

# Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction

Huangying Zhan<sup>1,2</sup>, Ravi Garg<sup>1,2</sup>, Chamara Saroj Weerasekera<sup>1,2</sup>, Kejie Li<sup>1,2</sup>, Harsh Agarwal<sup>3</sup>, Ian Reid<sup>1,2</sup>

<sup>1</sup>The University of Adelaide

<sup>2</sup>Australian Centre for Robotic Vision

<sup>3</sup>Indian Institute of Technology (BHU)

{huangying.zhan, ravi.garg, chamara.weerasekera, kejie.li, ian.reid}@adelaide.edu.au  
harsh.agarwal.eee14@iitbhu.ac.in

## Abstract

Despite learning based methods showing promising results in single view depth estimation and visual odometry, most existing approaches treat the tasks in a supervised manner. Recent approaches to single view depth estimation explore the possibility of learning without full supervision via minimizing photometric error. In this paper, we explore the use of stereo sequences for learning depth and visual odometry. The use of stereo sequences enables the use of both spatial (between left-right pairs) and temporal (forward backward) photometric warp error, and constrains the scene depth and camera motion to be in a common, real-world scale. At test time our framework is able to estimate single view depth and two-view odometry from a monocular sequence. We also show how we can improve on a standard photometric warp loss by considering a warp of deep features. We show through extensive experiments that: (i) jointly training for single view depth and visual odometry improves depth prediction because of the additional constraint imposed on depths and achieves competitive results for visual odometry; (ii) deep feature-based warping loss improves upon simple photometric warp loss for both single view depth estimation and visual odometry. Our method outperforms existing learning based methods on the KITTI driving dataset in both tasks. The source code is available at <https://github.com/Huangying-Zhan/Depth-VO-Feat>.

## 1. Introduction

Understanding the 3D structure of a scene from a single image is a fundamental question in machine perception. The related problem of inferring ego-motion from a sequence of images is likewise a fundamental problem in robotics, known as visual odometry estimation. These two

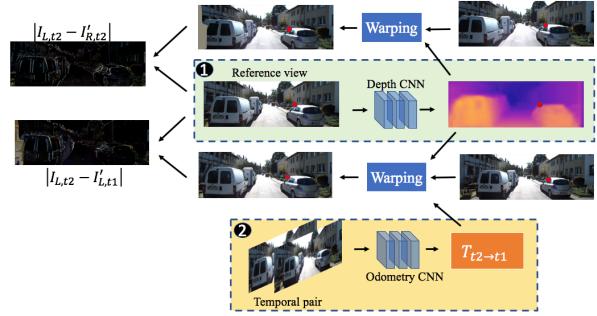


Figure 1. Training instance example. The known camera motion between stereo cameras  $T_{L \rightarrow R}$  constrains the Depth CNN and Odometry CNN to predict depth and relative camera pose with actual scale.

problems are crucial in robotic vision research since accurate estimation of depth and odometry based on images has many important applications, most notably for autonomous vehicles.

While both problems have been the subject of research in robotic vision since the origins of the discipline, with numerous geometric solutions proposed, in recent times a number of works have cast depth estimation and visual odometry as supervised learning problems [1][5][21][22]. These methods attempt to predict depth or odometry using models that have been trained from a large dataset with ground truth data. However, these annotations are expensive to obtain, e.g. expensive laser or depth camera to collect depths. In a recent work Garg *et al.*[6] recognised that these tasks are amenable to an unsupervised framework where the authors propose to use photometric warp error as a self-supervised signal to train a convolutional neural network (ConvNet / CNN) for the single view depth estimation. Following [6] methods like [9][19][41] use the photometric error based supervision to learn depth estimators

comparable to that of fully supervised methods. Specifically, [6] and [9] use the photometric warp error between left-right images in a stereo pair to learn depth. Recognising the generality of the idea, [44] uses monocular sequences to jointly train two neural networks for depth and odometry estimation. However, relying on the two frame visual odometry estimation framework, [44] suffers from the per frame scale-ambiguity issue, in that an actual metric scaling of the camera translations is missing and only direction is known. Having a good estimate of the translation scale per-frame is crucial for the success of any Simultaneous Localization and Mapping (SLAM) system. Accurate camera tracking in most monocular SLAM frameworks relies on keeping the scale consistency of the map across multiple images which is enforced using a single scale map. In absence of a global map for tracking, an expensive bundle adjustment over the per-frame scale parameter or additional assumptions like constant camera height from the already detected ground plane becomes essential for accurate tracking [34].

In this work, we propose a framework which jointly learns a single view depth estimator and monocular odometry estimator using stereo video sequences (as shown in Figure 1) for training. Our method can be understood as unsupervised learning for depth estimation and semi-supervised for pose which is known between stereo pairs. The use of stereo sequences enables the use of both spatial (between left-right pairs) and temporal (forward-backward) photometric warp error, and constrains the scene depth and camera motion to be in a common, real-world scale (set by the stereo baseline). Inference (i.e. depth and odometry estimation) without any scale ambiguity is then possible using a single camera for pure frame to frame VO estimation without any need for mapping.

Moreover, while the previous works have shown the efficacy of using the photometric warp error as a self-supervision signal, a simple warp of image intensities or colors carries its own assumptions about brightness/color consistency, and must also be accompanied by a regularization to generate “sensible” warps when the photometric information is ambiguous, such as in uniformly colored regions (see Sec.3.3). We propose an additional deep feature reconstruction loss which takes contextual information into consideration rather than per pixel color matching alone.

In summary, we make the following contributions: (i) an unsupervised framework for jointly learning a depth estimator and visual odometry estimator that does not suffer from the scale ambiguity; (ii) takes advantage of the full set of constraints available from spatial and temporal image pairs to improve upon prior art on single view depth estimations; (iii) produces the state-of-the-art frame-to-frame odometry results that significantly improve on [44] and are on par with geometric methods; (iv) uses a novel feature reconstruction

loss in addition to the color intensity based image reconstruction loss which improves the depth and odometry estimation accuracy significantly.

## 2. Related Work

Humans are capable of reasoning the relative depth of pixels in an image and perceive ego-motion given two images, but both single view depth estimation and two frame visual odometry are challenging problems. Avoiding visual learning, localization and 3D reconstruction in computer vision was considered a purely geometric problem for decades. While prior to deep learning graphical models based learning methods [32][33] were prevalent examples for single view reconstructions, methods based on the epipolar geometry were to the fore for two view odometry. While it is possible to estimate the relative pose between two frames based only on the data within those two frames up-to a scale (see e.g., [24], the “gold-standard” for geometric ego-motion estimation to date is based on a batch bundle adjustment of pose and scene structure [35], or on online Visual SLAM techniques [4]). After the surge of convolutional neural networks, both depth and visual odometry estimation problem have been attempted with deep learning methods.

**Supervised methods** Deep learning based depth estimation starts with Eigen *et al.* [5] which is the first work estimating depth with ConvNets. They used a multi-scale deep network and scale-invariant loss for depth estimation. Liu *et al.*[21] [22] formulated depth estimation as a continuous conditional random field learning problem. Laina *et al.* [20] proposed a residual network using fully convolutional architecture to model the mapping between monocular image and depth map. They also introduced reverse Huber loss and newly designed up-sampling modules. Kendall *et al.*[17] proposed an end-to-end learning framework to predict disparity from a stereo pair. In particular, they propose to use an explicit feature matching step as a layer in the network to create the cost-volume matching two images, which is then regularized to predict the state-of-the-art disparities for outdoor stereo sequences on KITTI dataset.

For odometry, Agrawal *et al.* [1] proposed a visual feature learning algorithm which aims at learning good visual features. Instead of learning features from a classification task (e.g. ImageNet[31]), [1] learns features from an ego-motion estimation task. The model is capable to estimate relative camera poses. Wang *et al.*[38] presented a recurrent ConvNet architecture for learning monocular odometry from video sequences.

Ummenhofer *et al.* [36] proposed an end-to-end visual odometry and depth estimation network by formulating structure from motion as a supervised learning problem.

However, the work is highly supervised: not only does it require depth and camera motion ground truths, in addition the surface normals and optical flow between images are also required.

**Unsupervised or semi-supervised methods** Recent works suggest that unsupervised pipeline for learning depth is possible from stereo image pairs using a photometric warp loss to replace a loss based on ground truth depth. Garg *et al.* [6] used binocular stereo pairs (for which the inter-camera transformation is known) and trained a network to predict the depth that minimises the photometric difference between the true right image and one synthesized by warping the left image into the right’s viewpoint, using the predicted depth. Godard *et al.* [9] made improvements to the depth estimation by introducing a symmetric left-right consistency criterion and better stereo loss function. [19] proposed a semi-supervised learning framework by using both sparse depth maps for supervised learning and dense photometric error for unsupervised learning.

An obvious extension to the above framework is to use structure-from-motion techniques to estimate the inter-frame motion (optic flow) [15] instead of depth using the known stereo geometry. But in fact it is possible to go further and to use deep networks also to estimate the camera ego-motion, as shown very recently by [44] and [37], both of which use a photometric error for supervising a monocular depth and ego-motion estimation system. Similar to other monocular frameworks, [44] and [37] suffer from scaling ambiguity issue.

Like [6][9], in our work we use stereo pairs for training, since this avoids issues with the depth-speed ambiguity that exist in monocular 3D reconstruction. In addition we jointly train a network to also estimate ego-motion from a pair of images. This allows us to enforce both the temporal *and* stereo constraints to improve our depth estimation in a joint framework.

All of the unsupervised depth estimation methods rely on photo-consistency assumption which gets violated often in practice. To cope with that [6][44] use robust norms like L1 norm of the warp error. [9] uses hand crafted features like SSIM [39]. Other handcrafted features like SIFT [25], HOG [3], ORB [30] are all usable and can be explored in unsupervised learning framework for robust warping loss. More interestingly, one can learn good features specifically for the task of matching. LIFT [42] and MC-CNN [43] learn a similarity measure on small image patches while [40][2] learns fully convolutional features good for matching. In our work, we compare the following features for their potential for robust warp error minimization: standard RGB photo-consistency; ImageNet features (conv1); features from [40]; features from a “self-supervised” version of [40]; and features derived from our depth network.

### 3. Method

This section describes our framework (shown in Figure 2) for jointly learning a single view depth ConvNet ( $\text{CNN}_D$ ) and a visual odometry ConvNet ( $\text{CNN}_{VO}$ ) from stereo sequences. The stereo sequences learning framework overcomes the scaling ambiguity issue with monocular sequences, and enables the system to take advantage of both left-right (spatial) and forward-backward (temporal) consistency checks.

#### 3.1. Image reconstruction as supervision

The fundamental supervision signal in our framework comes from the task of image reconstruction. For two nearby views, we are able to reconstruct the reference view from the live view, given that the depth of the reference view and relative camera pose between two views are known. Since the depth and relative camera pose can be estimated by a ConvNet, the inconsistency between the real and the reconstructed view allows the training of the ConvNet. However, a monocular framework without extra constraints [44] suffers from the scaling ambiguity issue. Therefore, we propose a stereo framework which constrains the scene depth and relative camera motion to be in a common, real-world scale, given an extra constraint set by the known stereo baseline.

In our proposed framework using stereo sequences, for each training instance, we have a temporal pair ( $I_{L,t1}$  and  $I_{L,t2}$ ) and a stereo pair ( $I_{L,t2}$  and  $I_{R,t2}$ ), where  $I_{L,t2}$  is the reference view while  $I_{L,t1}$  and  $I_{R,t2}$  are the live views. We can synthesize two reference views,  $I'_{L,t1}$  and  $I'_{R,t2}$ , from  $I_{L,t1}$  and  $I_{R,t2}$ , respectively. The synthesis process can be represented by,

$$I'_{L,t1} = f(I_{L,t1}, K, T_{t2 \rightarrow t1}, D_{L,t2}) \quad (1)$$

$$I'_{R,t2} = f(I_{R,t2}, K, T_{L \rightarrow R}, D_{L,t2}). \quad (2)$$

where  $f(\cdot)$  is a synthesis function defined in Sec.3.2;  $D_{L,t2}$  denotes the depth map of the reference view;  $T_{L \rightarrow R}$  and  $T_{t2 \rightarrow t1}$  are the relative camera pose transformations between the reference view and the live views; and  $K$  denotes the known camera intrinsic matrix. Note that  $D_{L,t2}$  is mapped from  $I_{L,t2}$  via  $\text{CNN}_D$  while  $T_{t2 \rightarrow t1}$  is mapped from  $[I_{L,t1}, I_{L,t2}]$  via  $\text{CNN}_{VO}$ .

The image reconstruction loss between the synthesized views and the real views are computed as a supervision signal to train  $\text{CNN}_D$  and  $\text{CNN}_{VO}$ . The image construction loss is represented by,

$$L_{ir} = \sum_p (|I_{L,t2}(p) - I'_{L,t1}(p)| + |I_{L,t2}(p) - I'_{R,t2}(p)|). \quad (3)$$

The effect of using stereo sequences instead of monocular sequences is two-fold. The known relative pose  $T_{L \rightarrow R}$  be-

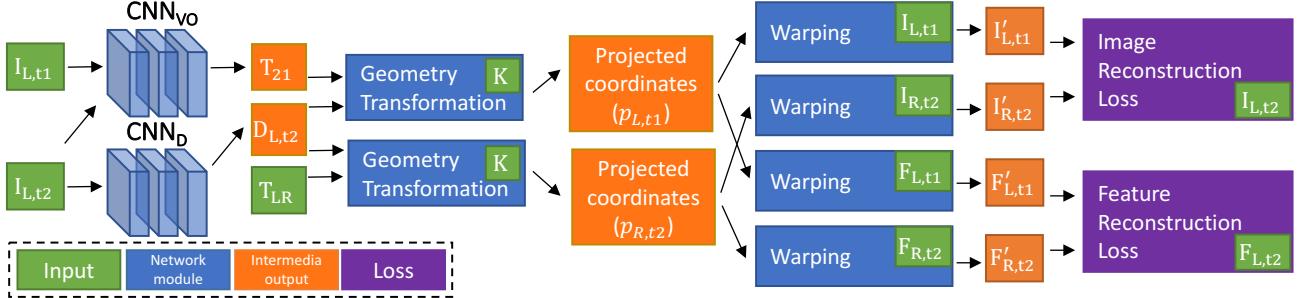


Figure 2. Illustration of our proposed framework in training phase.  $\text{CNN}_{VO}$  and  $\text{CNN}_D$  can be used independently in testing phase.

tween the stereo pair constrains  $\text{CNN}_D$  and  $\text{CNN}_{VO}$  to estimate depths and relative pose between the temporal pair in a real-world scale. As a result, our model is able to estimate single view depths and two-view odometry without the scaling ambiguity issue at test time. Second, in addition to stereo pairs with only one live view, the temporal pair provides a second live view for the reference view. The multi-view scenario takes advantage of the full set of constraints available from the stereo and temporal image pairs.

In this section, we describe an unsupervised framework that learns depth estimation and visual odometry without scaling ambiguity issue using stereo video sequences.

### 3.2. Differentiable geometry modules

As indicated in Eqn.1 - 2, an important function in our learning framework is the synthesis function,  $f(\cdot)$ . The function consists two differentiable operations which allow gradient propagation for the training of the ConvNet. The two operations are epipolar geometry transformation and warping. The former defines the correspondence between pixels in two views while the latter synthesizes an image by warping a live view.

Let  $p_{L,t2}$  be the homogeneous coordinates of a pixel in the reference view. We can obtain  $p_{L,t2}$ 's projected coordinates onto the live views using epipolar geometry, similar to [10, 44]. The projected coordinates are obtained by

$$p_{R,t2} = KT_{L \rightarrow R}D_{L,t2}(p_{L,t2})K^{-1}p_{L,t2} \quad (4)$$

$$p_{L,t1} = KT_{t2 \rightarrow t1}D_{L,t2}(p_{L,t2})K^{-1}p_{L,t2}, \quad (5)$$

where  $p_{R,t2}$  and  $p_{L,t1}$  are the projected coordinates on  $I_{R,t2}$  and  $I_{L,t1}$  respectively. Note that  $D_{L,t2}(p_{L,t2})$  is the depth at position  $p_{L,t2}$ ;  $T \in SE3$  is a 4x4 transformation matrix defined by 6 parameters, in which a 3D vector  $\mathbf{u} \in so3$  is an axis-angle representation and a 3D vector  $\mathbf{v} \in \mathbb{R}^3$  represents translations.

After getting the projected coordinates from Eqn.4 - 5, new reference frames can be synthesized from the live frames using the differentiable bilinear interpolation mechanism (warping) proposed in [14].

### 3.3. Feature reconstruction as supervision

The stereo framework we proposed above implicitly assumes that the scene is Lambertian, so that the brightness is constant regardless the observer's angle of view. This condition implies that the image reconstruction loss is meaningful for training the ConvNets. Any violation of the assumption can potentially corrupt the training process by propagating the wrong gradient back to the ConvNets. To improve the robustness of our framework, we propose a feature reconstruction loss: instead of using 3-channel color intensity information solely (image reconstruction loss), we explore the use of dense features as an additional supervision signal.

Let  $F_{L,t2}$ ,  $F_{L,t1}$  and  $F_{R,t2}$  be the corresponding dense feature representations of  $I_{L,t2}$ ,  $I_{L,t1}$  and  $I_{R,t2}$  respectively. Similar to the image synthesis process, two reference views,  $F'_{L,t1}$  and  $F'_{R,t2}$ , can be synthesized from  $F_{L,t1}$  and  $F_{R,t2}$ , respectively. The synthesis process can be represented by,

$$F'_{L,t1} = f(F_{L,t1}, K, T_{t2 \rightarrow t1}, D_{L,t2}) \quad (6)$$

$$F'_{R,t2} = f(F_{R,t2}, K, T_{L \rightarrow R}, D_{L,t2}). \quad (7)$$

Then, the feature reconstruction loss can be formulated as,

$$L_{fr} = \sum_p |F_{L,t2}(p) - F'_{L,t1}(p)| + \sum_p |F_{L,t2}(p) - F'_{R,t2}(p)| \quad (8)$$

In this work, we explore four possible dense features, as detailed in Section 4.3.

### 3.4. Training loss

As introduced in Sec.3.1 and Sec.3.3, the main supervision signal in our framework comes from the image reconstruction loss while the feature reconstruction loss acts as an auxiliary supervision. Furthermore, similar to [6][44][9], we have a depth smoothness loss which encourages the predicted depth to be smooth.

To obtain a smooth depth prediction, following the approach adopted by [12][9], we encourage depth to be smooth locally by introducing an edge-aware smoothness

term. The depth discontinuity is penalized if image continuity is showed in the same region. Otherwise, the penalty is small for discontinued depths. The edge-aware smoothness loss is formulate as

$$L_{ds} = \sum_{m,n}^{W,H} |\partial_x D_{m,n}| e^{-|\partial_x I_{m,n}|} + |\partial_y D_{m,n}| e^{-|\partial_y I_{m,n}|}, \quad (9)$$

where  $\partial_x(\cdot)$  and  $\partial_y(\cdot)$  are gradients in horizontal and vertical direction respectively. Note the  $D_{m,n}$  is inverse depth in the above regularization.

The final loss function becomes

$$L = \lambda_{ir} L_{ir} + \lambda_{fr} L_{fr} + \lambda_{ds} L_{ds}, \quad (10)$$

where  $\lambda_{ir}$ ,  $\lambda_{fr}$  and  $\lambda_{ds}$  are the loss weightings for each loss term.

### 3.5. Network architecture

**Depth estimation** Our depth ConvNet is composed of two parts, encoder and decoder. For the encoder, we adopt the convolutional network in a variant of ResNet50 [11] with half filters (ResNet50-1by2) for the sake of computation cost. The ResNet50-1by2 contains less than 7 million parameters which is around one fourth of the original ResNet50. For the decoder network, the decoder firstly converts the encoder output (1024-channel feature maps) into a single channel feature map using a 1x1 kernel, followed by conventional bilinear upsampling kernels with skip-connections. Similar to [23][6][9], the decoder uses skip-connections to fuse low-level features from different stages of the encoder. We use ReLU activation after the last prediction layer to ensure positive prediction comes from the depth ConvNet. For the output of the depth ConvNet, we design our framework to predict inverse depth instead of depth. However, the ReLU activation may cause zero estimation which results in infinite depth. Therefore, we convert the predicted inverse depth to depth by  $D = 1/(D_{inv} + 10^{-4})$ .

**Visual odometry** The visual odometry ConvNet is designed to take two concatenated views along the color channels as input and output a 6D vector  $[\mathbf{u}, \mathbf{v}] \in se3$ , which is then converted to a 4x4 transformation matrix. The network is composed of 6 stride-2 convolutions followed by 3 fully-connected layers. The last fully-connected layer gives the 6D vector, which defines the transformation from reference view to live view  $T_{ref \rightarrow live}$ .

## 4. Experiments

In this section we show extensive experiments for evaluating the performance of our proposed framework. We favorably compare our approach on KITTI dataset [8][7] with

prior art on both single view depth and visual odometry estimation. Additionally, we perform a detailed ablation study on our framework to show that using temporal consistency while training and use of learned deep features along with color consistency both improves the single view depth predictions. Finally, we show two variants of deep features and the corresponding effect, which we show examples of using deep features for dense matching.

We train all our CNNs with the Caffe [16] framework. We use Adam optimizer with the proposed optimization settings in [18] with  $[\beta_1, \beta_2, \epsilon] = [0.9, 0.999, 10^{-8}]$ . The initial learning rate is 0.001 for all the trained network, which we decrease manually when the training loss converges. For the loss weighting in our final loss function, we empirically find that the combination  $[\lambda_{ir}, \lambda_{fr}, \lambda_{ds}] = [1, 0.1, 10]$  results in a stable training. No data augmentation is involved in our work.

Our system is trained mainly in KITTI dataset [7][8]. The dataset contains 61 video sequences with 42,382 rectified stereo pairs, with the original image size being 1242x375 pixels. However, we use image size of 608x160 in our training setup for the sake of computation cost. We use two different splits of the KITTI dataset for evaluating estimated ego-motion and depth. For single view depth estimation, we follow the Eigen split provided by [5] for fair comparisons with [6, 9, 5, 22]. On the other hand, in order to evaluate our visual odometry performance and compare to prior approaches, we follow [44] by training both the depth and pose network on the official KITTI Odometry training set. Note that there are overlapping scenes between two splits (i.e. some testing scenes of Eigen Split are included in the training scenes of Odometry Split, and vice versa). Therefore, finetuning/testing models trained in any split to another split is not allowable/sensible. The detail about both splits are:

**Eigen Split** Eigen *et al.* [5] select 697 images from 28 sequences as test set for single view depth evaluation. The remaining 33 scenes contains 23,488 stereo pairs for training. We follow this setup and form 23,455 temporal stereo pairs.

**Odometry Split** The KITTI Odometry Split [8] contains 11 driving sequences with publicly available ground truth camera poses. We follow [44] to train our system on the Odometry Split (no finetuning from Eigen Split is performed). The split in which sequences 00-08 (sequence 03 is not available in KITTI Raw Data) are used for training while 09-10 are used for evaluation. The training set contains 8 sequences with 19,600 temporal stereo pairs.

For each dataset split, we form temporal pairs by choosing frame  $I_t$  as the live frame while frame  $I_{t+1}$  as the reference frame – to which the live frame is warped. The reason for this choice is that as the mounted camera in KITTI moves forward, most pixels in  $I_{t+1}$  have correspondence in

| Method                   | Seq. 09       |                          | Seq. 10       |                          |
|--------------------------|---------------|--------------------------|---------------|--------------------------|
|                          | $t_{err}(\%)$ | $r_{err}(\text{°}/100m)$ | $t_{err}(\%)$ | $r_{err}(\text{°}/100m)$ |
| ORB-SLAM (LC) [26]       | 16.23         | 1.36                     | /             | /                        |
| ORB-SLAM [26]            | 15.30         | 0.26                     | 3.68          | 0.48                     |
| Zhou <i>et al.</i> [44]  | 17.84         | 6.78                     | 37.91         | 17.78                    |
| <b>Ours (Temporal)</b>   | 11.93         | 3.91                     | 12.45         | 3.46                     |
| <b>Ours (Full-NYUv2)</b> | 11.92         | 3.60                     | 12.62         | 3.43                     |

Table 1. Visual odometry result evaluated on Sequence 09, 10 of KITTI Odometry dataset.  $t_{err}$  is average translational drift error.  $r_{err}$  is average rotational drift error.

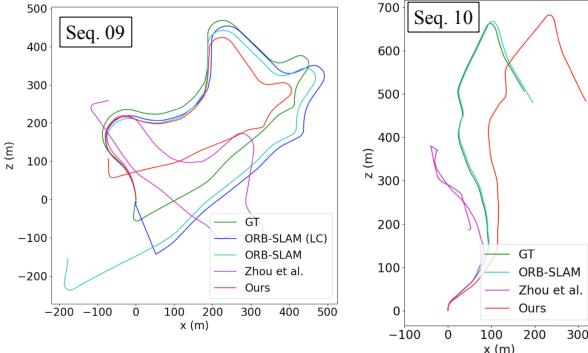


Figure 3. Qualitative result on visual odometry. Full trajectories on the testing sequences (09, 10) are plotted.

$I_t$  giving us a better warping error.

#### 4.1. Visual odometry results

We use the Odometry Split mentioned above to evaluate the performance of our frame to frame odometry estimation network. The result is compared with the monocular training based network [44] and a popular SLAM system – ORB-SLAM [26] (with and without loop closure) as very strong baselines. Both of the ORB-SLAM versions use local bundle adjustment and more importantly a single scale map to assist the tracking. We ignore the frames (First 9 and 30 respectively) from the sequences (09 and 10) for which ORB-SLAM fails to bootstrap with reliable camera poses due to lack of good features and large rotations. Following the KITTI Visual Odometry dataset evaluation criterion we use possible sub-sequences of length (100, 200, ..., 800) meters and report the average translational and rotational errors for the testing sequence 09 and 10 in Table 1.

As ORB-SLAM suffers from a single depth-translation scale ambiguity for the whole sequence, we align the ORB-SLAM trajectory with ground-truth by optimizing the map scale following standard protocol. For our method, we simply integrate the estimated frame-to-frame camera poses over the entire sequence without any post processing. Frame-to-frame pose estimation of [44] only avails small 5-frame long tracklets, each of which is already aligned independently with the ground-truth by fixing translation scales. This translation normalization leaves [44]’s error to only in-

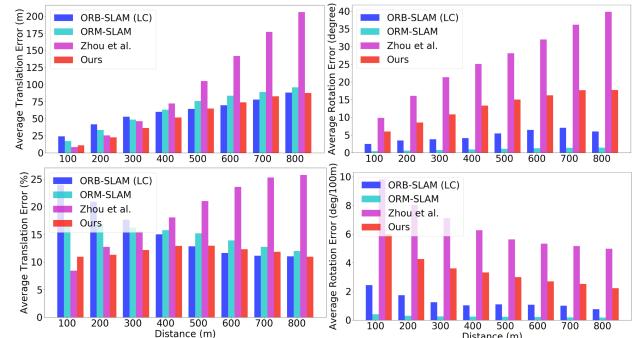


Figure 4. Comparison of VO error with different translation threshold for sequence 09 of odometry dataset.

dicate the relative translation magnitudes error over small sequences. As the KITTI sequences are recorded by camera mounted on a car which mostly move forward, even average 6DOF motion as reported in [44] overperforms frame-to-frame odometry methods (ORB-SLAM when used only on 5 frames does not bootstrap mapping). Nonetheless we simply integrate the aligned tracklets to estimate the full trajectory for [44] and evaluate. It is important to note that this evaluation protocol is highly disadvantageous to the proposed method as no scope for correcting the drift or translation scale is permitted. A visual comparison of the estimated trajectories for all the methods can be seen in Figure 3.

As can be seen in Table 1, our stereo based odometry learning method outperforms monocular learning method [44] by a large margin even without any further post-processing to fix translation scales. Our method is able to give comparable odometry results on sequence 09 to that of the full ORB-SLAM and respectable trajectory for sequence 10 on which larger error in our frame to frame rotation estimation leads to a much larger gradual drift which should be fixed by bundle adjustment.

To further compare the effect of bundle adjustment, we evaluate the average errors for different translation bins and report the result for sequence 09 in Figure 4. It can be seen clearly that both our method and [44] are better than ORB-SLAM when the translation magnitude is small. As translation magnitude increases, the simple integration of frame to frame VO starts drifting gradually, which suggests a clear advantage of a map based tracking over frame to frame VO without bundle adjustment.

#### 4.2. Depth estimation results

We use the Eigen Split to evaluate our system and compare the results with various state of the art depth estimation methods. Following the evaluation protocol proposed in [9] which uses the same crop as [6], we use both the 50m and 80m threshold of maximum depth for evaluation and report all standard error measures in Table 2 with some vi-

| Method                       | Dataset | Supervision | Error metric |       |       |          | Accuracy metric |                   |                   |
|------------------------------|---------|-------------|--------------|-------|-------|----------|-----------------|-------------------|-------------------|
|                              |         |             | Abs Rel      | SqRel | RMSE  | RMSE log | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ |
| Depth: cap 80m               |         |             |              |       |       |          |                 |                   |                   |
| Train set mean               | K       | Depth       | 0.361        | 4.826 | 8.102 | 0.377    | 0.638           | 0.804             | 0.894             |
| Eigen <i>et al.</i> [5] Fine | K       | Depth       | 0.203        | 1.548 | 6.307 | 0.282    | 0.702           | 0.890             | 0.958             |
| Liu <i>et al.</i> [22]       | K       | Depth       | 0.201        | 1.584 | 6.471 | 0.273    | 0.680           | 0.898             | 0.967             |
| Zhou <i>et al.</i> [44]      | K       | Mono.       | 0.208        | 1.768 | 6.856 | 0.283    | 0.678           | 0.885             | 0.957             |
| Garg <i>et al.</i> [6]       | K       | Stereo      | 0.152        | 1.226 | 5.849 | 0.246    | 0.784           | 0.921             | 0.967             |
| Godard <i>et al.</i> [9]     | K       | Stereo      | 0.148        | 1.344 | 5.927 | 0.247    | 0.803           | 0.922             | 0.964             |
| <b>Ours (Temporal)</b>       | K       | Stereo      | 0.144        | 1.391 | 5.869 | 0.241    | 0.803           | 0.928             | 0.969             |
| <b>Ours (Full-NYUv2)</b>     | K       | Stereo      | 0.135        | 1.132 | 5.585 | 0.229    | 0.820           | 0.933             | 0.971             |
| Depth: cap 50m               |         |             |              |       |       |          |                 |                   |                   |
| Zhou <i>et al.</i> [44]      | K       | Mono.       | 0.201        | 1.391 | 5.181 | 0.264    | 0.696           | 0.900             | 0.966             |
| Garg <i>et al.</i> [6]       | K       | Stereo      | 0.169        | 1.080 | 5.104 | 0.273    | 0.740           | 0.904             | 0.962             |
| Godard <i>et al.</i> [9]     | K       | Stereo      | 0.140        | 0.976 | 4.471 | 0.232    | 0.818           | 0.931             | 0.969             |
| <b>Ours (Temporal)</b>       | K       | Stereo      | 0.135        | 0.905 | 4.366 | 0.225    | 0.818           | 0.937             | 0.973             |
| <b>Ours (Full-NYUv2)</b>     | K       | Stereo      | 0.128        | 0.815 | 4.204 | 0.216    | 0.835           | 0.941             | 0.975             |

Table 2. Comparison of single view depth estimation performance with existing approaches. For training, K is KITTI dataset (Eigen Split). For a fair comparison, all methods (except [5]) are evaluated on the cropped region from [9]. For the supervision, “Depth” means ground truth depth is used in the method; “Mono.” means monocular sequences are used in the training; “Stereo” means stereo sequences with known stereo camera poses in the training.

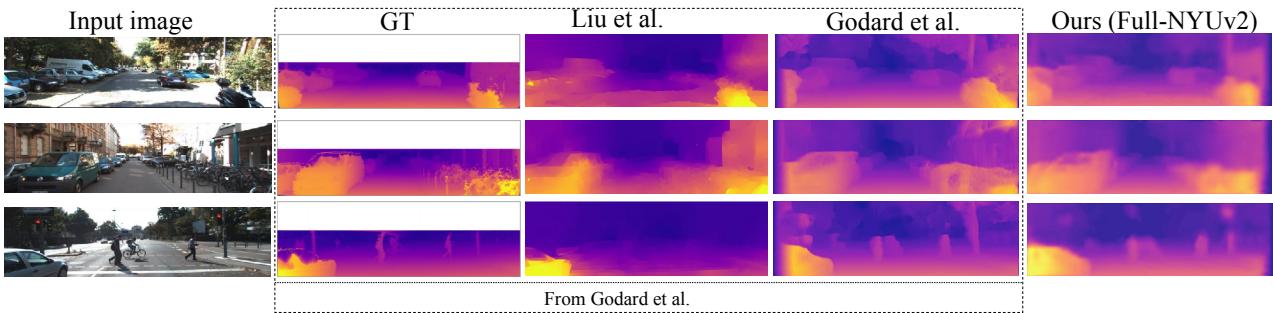


Figure 5. Single view depth estimation examples in Eigen Split. The ground truth depth is interpolated for visualization purpose.

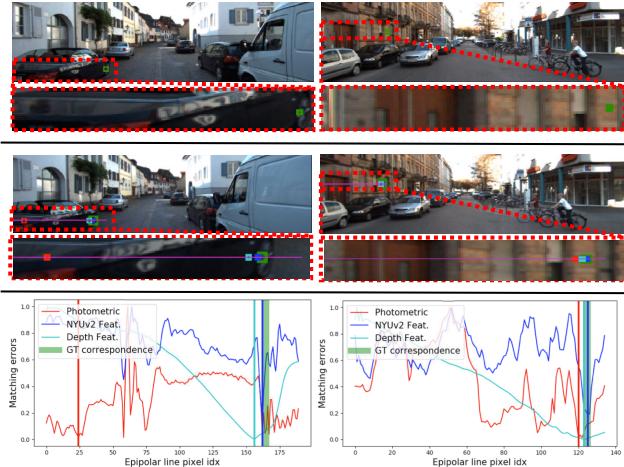


Figure 6. Stereo matching examples. Rows: (1) Left image; (2) Right image; (3) Matching error using color intensity and deep features. Photometric loss is not robust when compared with feature loss, especially in ambiguous regions.

sual examples in Figure 5. As shown in [6], photometric stereo based training with AlexNet-FCN architecture and Horn and Schunck [13] loss already gave more accurate results than the state of the art supervised methods [5][22] on KITTI. For fair comparison of [6] with other methods we evaluate the results reported by the authors publicly with 80m cap on maximum depth. All methods using stereo for training are substantially better than [44] which is using only monocular training. Benefited by the feature based reconstruction loss and additional warp error via odometry network, our method outperforms both [6] and [9] with reasonable margin. It is important to note that unlike [9] left-right consistency, data augmentation, run-time shuffle, robust similarity measure like SSIM[39] are not used to train our network and should lead to further improvement.

### 4.3. Ablation studies

Table 3 shows an ablation study on depth estimation for our method showing importance of each component of the loss function. Our first baseline is a simple architecture (ResNet50-1by2 as encoder; Bilinear upsampler as de-

| Method   | Stereo | Temporal | Feature | Error metric |       |       |          | Accuracy metric |                   |                   |
|--|--------|----------|---------|--------------|-------|-------|----------|-----------------|-------------------|-------------------|
|  |        |          |         | Abs Rel      | SqRel | RMSE  | RMSE log | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ |
| Encoder: ResNet50-1by2; Decoder: Bilinear upsampler  |        |          |         |              |       |       |          |                 |                   |                   |
| Baseline   | ✓      | ✗        | ✗       | 0.143        | 0.859 | 4.310 | 0.229    | 0.802           | 0.933             | 0.973             |
| Temporal   | ✓      | ✓        | ✗       | 0.135        | 0.905 | 4.366 | 0.225    | 0.818           | 0.937             | 0.973             |
| ImageNet Feat.                                       | ✓      | ✗        | ✓       | 0.136        | 0.880 | 4.390 | 0.230    | 0.823           | 0.935             | 0.970             |
| KITTI Feat.  | ✓      | ✗        | ✓       | 0.130        | 0.860 | 4.271 | 0.221    | 0.831           | 0.938             | 0.973             |
| NYUv2 Feat.  | ✓      | ✗        | ✓       | 0.132        | 0.906 | 4.279 | 0.220    | 0.831           | 0.939             | 0.974             |
| Full-NYUv2   | ✓      | ✓        | ✓       | 0.128        | 0.815 | 4.204 | 0.216    | 0.835           | 0.941             | 0.975             |
| Encoder: ResNet50-1by2; Decoder: Learnable upsampler |        |          |         |              |       |       |          |                 |                   |                   |
| Baseline2  | ✓      | ✗        | ✗       | 0.155        | 1.307 | 4.560 | 0.242    | 0.805           | 0.928             | 0.968             |
| Temporal2  | ✓      | ✓        | ✗       | 0.141        | 0.998 | 4.354 | 0.232    | 0.814           | 0.932             | 0.971             |
| Depth Feat.  | ✓      | ✗        | ✓       | 0.142        | 0.956 | 4.377 | 0.230    | 0.817           | 0.934             | 0.971             |
| Full-Depth   | ✓      | ✓        | ✓       | 0.137        | 0.893 | 4.348 | 0.228    | 0.821           | 0.935             | 0.971             |

Table 3. Ablation study on single view depth estimation. The result is evaluated in KITTI 2015 using Eigen Split test set, following the evaluation protocol proposed in [9]. The results are capped at 50m depth. Stereo: stereo pairs are used for training; Temporal: additional temporal pairs are used; Feature: feature reconstruction loss is used.

coder) trained on the stereo pairs with the loss described in Sec.3.4 which closely follows [6] (GitHub version). When we train the pose network jointly with the depth network, we get a slight improvement in depth estimation accuracy. Using features from ImageNet feature (conv1 features from pretrained ResNet50-1by-2) improves depth estimation accuracy slightly. In addition, using features from an off-the-shelf image descriptor [40] gives a further boost. However, [40] is trained using NYUv2 dataset [27] (ground truth poses and depths are required) so we follow [40] to train an image descriptor using KITTI dataset but using the estimated poses and depths generated from Method ‘‘Temporal’’ as pseudo ground truths. Using the features extracted from the self-supervised descriptor (KITTI Feat.) gives a comparable result with that of [40]. The system having all three components (Stereo + Temporal + NYUv2 Feat.) performs best as can be seen in the top part of Table 3.

As most other unsupervised depth estimation methods use a convolutional encoder with deconvnet architecture like [28][29] for dense predictions, we also experimented with learnable deconv architecture with the ResNet50-1by2 as encoder – learnable upsampler as decoder setup. The results in the bottom part of the table reflects that overall performance of this Baseline2 was slightly inferior to the first baseline. To improve the performance of this baseline, we explore the use of deep features extracted from the depth decoder itself. At the end the decoder outputs a 32-channel feature map which we directly use for feature reconstruction loss. Using these self-embedded depth features for additional warp error minimization also shows promising improvements in the accuracy of the depth predictions without requiring any explicit supervision for matching as required by [40].

In Figure 6, we compare the deep features of [40] and the self-embedded depth features against color consistency on the task of stereo matching. Photometric error is not as

robust as deep feature error, especially in texture-less regions, there are multiple local minima with similar magnitude. However, both NYUv2 Feature from [40] and self-embedded depth features show distinctive local minimum which is a desirable property.

## 5. Conclusion

We have presented an unsupervised learning framework for single view depth estimation and monocular visual odometry using stereo data for training. We have shown that the use of binocular stereo sequences for jointly learning the two tasks, enable odometry prediction in *metric scale* simply given 2 frames. We also show the advantage of using temporal image alignment, in addition to stereo pair alignment for single view depth predictions. Additionally, we have proposed a novel feature reconstruction loss to have state-of-the-art unsupervised single view depth and frame-to-frame odometry without scale ambiguity.

There are still a number of challenges to be addressed. Our framework assumes no occlusion and the scene is assumed to be rigid. Modelling scene dynamics and occlusions explicitly, in a deep learning framework will provide a natural means for more practical and useful navigation in real scenarios. Although we show odometry results that are comparable to the best two-frame estimates available the current systems do not compare favourably with state-of-the-art SLAM systems. An extensive study of CNN architectures more suitable for odometry estimation and a possible way of integrating the map information over time are challenging but very fruitful future directions.

## 6. Acknowledgement

This work was supported by the UoA Scholarship to HZ and KL, the ARC Laureate Fellowship FL130100102 to IR and the Australian Centre of Excellence for Robotic Vision CE140100016.

## References

- [1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 37–45, 2015.
- [2] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016.
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.
- [5] 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*, pages 2366–2374, 2014.
- [6] R. Garg, V. K. B G, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.
- [7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [9] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.
- [10] A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac, and A. Davison. gvnn: Neural network library for geometric computer vision. In *Computer Vision–ECCV 2016 Workshops*, pages 67–82. Springer, 2016.
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [12] P. Heise, S. Klose, B. Jensen, and A. Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013.
- [13] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.
- [14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.
- [15] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.
- [16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.
- [17] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.
- [18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [19] Y. Kuznetsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2215–2223. IEEE, 2017.
- [20] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.
- [21] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.
- [22] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2016.
- [23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [24] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.
- [25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [27] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- [28] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.
- [29] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
- [30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.
- [31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,

- A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [32] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.
  - [33] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2009.
  - [34] S. Song and M. Chandraker. Robust scale estimation in real-time monocular sfm for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1566–1573, 2014.
  - [35] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment: modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.
  - [36] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
  - [37] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.
  - [38] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2043–2050. IEEE, 2017.
  - [39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
  - [40] C. S. Weerasekera, R. Garg, and I. Reid. Learning deeply supervised visual descriptors for dense monocular reconstruction. *arXiv preprint arXiv:1711.05919*, 2017.
  - [41] M. Ye, E. Johns, A. Handa, L. Zhang, P. Pratt, and G.-Z. Yang. Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery. *arXiv preprint arXiv:1705.08260*, 2017.
  - [42] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.
  - [43] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.
  - [44] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.