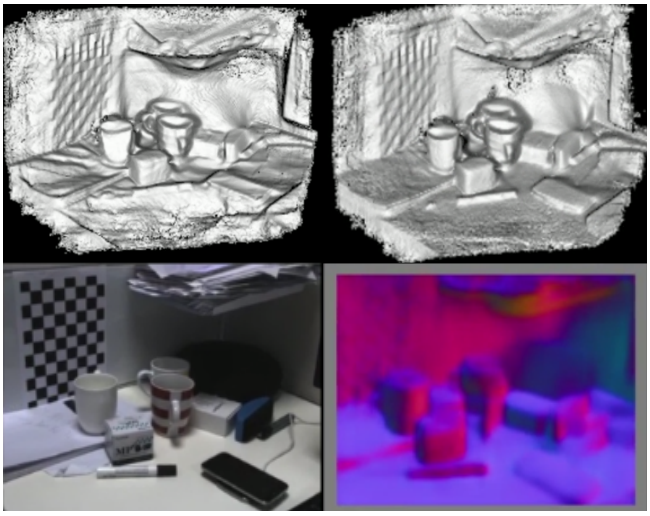

Fig. 1. Reconstruction using smoothness regularizer (top left) and normal prior (top right), an RGB image in the sequence (bottom left), and corresponding predicted normals (bottom right). Note the more accurate high-fidelity reconstruction obtained using the normal prior. A live comparison video is available at https://youtu.be/atq1EhX-75k.

In particular, we use normal predictions from a learned neural network [12] as a strong prior and aim to estimate a map which (i) minimizes photometric cost and (ii) is consistent with the single-view normal predictions. To that end we incorporate a depth/normal consistency term in our energy minimization framework which acts as a regularizer to fill in the gaps in the map with very little texture. We closely follow [4]'s energy minimization method for mapping where our proposed regularizer replaces the inverse-depth smoothness prior used in [4].

We extensively evaluate the proposed method quantitatively against the traditional hand-crafted smoothness regularizer such as [4], and pure learning-based system like [12] on a diverse range of indoor sequences in a large dataset like raw NYU-D V2 [16]. Our benchmarking highlights

All authors are with the School of Computer Science, at the University of Adelaide, and Australian Centre for Robotic Vision.

limitations of both pure learning-based and pure-geometry-based systems, which are addressed by the proposed method.

II. Background and Related Work

The problem of monocular SLAM is well studied and consists of estimating the structure of the environment together with the location of the camera at any given moment. The environment is generally represented as a collection of points, whose density varies from sparse [3] to semi-dense [2] to dense [4], [17]. Geometry-based methods rely on minimizing multi-view photometric error/feature-correspondences under the assumption that the environment is sufficiently textured. Therefore, in featureless regions they are unable to reason about the environment's geometry sufficiently well. This is particularly an issue for dense mapping. Several hand-crafted priors have thus been proposed to address this. Some of them include smoothness in disparity/depth priors [4], [11], [17] or Manhatton/piewise-planar priors [6], [18], [7], [19], [8]. Another line of work [20], [9], [21], [10], [22], [23] used a limited set of detected object classes as a prior for reconstruction.

Single-image surface normal estimation/prediction has been a topic of interest over the years. Early work includes estimating normals using shape from shading [14], and other low-level image features [24]. More recently, learning based approaches such as [25], [26], especially those using neural nets [12], [27] have achieved state-of-the art performance in normal prediction. In [12], an efficient multi-scale CNN was proposed for estimation of depth, surface normals, and semantic labels. In their neural network architecture, all three tasks shared common weights for the course scale, and the finer scales received additional input from the output of the courser scales. A separate line of work aim to reconstruct purely based on normal information [28].

More closely aligned with our work, in [29], a formulation to integrate normals with photometric consistency for reconstruction was proposed. The normals were computed based on local-planar patch fitting that maximized photo-consistency. In [30] surface normals from detected object classes were used to formulate a regularizer to penalize reconstruction errors. More recently, in [15], in a direction very similar to ours, a method was proposed for refining single-image depth predictions, or stereo-based depth estimates using surface normal predictions from a neural network. Pixel-wise classification scores of a discrete set of surface normals were used to construct a Wulff shape which served as a regularization function to penalize incorrect pairwise depth relationships. The experiments in [15] however were limited to improving stereo reconstruction and single-view depth predictions, and the formulation utilized discretized normal predictions.

III. Method

Our proposed framework incrementally generates a fully dense reconstruction of a scene from a video sequence given pixel-wise surface normal maps of keyframes and photometric evidence from a series of overlapping images corresponding to those keyframes. The following sub-sections elaborate on the key components.

A. Notations

The following notations and conventions will be used in this section. K is the camera intrinsic matrix. $\mathbf{I}_r \in \mathbb{R}^3$ is a $M \times N$ keyframe image and $\mathbf{I}_n \in \mathbb{R}^3$ is a $M \times N$ image in the set of images overlapping $\mathbf{I}_r$. We assume images are undistorted. $\mathbf{u}_p = (u, v)^T$ is a pixel location in $\mathbf{I}_r$, where $p = 1, ..., MN \in \mathcal{P}$ is a pixel location-based index. $\dot{\mathbf{u}}_p = (u, v, 1)^T$ is $\mathbf{u}_p$ in homogeneous form, and $\tilde{\mathbf{x}}_\mathbf{p} := K^{-1}\dot{\mathbf{u}}_p$. For a pixel $\mathbf{u}_p$, d_p and ρ_p are the corresponding depth and inverse-depth respectively, and $\mathbf{d}$ and $\boldsymbol{\rho}$ are $MN \times 1$ vectors of stacked d_p and ρ_p values respectively. $T_{nr} \in \mathbb{SE}(3)$ is a matrix describing the transformation of a point from camera coordinates of $\mathbf{I}_r$ to that of $\mathbf{I}_n$. $\pi(.)$ and $\pi^{-1}(.,.)$ are the projection and back-projection operations, such that $\pi(K^{-1}\dot{\mathbf{u}}_p/\rho_p) = \mathbf{u}_p$ and $\pi^{-1}(\mathbf{u}_p, \rho_p) = K^{-1}\dot{\mathbf{u}}_p/\rho_p$. The predicted surface normal vectors $\hat{\mathbf{n}}_p \in \mathbb{R}^3$ are normalized in Euclidean space and are in camera coordinates of $\mathbf{I}_r$.

B. Energy Formulation for Mapping

We formulate depth estimation of a keyframe as an energy minimization problem. Closely following [4] our energy function consists of a dataterm and a regularization term as follows and will be minimized with respect to $\boldsymbol{\rho}$:

$$E(\boldsymbol{\rho}) = \sum_{p \in \mathcal{P}} \frac{1}{\lambda} E_\phi(\rho_p) + E_{\hat{n}}(\rho_p), \tag{1}$$

where λ controls the regularization strength.

E_ϕ is the dataterm that computes the photometric error for a keyframe $\mathbf{I}_r$ accumulated over N overlapping frames:

$$E_\phi(\rho_p) = \frac{1}{N} \sum_{n=1}^{N} \left\| \mathbf{I}_r^{'}(\mathbf{u}_p) - \mathbf{I}_n^{'}(\pi(T_{nr}\pi^{-1}(\mathbf{u}_p, \rho_p))) \right\|_1 \tag{2}$$

For added robustness in the photometric matching, we concatenate the RGB channels of $\mathbf{I}_r$ and $\mathbf{I}_n$ with an additional image gradient-based channel, computed using eqn. (6), forming $\mathbf{I}_r^{'}$ and $\mathbf{I}_n^{'}$ respectively. Using eqn. (2) a cost volume [4] can be created that stores average photometric error for a discrete set of inverse depth labels $l \in \mathcal{L}$, for each pixel $\mathbf{u}_p$. Sections *D* and *E* contain more information pertaining to cost volume creation.

Our proposed regularization term is based on the relationship between a pair of 3D points and its corresponding normal (Fig. 2) in the camera coordinates of $\mathbf{I}_r$:

$$\langle \hat{\mathbf{n}}_p, d_q\tilde{\mathbf{x}}_q - d_p\tilde{\mathbf{x}}_p \rangle = 0, \tag{3}$$

where $\langle\ ,\ \rangle$ is the dot product operator and $q \in \mathcal{N}(p)$ corresponds to a pixel location in the neighbourhood of that of p. Equation (3) can be simplified as:

$$\begin{aligned} d_q\langle \hat{\mathbf{n}}_p, \tilde{\mathbf{x}}_q \rangle - d_p\langle \hat{\mathbf{n}}_p, \tilde{\mathbf{x}}_p \rangle &= 0 \\ d_q c_{pq} - d_p c_{pp} &= 0 \\ \rho_p c_{pq} - \rho_q c_{pp} &= 0 \end{aligned} \tag{4}$$

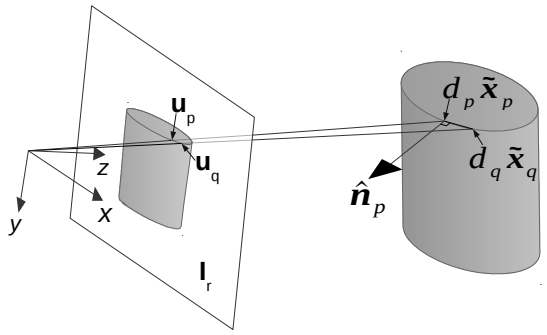

Fig. 2. A neighbouring 3D point pair and corresponding surface normal.

where c_{pq} and c_{pp} are constants equal to $\hat{\mathbf{n}}_p \cdot \tilde{\mathbf{x}}_q$ and $\hat{\mathbf{n}}_p \cdot \tilde{\mathbf{x}}_p$ respectively. Thus we can minimize the following energy, penalizing inconsistent inverse depth values:

$$E_{\hat{n}}(\rho_p) = g_p \left\| \boldsymbol{\nabla}_p^{\boldsymbol{\rho}} \right\|_\epsilon \tag{5}$$

where $\boldsymbol{\nabla}_p^{\boldsymbol{\rho}}$ denotes a vector of operations as follows:

$$\boldsymbol{\nabla}_p^{\boldsymbol{\rho}} = \begin{bmatrix} \rho_p c_{pi} - \rho_i c_{pp} \\ \rho_p c_{pj} - \rho_j c_{pp} \end{bmatrix}.$$

The indices i and j correspond to pixel locations neighbouring that of p in the *positive* x and y directions in the image plane. Note that we have restricted the neighbourhood pairwise connectivity to the latter for computational efficiency.

The image-edge based weight $g_p = g(\mathbf{I}_r, \mathbf{u}_p)$ where

$$g(\mathbf{I}, \mathbf{u}) = e^{-\alpha \|\nabla \mathbf{I}(\mathbf{u})\|_2^\beta} \tag{6}$$

reduces regularization at image edges, under the assumption that these regions align with depth discontinuities, and also have higher data-term quality. α and β are tunable parameters.

The Huber norm, defined as

$$\|\mathbf{x}\|_\epsilon = \begin{cases} \dfrac{\|\mathbf{x}\|_2^2}{2\epsilon} & \text{if } \|\mathbf{x}\|_2 \leq \epsilon \\ \|\mathbf{x}\|_1 - \dfrac{\epsilon}{2} & \text{otherwise} \end{cases}, \tag{7}$$

also minimises penalties at surface discontinuities, and makes the overall energy more robust to errors in normal predictions.

In the special case when $c_{pq} = c_{pp} = -1$ which occurs when $\hat{\mathbf{n}} = (0, 0, -1)^T$, i.e. the normal is pointed directly at the camera, eqn. (5) reduces to the smoothness prior used in [4]. Hence ours is a more general form that can enforce inverse depth relationships for arbitrary surface orientations visible to the camera. Adding more flexibility, we introduce a tunable parameter γ which balances smoothness/normals regularization:

$$\begin{aligned} c_{pq} &\Leftarrow (1-\gamma) c_{pq} - \gamma \\ c_{pp} &\Leftarrow (1-\gamma) c_{pp} - \gamma \end{aligned} \tag{8}$$

where γ is a value between 0 and 1. The parameter γ could also be a function of the normal prediction uncertainty estimated using a technique such as in [31].

C. Optimisation of Keyframe Inverse Depths

The objective can be written as follows:

$$\min_{\boldsymbol{\rho}} E(\boldsymbol{\rho}) = \sum_{p \in \mathcal{P}} \frac{1}{\lambda} E_\phi(\rho_p) + g_p \left\| \boldsymbol{\nabla}_p^{\boldsymbol{\rho}} \right\|_\epsilon \tag{9}$$

Based on the Legendre-Fenchel transform, the convex problem $\min_{\boldsymbol{\rho}} \sum_{p \in \mathcal{P}} g_p \left\| \boldsymbol{\nabla}_p^{\boldsymbol{\rho}} \right\|_\epsilon$ is equivalent to [32]:

$$\min_{\boldsymbol{\rho}} \max_{\mathbf{q}} \sum_{p \in \mathcal{P}} \left\{ \langle \boldsymbol{\nabla}_p^{\boldsymbol{\rho}}, \mathbf{q}_p \rangle - \delta_q(\frac{\mathbf{q}_p}{g_p}) - \frac{\epsilon}{2} \frac{\|\mathbf{q}_p\|_2^2}{g_p} \right\}, \tag{10}$$

where $\mathbf{q}_p = [q_{px}, q_{py}]^T$ is the dual variable and $\mathbf{q} = [\mathbf{q}_1, ..., \mathbf{q}_{MN}]^T$. $\delta_q(\mathbf{q}_p/g_p) = 0$ if $\|\mathbf{q}_p/g_p\|_2 \leq 1$ and ∞ otherwise.

Following [4] and [33], we introduce a linking term $\frac{1}{2\theta} \|\rho_p - a_p\|_2^2$ into (9) and replace $E_\phi(\rho_p)$ with $E_\phi(a_p)$, where a_p is an auxiliary variable. We define $\mathbf{a}$ to be a vector such that $\mathbf{a} = [a_1, ..., a_{MN}]^T$.

The objective (9) can now be written as:

$$\min_{\boldsymbol{\rho}, \mathbf{a}} \max_{\mathbf{q}} E(\boldsymbol{\rho}, \mathbf{a}, \mathbf{q}) \tag{11}$$

where

$$\begin{aligned} E(\boldsymbol{\rho}, \mathbf{a}, \mathbf{q}) = \sum_{p \in \mathcal{P}} & E_a(a_p, \rho_p) + \langle \boldsymbol{\nabla}_p^{\boldsymbol{\rho}}, \mathbf{q}_p \rangle \\ & - \delta_q(\frac{\mathbf{q}_p}{g_p}) - \frac{\epsilon}{2} \frac{\|\mathbf{q}_p\|_2^2}{g_p} \end{aligned} \tag{12}$$

and

$$E_a(a_p, \rho_p) = \frac{1}{\lambda} E_\phi(a_p) + \frac{1}{2\theta} \|\rho_p - a_p\|_2^2. \tag{13}$$

The energy in eqn. (12) can be optimised by performing gradient ascent on dual variables $\mathbf{q}_p, p \in \mathcal{P}$, followed by gradient descent on primal variables $\rho_p, p \in \mathcal{P}$, and an exhaustive point-wise search in the discrete label space $\mathcal{L}$ for finding a_p that minimizes $E_a(a_p, \rho_p), p \in \mathcal{P}$ [34][4]. This process is repeated iteratively while decreasing θ slowly. The variable updates within the primal and dual steps and the point-wise search can occur in parallel, and thus (12) can be optimized efficiently in GPU hardware [34]. Using a sparse pairwise graph structure for the regularization term adds to the gradient computation efficiency.

Using the method in [4] we compute the upper and lower bounds for the exhaustive search since the required search space in $\mathcal{L}$ grows narrower as θ decreases, and also perform a single Newton step on the optimal $\mathbf{a}$ at each time step for achieving sub-label accuracy (using numerical derivatives of $E_\mathbf{a}$ w.r.t $\mathbf{a}$ around its current discrete solution [4]). Algorithm 1 summarizes the steps for obtaining the solution.

In Algorithm 1, ∇_p^q denotes the following operation:

$$\nabla_p^q = (q_{px} c_{pi} - q_{rx} c_{rr}) + (q_{py} c_{pj} - q_{sy} c_{ss}), \tag{14}$$

where r and s correspond to pixel locations neighbouring that of p in the *negative* x and y directions in the image plane. Note that $\boldsymbol{\nabla}_p^{\boldsymbol{\rho}}$ and ∇_p^q can be considered a generalization of the gradient and divergence operations. The dual and primal step sizes are denoted by σ_q and σ_ρ respectively.

Algorithm 1: Optimisation procedure for solving for optimal inverse depth values $\rho_p = a_p, p \in \mathcal{P}$ for a keyframe $\mathbf{I}_r$

Initialize $q_p = 0, p \in \mathcal{P}$;
Initialize $a_p = \rho_p = \arg\min_{a_p \in \mathcal{L}} E_\phi(a_p), p \in \mathcal{P}$;
Initialize $\theta = \theta_{start}$;
Compute g_p, c_{pq}, and $c_{pp}, p \in \mathcal{P}, q \in \mathcal{N}(p)$;
repeat
  $\mathbf{q}_p \Leftarrow (\mathbf{q}_p + \sigma_q \boldsymbol{\nabla}_p^\rho)/(g_p + \sigma_q \epsilon), p \in \mathcal{P}$;
  $\mathbf{q}_p \Leftarrow g_p \mathbf{q}_p / \max(1, \|\mathbf{q}_p\|_2), p \in \mathcal{P}$;
  $\rho_p \Leftarrow (\rho_p + \sigma_\rho(-\nabla_p^q + \frac{1}{\theta} a_p))/(1 + \frac{\sigma_\rho}{\theta}), p \in \mathcal{P}$;
  Compute bounds for point-wise search [4] ;
  $a_p \Leftarrow \arg\min_{a_p \in \mathcal{L}} E_a(a_p, \rho_p), p \in \mathcal{P}$;
  Do Newton-step on a_p (if step-size $<$ bin size) [4] ;
  Decrease θ ;
until *convergence*;

D. Camera Tracking and Frame selection

We use the framework in [3] for accurate feature-based camera tracking, and providing the required transformation T_{nr} for computing the photometric cost. The choice of $\mathbf{I}_r$ and its overlapping frame set greatly influence the reliability of the data term. We consider a pre-defined frame window of size $N = N_p + N_f$ around $\mathbf{I}_r$, where N_p is the maximum number of past overlapping *keyframes* (large-baseline) to consider, and N_f is the maximum number of future overlapping *frames* (small-baseline) to consider. The past keyframes are selected with the help of [3]'s covisibility graph, and they are stored in a fixed length rolling buffer in GPU memory so that they need not be re-copied. Setting the future frame count to 0 allows for "just-in-time" reconstruction as demonstrated in the video accompanying Fig. 1, while increasing it correspondingly increases the mapping latency. Each keyframe change in the tracker [3] sets a flag that indicates sufficient motion to initialize a new $\mathbf{I}_r$. Once the flag is set, and the previous keyframe's cost volume update and inverse depth optimisation is complete, the current image in the sequence is set as $\mathbf{I}_r$. Fig 3 illustrates the data flow and steps undertaken during a single keyframe reconstruction.

E. Scaling Camera Translations

As an optional step we scale camera translations to a fixed scale to facilitate the choice of a consistent set of inverse depth labels $\mathcal{L}$. This is due to the inherent scale ambiguity in monocular SLAM which may require the inverse depth label range to be manually tuned every time the system is re-run. For automatically recovering the approximate scale we use absolute depth predictions from [12] which are predicted in parallel with surface normals and bear minimal overhead to prediction time (sub-section F). These depth predictions are able to provide a rough idea about the scene's scale, having learnt approximately the relationship between object features and their typical size. Note that we use depth predictions *only* for scale recovery and hence up-to-scale reconstructions are possible without it.

We do a 1-point RANSAC based least-square fit to find the approximate multiplicative scale factor that will align [3]'s sparse 3D map with the corresponding CNN depth predictions for the current keyframe. This scale is then used to normalize the camera translations in T_{nr}, which in turn enables the use of a fixed set of inverse depth labels (based on metric units) for the cost volume. We perform a running average of the scale factors recovered for each $\mathbf{I}_r$ to improve the reliability of the scale factor estimate.

F. Surface Normals Prediction

For regressing surface normals directly from the keyframe image, we utilize the multi-scale CNN model proposed in [12]. We use their VGG model variant with VGG-16-based convolutional layers in scale 1 followed by 2 fully connected layers. The input RGB image to the network is first resized to 320x240 and then centre cropped to 304x228. The cropping is required as the network was trained with randomly cropped images at that same crop resolution. Scale 1 mainly operates on a courser image resolution and extracts more global features. Scales 2 and 3 consist of fully convolutional layers which operate on fine and finer image resolutions respectively and extract more local features. Scales 2 and 3 receive upsampled output from the preceding courser scales [12].

The network at scale 3 simultaneously regresses a surface normal map and depth map for the input keyframe image at 147x109 resolution. In spite of the low-resolution output a major portion of the scene detail is still captured. We bilinearly upsample the predictions by a factor of 2 to the corresponding region in the 320x240 input image. The small amount of missing information at the border of the resulting normal/depth map is due to effects of cropping at the input and intermediate layers in the network. We do not perform inverse depth regularization in this border region.

We replicated [12]'s model in the efficient Caffe [36] framework and transferred the learnt weights. Their model has been trained on millions of indoor images (data augmentation included) in the training set of the raw NYU-Depth V2 dataset [16]. The combined prediction time for a surface normal map and depth map in Caffe is $\approx 40ms$ in GPU mode.

G. Volumetric Fusion

As a post-processing step, the depth maps resulting from the optimisation are fused into a global volumetric model based on truncated signed distance function, using the open-source InfiniTAM system [37]. The overall framework is summarized in Fig. 3.

IV. Evaluation

Using the smoothness regularizer ($\gamma = 1$ in eqn. (8)) as the baseline, we explore the improvements after using the surface normals regularizer ($\gamma = 0$) on a large number of sequences in several video datasets. All our experiments are conducted on a standard desktop PC with an Intel i7 4790 CPU and a

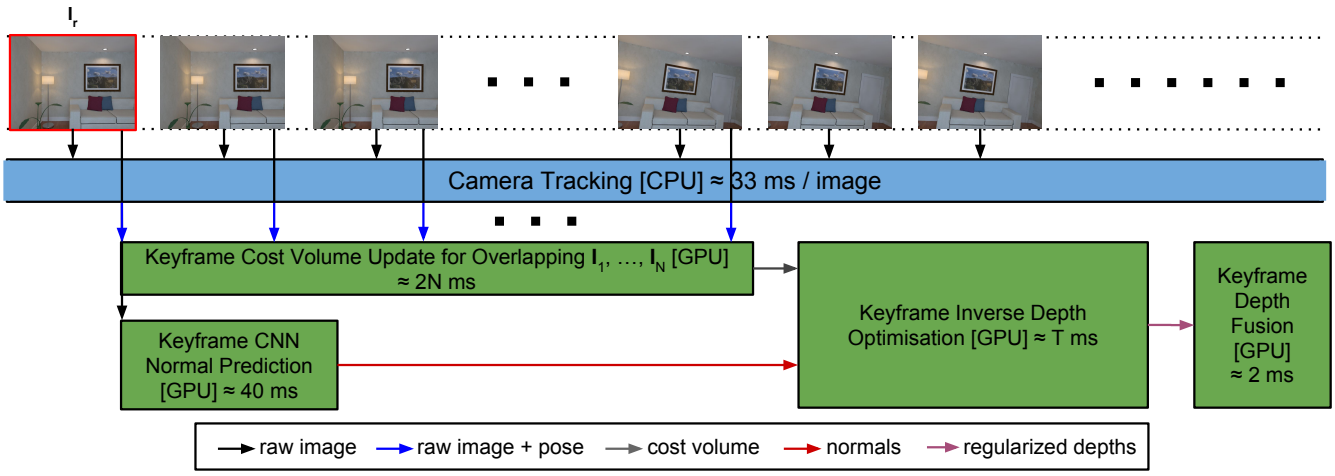

Fig. 3. Key components of the framework and data flow. The depicted steps are repeated for each keyframe $\mathbf{I}_r$. Our GPU implementation is based on CUDA [35]. Camera tracking [3] runs in the main CPU thread. Cost volume update and CNN normal prediction both run on GPU on two independent CUDA streams, and are managed in parallel with the main thread. Optimisation and depth map fusion run subsequently on GPU and are also managed in parallel with the main thread. Similar to [4] the optimisation time T is dependent on the optimisation parameters like step sizes and θ scheduling policy, and is $\approx$ 50 ms/keyframe for fast but less accurate reconstructions, and $\approx$ 800 ms/keyframe for more accurate but higher latency reconstructions. Mapping latency (time between successive keyframe reconstructions) is $\approx (33 + max(40, max(33N_f, 2(N-1)) + 2) + T + 2)$ ms where $N = N_p + N_f$ is the total number of images overlapping $\mathbf{I}_r$. If the number of future frames N_f is set to 0, minimum mapping latency for reasonable accuracy is $\approx$ 150 ms in practice. Frequency of keyframe reconstruction also depends on amount of camera translation which determine when a keyframe change should occur.

Nvidia GTX 980 4GB GPU. We also compare against single-view CNN depth predictions from [12] to observe how well a pure-learning based approach compares with our combined learning and photometric error-based approach.

In our experiments we bilinearly downsample input images to 320x240 to ensure low-latency fusions and smooth operation for the entire pipeline. This is at a slight compromise of data-term quality. The neural network also takes in input at this resolution which made the choice more appropriate. Given hardware speed and memory constraints, the reduced resolution also allowed the use of a relatively large label set $\mathcal{L}$ for the cost volume. We use a fixed inverse depth range of 0 to 4 with 256 bins, sufficient for reconstructing small to large scale environments. Note that no finetuning of the network is done on any of the sequences used for evaluation. This allows us to test the generalization capabilities of the neural network.

We performed quantitative comparisons using the RGB-D tracking feature in [3] as (i) it allowed for ease of repeatability of experiments (more consistent camera pose estimates and keyframes selected, no scale ambiguity) and (ii) it leaves out errors in camera poses when comparing the two regularization methods. The qualitative results were generated using regular monocular tracking.

We first tune λ for the two regularizer types using test scenes in the raw NYU-D V2 dataset, so that the optimal λ can be chosen. Following [12] we split the raw NYU-Depth dataset [16] into test and train scenes based on the official dataset split, using scenes that don't contain train images for testing. This gives 247 different indoor test sequences out of which we choose 25 by sampling every tenth test sequence to roughly correspond to all the different types of indoor scene categories present in the raw test set [16].

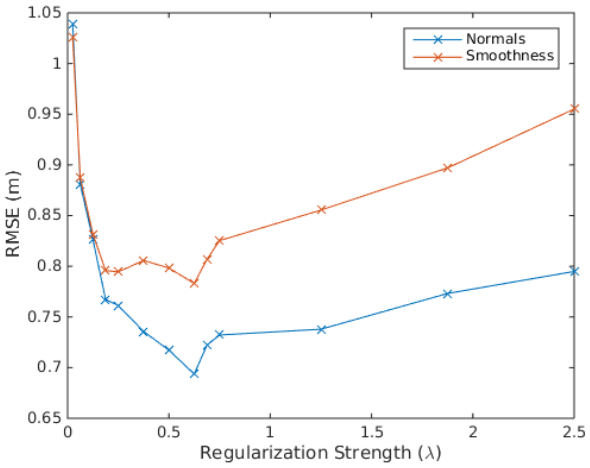

Fig. 4. Average keyframe RMS reconstruction error (m) w.r.t regularization strength (λ) on the raw NYU-D V2 test sequences. Notice that the error for the normal-based regularizer remain lower.

The plot of RMS error vs λ for the two regularizers are provided in Fig. 4. It can be seen that as we vary λ, the error compared to the ground-truth remains lower for the normals-based regularization. This is because even though the regularization strength is high, the normals guide the depths to the right solution more accurately, while with the smoothness regularizer, a higher strength causes piecewise planar reconstructions that are fronto-parallel to the keyframe image plane. This effect is more apparent in areas where there is little texture as expected.

Quantitative results on these sequences for the following error and accuracy measures are given in Table I. Note that d_p and d_p^{gt} denote regularized depth and groundtruth

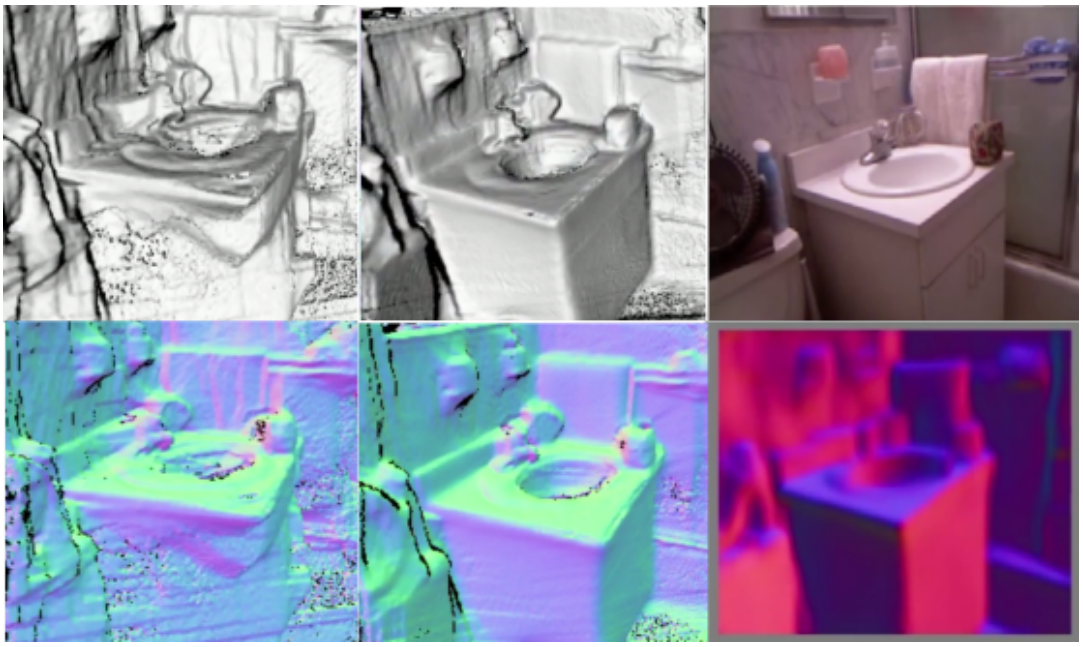

Fig. 5. Qualitative results on NYU raw dataset 'bathroom_0003' test sequence. Phong shaded fused reconstruction using smoothness prior (top left), phong shaded fused reconstruction using normals prior (top middle), a rgb keyframe image in the sequence (top right), surface normal rendering of fused smoothness-prior reconstruction (bottom left), surface normal rendering of fused normal-prior reconstruction (bottom middle), corresponding normal predictions for rgb keyframe image (bottom right). Note the more accurate reconstruction of textureless regions like the inside of the round sink using the normal-prior. A live comparison video is available at https://youtu.be/BRLN-1MTZtw.

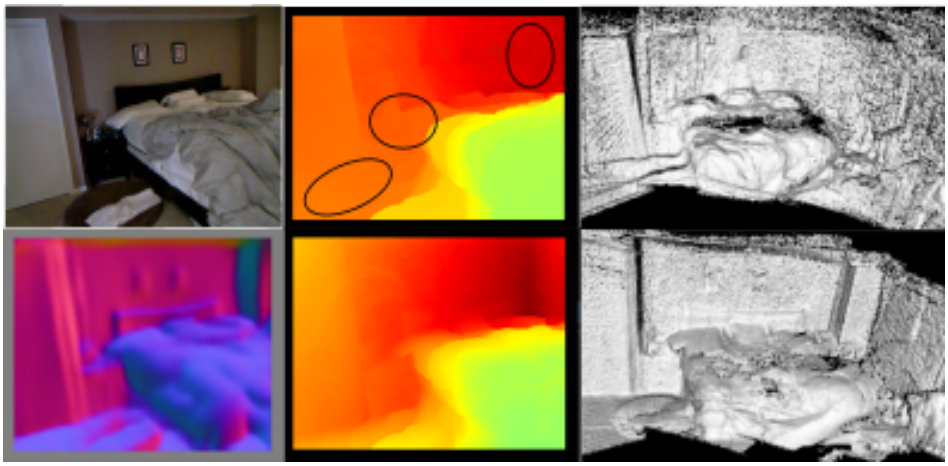

Fig. 6. Qualitative results on NYU raw dataset 'bedroom_0048' test sequence. Input rgb image (top left), reconstructed keyframe depth map with smoothness regularizer (top middle), fused reconstruction using smoothness regularizer (top right), keyframe surface normal prediction (bottom left), reconstructed keyframe depth map with normal-based regularizer (bottom middle), fused reconstruction using normal-based regularizer (bottom right). Note the more accurate reconstruction of the wall and floor overall when using the normal prior.

Fig. 7. Qualitative results on TUM dataset 'fr2_desk' sequence. From left-to-right are fused reconstruction using smoothness prior, fused reconstruction using normals prior, a rgb keyframe image in the sequence, and corresponding normal predictions for rgb keyframe image.

		Error (lower is better)				Accuracy (higher is better)		
		rms (m)	log	abs.rel	sq.rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
NYU-D V2 Raw 25 Test Scenes	CNN Depth [12]	0.637	0.226	0.163	0.135	0.738	0.937	0.982
	P.E. + Smoothness	0.522	0.206	0.123	0.111	0.834	0.949	0.979
	P.E. + Normals	**0.449**	**0.174**	**0.086**	**0.076**	**0.893**	**0.964**	**0.985**
TUM dataset 'fr2_desk'	CNN Depth [12]	1.141	0.368	0.227	0.261	0.543	0.820	0.923
	P.E. + Smoothness	0.678	0.254	0.132	0.127	0.788	0.889	**0.963**
	P.E. + Normals	**0.654**	**0.242**	**0.119**	**0.115**	**0.829**	**0.898**	**0.963**
ICL-NUIM dataset 'lr kt0'	CNN Depth [12]	0.829	0.426	0.295	0.261	0.472	0.781	0.905
	P.E. + Smoothness	0.322	0.175	0.123	0.058	0.828	0.966	**0.998**
	P.E. + Normals	**0.221**	**0.118**	**0.073**	**0.024**	**0.936**	**0.991**	**0.998**

TABLE I

QUANTITATIVE RESULTS ON 25 RAW NYU-D V2 DATASET TEST SEQUENCES, TUM DATASET 'FR2_DESK' SEQUENCE, AND ICL-NUIM DATASET 'LR KT0' SEQUENCE. P.E. = PHOTOMETRIC ERROR. THE AVERAGE ERRORS AND ACCURACY ARE FOR KEYFRAME RECONSTRUCTIONS AGAINST KINECT DEPTH MAPS (WHERE VALID DEPTHS ARE AVAILABLE). THE RESULTS HERE ARE SHOWN FOR THE OPTIMAL LAMBDA VALUES FOR NORMALS AND SMOOTHNESS REGULARIZER BASED ON FIG. 4, BUT WITH HIGHER NUMBER OF ITERATIONS WHICH ALLOWED FOR HIGHER ACCURACY IN THE RECONSTRUCTIONS.

depth respectively of a pixel location corresponding to p.

rms: $\sqrt{\frac{1}{|P|}\sum_{p\in P}\|d_p - d_p^{gt}\|^2}$

log rms : $\sqrt{\frac{1}{|P|}\sum_{p\in P}\|log(d_p) - log(d_p^{gt})\|^2}$

abs. rel: $\frac{1}{|P|}\sum_{p\in P}\frac{|d_p - d_p^{gt}|}{d_p^{gt}}$

sq. rel: $\frac{1}{|P|}\sum_{p\in P}\frac{\|d_p - d_p^{gt}\|^2}{d_p^{gt}}$

Accuracies: $\%$ of d_p s.t. $max(\frac{d_p}{d_p^{gt}}, \frac{d_p^{gt}}{d_p}) = \delta < thr$

The errors are computed at locations where both Kinect raw depth data is available and where depth regularization is performed (regions excluding the small border where predictions are not made). The regularized depth maps and CNN depth predictions are bilinearly upsampled to 640x480 resolution prior to evaluating against the raw Kinect depth maps. Note that the same optimisation and cost-volume-related parameters were used for comparing the two regularizer types. We follow the same θ scheduling policy as [4] with similar choice of parameters. The table, in particular the low threshold accuracy column, help validate that the normal-prior helps in recovering the fine details in the scene. Qualitative comparisons are shown for two NYU raw test sequences in Figures 5 and 6. The improvements in reconstruction in terms of both fine detail and global scene structure are apparent, especially in textureless regions.

The same experiments were carried out on the TUM dataset [38] and the living room sequence 'lr kt0' in the ICL-NUIM dataset [39]. Quantitative results for these sequences are also shown in Table I, and qualitative results for the TUM sequence is shown in Fig 7. Again a similar trend to that observed before can be seen. It can also be seen that CNN depth predictions do not generalize to new scene types as well as the other two methods.

While our experiments were limited to reconstructing indoor environments, the same framework in theory can be used for building dense maps of outdoor scenes, given the large depth range covered by the cost volume and large-scale volumetric fusion capabilities of [37]. However, the neural network (which is trained on indoor scenes) will likely require finetuning to adapt – this is yet to be validated. The main difficulty here is in acquiring densely labelled outdoor depth maps (required for generating ground truth normals) for training, although an unsupervised learning scheme similar to [40] should help in this regard.

V. CONCLUSION

In this work we presented a simple yet efficient solution that jointly exploits low-level geometry-based photometric evidence and high-level scene information captured from a multi-scale CNN architecture in the form of surface normals, for improving the accuracy of dense reconstructions in cases where otherwise there is very little photometric evidence. It was seen that incorporating learnt surface orientations enabled smooth and accurate reconstructions especially in areas with little photometric evidence to guide the solution. Deep learning has enabled prediction of geometry of objects and scenes directly from a single image and this alleviates the need for prior assumptions about scene structure, and handcrafted scene priors that are otherwise required for dense reconstruction. It was also seen that these networks are capable of generalizing to new types of environments well enough for practical use. We believe this work is a step forward in unifying the two complementary tasks of 3D reconstruction and scene understanding, aiding purely vision-based autonomous robots.

REFERENCES

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, September 2014.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2320–2327.

[5] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Proceedings of the 32Nd DAGM Conference on Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 11–20.

[6] A. Flint, D. Murray, and I. Reid, "Manhattan scene understanding using monocular, stereo, and 3d features," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2228–2235.

[7] A. Concha, M. W. Hussain, L. Montano, and J. Civera, "Manhattan and piecewise-planar constraints for dense monocular mapping," in *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[8] A. Concha and J. Civera, "Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 5686–5693.

[9] A. Dame, V. Prisacariu, C. Ren, and I. Reid, "Dense reconstruction using 3d object shape priors," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1288–1295.

[10] S. Bao, M. Chandraker, Y. Lin, and S. Savarese, "Dense object reconstruction with semantic priors," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1264–1271.

[11] D. Herrera C., J. Kannala, L. Ladický, and J. Heikkila, "Depth map inpainting under a second-order smoothness prior," in *Image Analysis*. Springer Berlin Heidelberg, 2013, vol. 7944, pp. 555–566.

[12] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *CoRR*, vol. abs/1411.4734, 2014.

[13] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015. [Online]. Available: http://arxiv.org/abs/1411.6387

[14] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape from shading: A survey," *IEEE Transactions on Pattern Analaysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999.

[15] C. Hane, L. Ladicky, and M. Pollefeys, "Direction matters: Depth estimation with a surface normal classifier," in *CVPR*, 2015, pp. 381–389.

[16] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[17] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2609–2616.

[18] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche, "Monofusion: Real-time 3d reconstruction of small scenes with a single web camera," in *ISMAR*, 2013.

[19] A. Concha and J. Civera, "Using superpixels in monocular slam," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 365–372.

[20] S. Y. Bao and S. Savarese, "Semantic structure from motion: A novel framework for joint object recognition and 3d reconstruction," in *Proceedings of the 15th International Conference on Theoretical Foundations of Computer Vision: Outdoor and Large-scale Real-world Scene Analysis*, Berlin, Heidelberg, 2012, pp. 376–397.

[21] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr, "Joint optimization for object class segmentation and dense stereo reconstruction," *International Journal of Computer Vision*, vol. 100, no. 2, pp. 122–133, 2012.

[22] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3d scene reconstruction and class segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 97–104.

[23] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science, 2014, vol. 8694, pp. 703–718.

[24] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *Int. J. Comput. Vision*, vol. 75, no. 1, pp. 151–172, Oct. 2007.

[25] L. Ladický, B. Zeisl, and M. Pollefeys, "Discriminatively trained dense surface normal estimation," in *ECCV*, 2014, vol. 8693, pp. 468–484.

[26] D. F. Fouhey, A. Gupta, and M. Hebert, "Data-driven 3d primitives for single image understanding," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3392–3399.

[27] X. Wang, D. F. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *CVPR*, 2015.

[28] J.-D. Durou, Y. Quéau, and J.-F. Aujol, "Normal Integration – Part I: A Survey," June 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01334349

[29] K. Kolev, T. Pock, and D. Cremers, "Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo," in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ser. ECCV'10, 2010, pp. 538–551.

[30] C. Hane, N. Savinov, and M. Pollefeys, "Class specific 3d object shape priors using surface normals," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014.

[31] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *ArXiv e-prints*, June 2015.

[32] R. Rockafellar, *Convex Analysis*, ser. Princeton landmarks in mathematics and physics. Princeton University Press, 1997.

[33] F. Steinbrücker, T. Pock, and D. Cremers, "Large displacement optical flow computation without warping," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1609–1614.

[34] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2010.

[35] http://www.nvidia.com.

[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[37] V. A. Prisacariu, O. Kahler, M. M. Cheng, C. Y. Ren, J. Valentin, P. H. S. Torr, I. D. Reid, and D. W. Murray, "A Framework for the Volumetric Integration of Depth Images," *ArXiv e-prints*, 2014.

[38] 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.

[39] 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.

[40] R. Garg, V. Kumar, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision, (ECCV)*, 2016.