BA-NET: DENSE BUNDLE ADJUSTMENT NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper introduces a network architecture to solve the structure-from-motion (SfM) problem via feature bundle adjustment (BA), which explicitly enforces multi-view geometry constraints in the form of feature reprojection error. The whole pipeline is differentiable, so that the network can learn suitable feature representations that make the BA problem more tractable. Furthermore, this work introduces a novel depth parameterization to recover dense per-pixel depth. The network first generates some bases depth maps according to the input image, and optimizes the final depth as a linear combination of these bases via feature BA. The bases depth map generator is also learned via end-to-end training. The whole system nicely combines domain knowledge (i.e. hard-coded multi-view geometry constraints) and machine learning (i.e. feature learning and basis depth map generator learning) to address the challenging dense SfM problem. Experiments on large scale real data prove the success of the proposed method.

1 INTRODUCTION

The Structure-from-Motion (SfM) problem has been extensively studied in the past a few decades. Almost all conventional SfM algorithms (Agarwal et al., 2011; Wu et al., 2011; Schönberger and Frahm, 2016; Engel et al., 2017; Delaunoy and Pollefeys, 2014) jointly optimize scene structures and camera motion via the Bundle-Adjustment (BA) algorithm (Triggs et al., 2000a; Agarwal et al., 2010), which minimizes the geometric (Agarwal et al., 2011; Wu et al., 2011; Schönberger and Frahm, 2016) or photometric (Engel et al., 2014; 2017; Delaunoy and Pollefeys, 2014) error through the Levenberg-Marquet (LM) algorithm (Nocedal and Wright, 2006). Some recent works (Ummenhofer et al., 2017; Zhou et al., 2017; Wang et al., 2018) attempt to solve SfM using deep learning techniques, but most of them do not enforce the geometric constraints between the 3D structures and camera motion in their network. For example, in the recent work DeMoN(Ummenhofer et al., 2017), the scene depth and camera motion are estimated by two individual sub-network branches.

This paper formulates BA as a differentiable layer, the BA-Layer, to bridge the gap between classic methods and recent deep learning based approaches. To this end, we learn a feed-forward multilayer perceptron (MLP) to predict the damping factor in the LM algorithm, which makes all involved computation differentiable. Forthermore, unlike conventional BA that minimizes geometric or photometric error, our BA-layer minimizes the distance between reprojected CNN features. Our novel 'feature-metric BA' takes CNN features of multiple images as input and optimizes for the camera motion and scene structures. This feature-metric BA is desirable, because it has been observed by Engel et al. (2014; 2017) that the geometric BA does not exploit all image information, while the photometric BA is sensitive to moving objects, exposure or white balance changes, etc. Most importantly, our BA-Layer can back-propagate loss from scene structures and camera motion to learn appropriate features that are most suitable for structure-from-motion and bundle adjustment. In this way, our network hard-codes the multi-view geometry constraints in the BA-Layer and learns suitable feature representations from training data.

We strive to estimate a dense per-pixel depth, because dense depth is critical for many tasks such as object detection and robot navigation. A major challenge in solving dense per-pixel depth is to find a compact parameterization. Direct per-pixel depth is computational expensive, which makes the network training intractable. So we train a network to generate a set of basis depth maps for an arbitrary input image and represent the result depth map as a linear combination of these bases. The combination coefficients will be optimized in the BA-Layer together with camera motion. This novel parameterization guarantees a smooth depth map with good consistency with object boundaries. It also reduces the number of unknowns and makes dense BA possible in neural networks.

Similar depth parameterization is introduced in a recent work, CodeSLAM (Bloesch et al., 2018). The major difference is that our method learns the basis depth map generator through the gradients back-propagated from the BA-Layer, while CodeSLAM learns the generator separately and uses its results for a standalone optimization component. Thus, our basis depth map generator has the chance to be better trained for the SfM problem. Furthermore, we use different network structures to generate depth bases. CodeSLAM employs a variational auto-encoder (VAE), while we use a standard encoder-decoder. This design enables us to use the same backbone network for both feature representations and depth parameterization, making joint training of the whole network possible.

To demonstrate the effectiveness of our method, we evaluate on the ScanNet Dai et al. (2017a) and KITTI (Geiger et al., 2012) dataset. Our method outperforms DeMoN (Ummenhofer et al., 2017), LS-Net (Clark et al., 2018), as well as several conventional baselines. Due to page limit, we move the ablation studies, evaluation on DeMoN's dataset, and multi-view SfM (up to 5 views) to the appendix.

2 Related Work

Monocular Depth Estimation Networks Estimating depth from a monocular image is an ill-posed problem because an infinite number of possible scenes may have produced the same image. Before the raise of deep learning based methods, some works predict depth from a single image based on MRF (Saxena et al., 2005; 2009), semantic segmentation (Ladický et al., 2014), or manually designed features (Hoiem et al., 2005). Eigen et al. (2014) proposed a multi-scale approach for depth prediction with two CNNs, where a coarse-scale network first predicts the scene depth at global level and then a fine-scale network will refine local regions. This approach was extended in Eigen and Fergus (2015) to handle semantic segmentation and surface normal estimation as well. Recently, Laina et al. (2016) proposed to use ResNet (He et al., 2016) based structure to predict depth, and Xu et al. (2017) constructed multi-scale CRFs for depth prediction. In comparison, we exploit monocular image depth estimation network for depth parameterization, which only produces a compact depth representation and the final result will be further improved through optimization.

Structure-from-Motion Networks Recently, some works exploit CNNs to resolve the structure-from-motion problem. Handa et al. Handa et al. (2016) solved the camera motion by a network from a pair of images with known depth. Zhou et al. Zhou et al. (2017) employed two CNNs for depth and camera motion estimation respectively, where both CNNs are trained jointly by minimizing the photometric loss in an unsupervised manner. Wang et al. Wang et al. (2018) implmented the direct method Steinbruecker et al. (2011) as a differentiable component to compute camera motion after scene depth is estimated by the method in Zhou et al. (2017). Ummenhofer et al. Ummenhofer et al. (2017) predicted the camera motion and scene depth from flow features, which help to make it generalizing better to unseen data. However, motion and depth are solved by two separate network branches, multi-view geometry constraints between motion and depth are not enforced. Recently, Ronald et al. Clark et al. (2018) proposed to solve nonlinear least squares optimization in two-view structure from motion using a LSTM-RNN(Hochreiter et al., 2001) as optimizer.

Our method belongs to this category. Unlike all previous works, we propose the BA-Layer to simultaneously predict the camera motion and scene depth from CNN features, which explicitly enforces multi-view geometry constraints. The hard-coded multi-view geometry constraint enables our method to reconstruct more than two images, while most deep learning methods can only handle two images. Furthermore, we proposed to minimze a 'feature-metric errors' instead of the photometric error in (Zhou et al., 2017; Wang et al., 2018; Clark et al., 2018) to enhance robustness.

3 Bundle Adjustment Revisit

Before introducing our BA-Net architecture, we revisit the classic BA to have a better understanding about where the difficulty is and why feature BA and feature learning are desirable. We only introduce the most relevant content and refer the audience to (Triggs et al., 2000b; Agarwal et al., 2010) for a comprehensive introduction. Given images $\mathbb{I} = \{I_i | i = 1 \cdots N_i\}$, the geometric BA (Triggs et al., 2000b; Agarwal et al., 2010) jointly optimizes camera pose parameters $\mathbb{T} = \{\boldsymbol{T}_i | i = 1 \cdots N_i\}$ and 3D point coordinates $\mathbb{P} = \{\boldsymbol{p}_j | j = 1 \cdots N_j\}$ by minimizing the reprojection error:

$$\mathcal{X} = \text{argmin} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \| e^g_{i,j}(\mathcal{X}) \|, \tag{1}$$

where the geometric distance

$$e^g_{i,j}(\mathcal{X}) = \pi(\boldsymbol{T}_i, \boldsymbol{p}_j) - \boldsymbol{q}_{i,j},$$

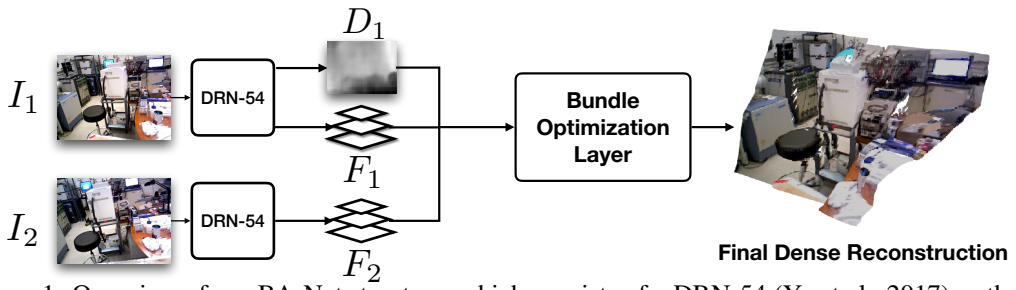


Figure 1: Overview of our BA-Net structure, which consists of a DRN-54 (Yu et al., 2017) as the backbone network, a **Depth Parameterization** sub-network that generates compact depth parameterization, a **Feature Pyramids** that extract multi-scale feature maps, and a **Bundle Adjustment Layer** that optimizes both the depth map and the camera poses through a novel differentiable LM algorithm.

measures the difference between a projected scene point and its corresponding feature point $\boldsymbol{q}_{i,j}$. The function π projects scene points to image space, and $\mathcal{X} = [\boldsymbol{T}_1, \boldsymbol{T}_2 \cdots \boldsymbol{T}_{N_i}, \boldsymbol{p}_1, \boldsymbol{p}_2 \cdots \boldsymbol{p}_{N_j}]^\top$ contains all point and camera parameters. The general strategy to minimize Equation (1) is the Levenberg-Marquardt (LM) (Nocedal and Wright, 2006; Lourakis and Argyros, 2005) algorithm. At each iteration, the LM algorithm solves for an update $\Delta\mathcal{X}$ to the solution by minimizing:

$$\Delta\mathcal{X} = \text{argmin}\|J(\mathcal{X})\Delta\mathcal{X} + E(\mathcal{X})\| + \lambda\|D(\mathcal{X})\Delta\mathcal{X}\|. \tag{2}$$

Here, $E(\mathcal{X}) = [e^g_{1,1}(\mathcal{X}), e^g_{1,2}(\mathcal{X}) \cdots e^g_{N_i,N_j}(\mathcal{X})]$, and $J(\mathcal{X})$ is the Jacobian of $E(\mathcal{X})$ respect to $\mathcal{X}$, $D(\mathcal{X})$ is a non-negative diagonal matrix, typically the square root of the diagonal of the approximated Hessian matrix $J(\mathcal{X})^\top J(\mathcal{X})$. The parameter λ is a non-negative parameter that controls the strength of regularization. The special structure of the parameters in BA motivates the use of Schur-Complement (Brown, 1958).

This geometric BA with reprojection error is the golden standard in structure-from-motion in the last two decades, but with two main drawbacks:

- Only image information conforming to the respective feature types, typically image corners, blobs, or line segments, is utilized.
- Features have to be matched to each other, which often result in a lot of outliers. Outlier rejection like RANSAC is necessary, which still cannot guarantee correct result.

These two difficulties motivate the recent development of direct methods (Engel et al., 2014; 2017; Delaunoy and Pollefeys, 2014), which propose the 'photometric BA' algorithm to eliminate feature matching and directly minimize the photometric error (pixel intensity difference) of reprojected pixels. The photometric error is defined as:

$$e^p_{i,j}(\mathcal{X}) = I_i(\pi(\boldsymbol{T}_i, d_j)) - I_1(\boldsymbol{q}_{1,j}), \tag{3}$$

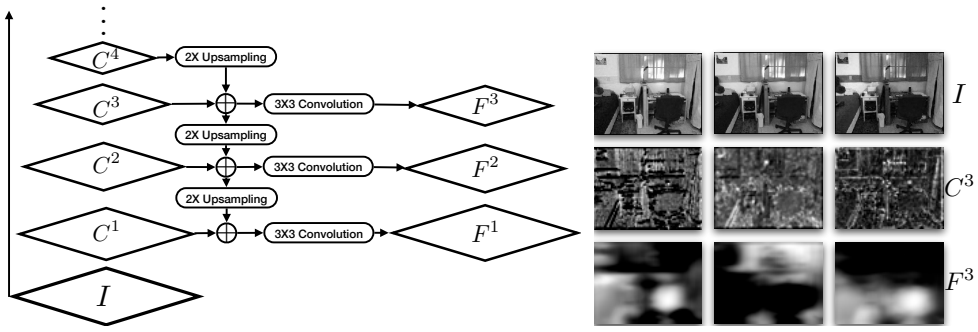
where $d_j \in \mathbb{D} = \{d_j | j = 1 \cdots N_j\}$ is the inverse depth of a pixel $\boldsymbol{q}_{1,j}$ at the image I_1. The direct methods have the advantages of using all pixels with sufficient gradient magnitude. They have demonstrated superior performance, especially at less textured scenes. However, these methods also have some drawbacks:

- They are sensitive to initialization as demonstrated in (Mur-Artal et al., 2015; Tang et al., 2017) because the photometric error increases the non-convexity (Engel et al., 2017).
- They are sensitive to camera exposure and white balance changes. An automatic photometric calibration is required (Engel et al., 2017; 2016).
- They are more sensitive to outliers such as moving objects.

4 THE BA-NET ARCHITECTURE

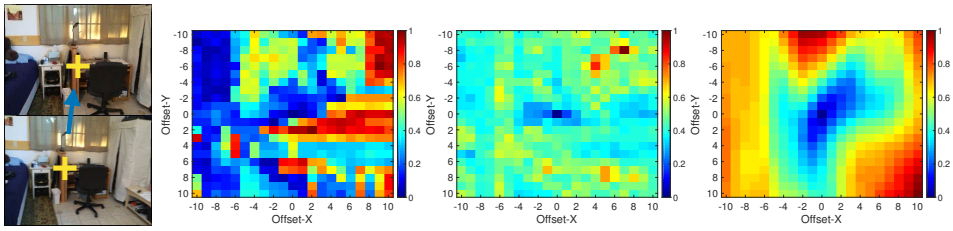
To deal with the above challenges, we propose a 'feature-metric BA' algorithm which minimizes the feature difference of reprojected pixels:

$$e^f_{i,j}(\mathcal{X}) = F_i(\pi(\boldsymbol{T}_i, d_j)) - F_1(\boldsymbol{q}_{1,j}), \tag{4}$$



(a) Feature Pyramid Construction (b) Typical Channels of feature maps

Figure 2: A feature pyramid and some typical channels from different feature maps.



(a) Inputs (b) RGB (c) C^3 (d) F^3

Figure 3: Feature distance maps defined over raw pixel values, pretrained CNN feature C^3, or our learned feature F^3. Our features produce smoother objective function to facilitate optimization.

where $\mathbb{F} = \{F_i | i = 1 \cdots N_i\}$ are feature pyramids of images $\mathbb{I} = \{I_i | i = 1 \cdots N_i\}$. Similar to the photometric BA, our featuremetric BA considers more pixels than corners or blobs. It has the potential to learn more suitable features for SfM to deal with exposure changes, moving objects, etc.

We aim to learn features suitable for SfM through back-propagation, instead of using pre-trained CNN features for image classification (Czarnowski et al., 2017). Therefore, it is crucial to design a differentiable optimization layer, our BA-Layer, to solve the optimization problem, so that the loss information can be back-propagated. The BA-Layer predicts camera poses $\mathbb{T}$ and dense depth $\mathbb{D}$ during forward pass and back-propagates the loss from $\mathbb{T}$ and $\mathbb{D}$ to the feature pyramids $\mathbb{F}$ for training.

4.1 Overview

As illusrated in Figure 1, our BA-Net takes multiple images as inputs and then feed them to the backbone DRN-54. We use DRN (Yu et al., 2017) as the backbone because it replaces max-pooling with convolution layers and generate smoother feature maps, which is desirable for BA optimization. Note the conventional DRN is memory inefficient due to the high resolution feature maps after dilation convolutions. We replace the dilation convolution with ordinary convolution with strides to address this issue. After DRN, a feature pyramid is then constructed for each input image, which are the inputs for the BA-Layer.

At the same time, the depth basis sub-network generates multiple basis depth maps for the frame I_1, and the final depth is represented as a linear combination of these bases.

Finally, the BA-Layer optimizes camera poses and depths jointly by minimizing the feature error defined in Equation (4). Our differentibale LM algorithm makes the whole pipeline end-to-end trainable.

4.2 Feature Pyramid

The feature pyramid learns a suitable feature for the BA-Layer. Similar to the feature pyramid networks (FPN) for object detection (Lin et al., 2017), we exploit the inherent multi-scale hierarchy of deep convolutional networks to construct feature pyramids. A top-down architecture with lateral connections is applied to propagate richer context information from coarser scales to finer scales. Thus, our feature BA will have a larger convergence radius.

As shown in Figure 2(a), we construct the feature pyramids from the backbone DRN-54 network. We denote the last residual blocks of conv1, conv2, conv3, conv4 in DRN-54 as $\{C^1, C^2, C^3, C^4\}$, with strides $\{1, 2, 4, 8\}$ respectively. We upsample a feature map C^{k+1} by a factor of 2 with bilinear interpolation and concatenate the upsampled map with the feature map C^k in the next level. This procedure is iterated until the finest level. Finally, we apply a 3×3 convolution on the concatenated feature maps to reduce its dimensionality to 128 to balance the expressiveness and computational complexity, which leads to the final feature pyramids $F_i = [F_i^1, F_i^2, F_i^3]$ for image I_i.

We visualize some typical channels, e.g. the pre-trained DRN C^3 and our learned F^3 in Figure 2(b). It is evident that, after training with our BA-Layer, the feature pyramid becomes smoother and each channel correspondences to different regions in the image. Note that our feature pyramids have higher resolution than FPN to facilitate precise alignment.

To have a better intuition about how much the BA optimization benefits from our learned feature, we visualize different feature distances in Figure 3. We evaluate the distance between a pixel marked by a yellow plus in the top image in Figure 3 (a) and all pixels in a neighbourhood of its corresponding point in the bottom image of Figure 3 (a). The distances evaluated from raw RGB value, pretrained feature C^3, and our learned feature F^3 are visualized in (b), (c), and (d) respectively. All distances are normalized to $[0, 1]$ and visualized as a heat map. The x-axis and y-axis there are the offsets to the ground-truth corresponding point. The RGB distance in (b) (i.e. e_p in Equation (3)) has no clear global minimum, which makes the photometric BA sensitive to initialization (Engel et al., 2014; 2017). The distance measured by the pretrained feature C^3 has both global and local minimums. Finally, the distance measured by our learned feature F^3 has a clear global minimum and smooth basin, which is helpful in gradient descent based optimization such as the LM algorithm.

4.3 Bundle Adjustment Layer

After building feature pyramids for all images, we optimize camera poses and a dense depth map by minimizing the feature distance in Equation (4). Following the conventional Bundle Adjustment principle, we optimize Equation (4) using the Levenberg-Marquardt (LM) algorithm. However, the original LM algorithm is non-differentiable for two difficulties:

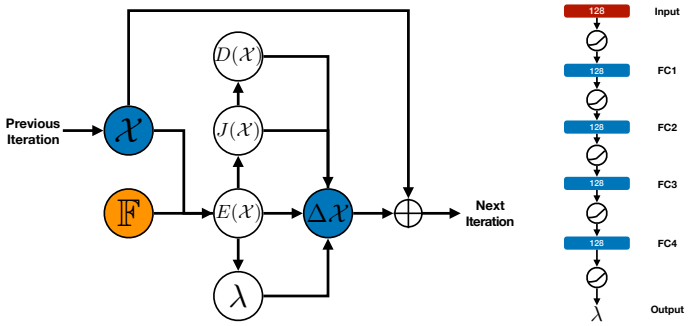
- The iterative computation terminates when a specified convergence threshold is reached. This if-else based termination strategy makes the output solution $\mathcal{X}$ non-differentiable with respect to the input $\mathbb{F}$ Domke (2012).
- In each iteration, it updates the damping factor λ based on the current value of the objective function. It raises λ if a step fails to reduce the objective; otherwise it reduces λ. This if-else also makes $\mathcal{X}$ non-differentiable with respect to $\mathbb{F}$.

When the solution $\mathcal{X}$ is non-differentiable with respect to $\mathbb{F}$, feature learning by back-propagation becomes impossible. The first difficulty has been studied in Domke (2012) and the author proposes to fix the number of iterations, which is refered as 'incomplete optimization'. Besides making the optimization differentiable, this 'incomplete optimization' technique also reduces memory consumption because the number of iterations is usually fixed at a small value.

The second difficulty has never been studied. Previous works mainly focus on gradient descent Domke (2012) or quadratic minimization Amos and Kolter (2017); Schmidt and Roth (2014). In this section, we propose a simple yet effective approach to soften the if-else decision and yields a differentiable LM algorithm. We send the current objective function value to a feed-forward network to predict λ. This technique not only makes the optimization differentiable, but also learns to predict a better damping factor λ, which gaurantees that the optimziation will converged to a better solution within limited iterations.

To start with, we illustrate a single iteration of the LM optimization as a diagram in Figure 4(a) by interpreting intermediate variables as network nodes. During the *forward* pass, we compute the solution update $\Delta\mathcal{X}$ from the feature $\mathbb{F}$ and the damping factor λ as the following:

- According to the result $\mathcal{X}$ of the previous iteration, we compute feature differences $E(\mathcal{X}) = [e^f_{1,1}(\mathcal{X}), e^f_{1,2}(\mathcal{X}) \cdots e^f_{N_i,N_j}(\mathcal{X})]$ in Equation (4) from all N_i images and N_j points;
- We then compute the Jacobian matrix $J(\mathcal{X})$ and the diagonal matrix $D(\mathcal{X})$ from $J(\mathcal{X})^\top J(\mathcal{X})$;



(a) (b)

Figure 4: (a):A single iteration of the differentiable LM and (b): the MLP network to predict damping factor λ.

- To predict the damping factor λ, we use global average pooling to aggregate the aboslute value of $E(\mathcal{X})$ over all pixels for each feature channel, and get a 128D feature vector. We then send it to a MLP sub-network to predict λ;
- Finally, the update $\Delta\mathcal{X}$ to the current solution is computed as a standard Levenberg-Marquardt step:

$$\Delta\mathcal{X} = (J(\mathcal{X})^\top J(\mathcal{X}) + \lambda D(\mathcal{X}))^{-1} J(\mathcal{X})^\top E(\mathcal{X}).$$

In this way, we can consider λ as an intermediate variable and denote each LM step as a function g about features pyramids $\mathbb{F}$ and the solution $\mathcal{X}_{k-1}$ from the previous iteration. In other words, $\Delta\mathcal{X} = g(\mathcal{X}_{k-1}; \mathbb{F})$. Therefore, the solution after the k-th iteration is:

$$\mathcal{X}_k = g(\mathcal{X}_{k-1}; \mathbb{F}) \circ \mathcal{X}_{k-1}. \tag{5}$$

Here, $\circ$ denotes parameters updating, which is addition for depths and SE(3) exponential mapping for camera poses. Clearly, Equation (5) is differentiable with respect to the feature pyramids $\mathbb{F}$, which makes back-propagation possible through the whole pipeline for feature learning.

The MLP sub-network that predicts λ is shown in Figure 4(b). We stack four fully-connected layers to predict λ from the input 128D vector. We use ReLU as the activation function to make sure λ is a non-negative value. Following the photometric BA Engel et al. (2014; 2017), we solve our featuremetric BA using a coarse-to-fine strategy with warping of the feature pyramids. We apply the differentiable LM algorithm for 5 iterations at each feature pyramid level, leading to 15 iterations in total.

4.4 Dense Depth Parameterization

Parameterizing a dense depth map by a per-pixel depth value is impractical under our formulation. Firstly, it introduces too many parameters to be optimized. For example, an image of 320×240 pixels results in 76.8k parameters. Secondly, in the beginning of training, many pixels will become invisible in the other views because of the poorly predicted depth or motion. So little information can be back-propagated to improve the network, which makes training difficult.

To deal with these problems, we use the convolutional network for monocular image depth estimation as a compact parameterization, rather than using it as an initialization like in (Tateno et al., 2017; Yang et al., 2018). We use a standard encoder-decoder architecture for monocular depth learning (Laina et al., 2016). We modifies the last convolutional feature maps to 128D and use them as depth bases for optimization. The depth map is generated by the last convolutional layer of the decoder, which is:

$$\mathbb{D} = \mathrm{ReLU}(\boldsymbol{w}^\top \boldsymbol{B}). \tag{6}$$

Here, $\mathbb{D}$ is the $\times h * w$ depth map that contains depth values for all pixels, $\boldsymbol{B}$ is a $128 \times h * w$ matrix, representing 128 depth map bases generated from network, $\boldsymbol{w}$ is the linear combination weights of these bases. This $\boldsymbol{w}$ will be optimized in our BA-Layer. The ReLU activation function guarantees the depth to be non-negative. Once the depth basis $\boldsymbol{B}$ is generated from the network, we fix $\boldsymbol{B}$ and use $\boldsymbol{w}$ as a compact depth representations in BA optimization. Therefore, the feature-metric difference is:

$$e^f_{i,j}(\mathcal{X}) = F_i(\pi(\boldsymbol{T}_i, \text{ReLU}(\boldsymbol{w}^\top \boldsymbol{B}))) - F_1(\boldsymbol{q}_{1,j}). \tag{7}$$

To further speedup convergence, we learn a initial weight $\boldsymbol{w}_0$ as a 1D convolution filter for an arbitrary image, i.e. $\mathbb{D}_0 = \text{ReLU}(\boldsymbol{w}_0^\top \boldsymbol{B})$.

4.5 Training

The BA-Net learns the feature pyramids, depth bases generator, and the damping factor predictor in a supervised manner. We apply the following commonly used loss for training, though more sophisticated ones might be designed.

Camera Pose Loss The camera rotation loss is the distance between rotation quaternion vectors $\mathcal{L}_{rotation} = \|\boldsymbol{q} - \boldsymbol{q}^*\|$. Similarly, translation loss is the Euclidean distance between prediction and groundtruth in metric scale, $\mathcal{L}_{translation} = \|\boldsymbol{t} - \boldsymbol{t}^*\|$.

Depth Map Loss For each dense depth map we applies the berHu Loss (Zwald and Lambert-Lacroix, 2012) as in Laina et al. (2016).

We initialize the back-bone network from DRN (Yu et al., 2017), and the other components are trained with ADAM Kingma and Ba (2014) from scratch with initial learning rate 0.001, and the learning rate is divided by two when we observe plateaus from the Tensorboard interface.

5 Evaluation

5.1 Dataset

ScanNet ScanNet (Dai et al., 2017a) is a large-scale indoor dataset with 1,513 sequences in 706 different scenes. Camera poses and depth maps are not perfect, because they are estimated via the BundleFusion system (Dai et al., 2017b). The metric scale is known in all data from ScanNet, because the raw data is recorded with a depth camera which returns absolute depth values.

To sample image pairs for training, we apply a simple filtering process. We first filter out pairs with a large photo-consistency error, to avoid image pairs with large pose or depth error. We also filter out image pairs, if less than 50% of the pixels from one image are visible in the other image. In addition, we also discard a pair if their roundness score Beder and Steffen (2006) is less than 0.001, which avoids pairs with too narrow baselines.

We split the whole dataset into training and testing sequences. The training set contains 1,413 sequences and the testing set contains the rest 100 sequences. We sample 547,991 training pairs and 2,000 testing pairs from the training and testing sequences respectively.

KITTI KITTI (Geiger et al., 2012) is a widely used benchmark dataset collected by car-mounted cameras and a LIDAR sensor on streets. It contains 61 scenes belonging to the "city", "residential", or "road" categories. Eigen et al. (Eigen et al., 2014) selects 28 scenes for testing and 28 scenes from the remaining for training. We use the same data split, to make a fair comparison with previous methods.

Since ground truth pose is unavailable from the original KITTI dataset, we compute camera poses by LibVISO2 (Geiger et al., 2011) and take them as ground truth after discarding poses with large errors. To evaluate the camera poses accuracy, we follow (Zhou et al., 2017; Wang et al., 2018) to measure the Absolute Trajectory Error on the 9th and 10th sequences from the KITTI odometry data. In this experiment, we create short sequences of 5 frames by first computing 5 two-view reconstructions from our BA-Net and then align the two-view reconstructions in the coordinate system anchored at the first frame.

5.2 Comparisons with Other Methods

ScanNet To evaluate the results quality, we use the depth error suggested in Eigen and Fergus (2015). The error in camera motion is measured by the rotation error (the minimum rotation to align the ground truth and estimated camera coordinates), the translation direction error (the angle between

	Ours	Ours*	DeMoN*	Photometric BA	Geometric BA
Rotation (degree)	**1.018**	1.587	3.791	4.409	8.56
Translation (cm)	**3.39**	10.81	15.5	21.40	36.995
Translation (degree)	**20.577**	31.005	31.626	34.36	39.392
abs relative difference	**0.161**	0.238	0.231	0.268	0.382
sqr relative difference	**0.092**	0.176	0.520	0.427	1.163
RMSE (linear)	**0.346**	0.488	0.761	0.788	0.876
RMSE (log)	**0.214**	0.279	0.289	0.330	0.366
RMSE (log, scale inv.)	**0.184**	0.276	0.284	0.323	0.357

Table 1: Quantitative Comparisons with DeMoN and classic BA. The superindex * denotes that the model is trained on the trainning set described in Ummenhofer et al. (2017).

	Ours	Wang et al. (2018)	Zhou et al. (2017)	Godard et al. (2017)	Eigen et al. (2014)
ATE(km)	**0.019**	0.045	0.063	N/A	N/A
abs rel	**0.083**	0.151	0.208	0.148	0.203
sqr rel	**0.025**	1.257	1.768	1.344	1.548
RMSE(linear)	**3.640**	5.583	6.856	5.927	6.307
RMSE(log)	**0.134**	0.228	0.283	0.247	0.282

Table 2: Quantitative Comparison on KITTI.

ground truth and estimated camera translation direction), the absolute position error (the distance between ground truth and estimated camera center point). In Table 1, we compare our method with DeMoN (Ummenhofer et al., 2017) and the conventional photometric and geometric BA. Note that we cannot get DeMoN trained on the ScanNet data. For fair comparison, we train our network on the same training data as DeMoN and test both networks on our testing data[1]. We also show the results of our network trained on ScanNet. Our BA-Net consistently performs better than DeMoN no matter which training data is used. Since DeMoN does not recover the absolute scale, we align its depth map with the groundtruth to recover its metric scale for evaluation. We further compare with conventional methods with geometric (Nister, 2004; Agarwal et al.) and photometric (Engel et al., 2014) BA. Again, our method produces better results. The geometric BA works poorly here, because feature matching is difficult in indoor scenes. Even the RANSAC process cannot get rid of all outliers. While for photometirc BA, the highly non-convex objective function is difficult to optimize as described in Section 3.

KITTI Table2 summarizes our results on the KITTI dataset. We use the same metrics as the comparison on ScanNet for depth evaluation. Our method outperforms the supervised methods (Eigen et al., 2014) as well as recent unsupervised methods (Zhou et al., 2017; Wang et al., 2018; Godard et al., 2017). Our method also achieves more accurate camera motion than the methods Zhou et al. (2017); Wang et al. (2018). We believe this is due to our 'feature-metric BA' with features learned specifically for SfM problem, which makes the objective function closer to convex and easier to optimize as discussed in Section 3. In comparison, the methods in Zhou et al. (2017); Wang et al. (2018) minimizes the photometric error.

More comparison with DeMoN, ablation studies, and multi-view SfM (up to 5 views) are reported in the appendix due to page limit.

6 Conclusions and Future Works

This paper presents the BA-Net, a network that explicitly enforces multi-view geometry constraints in terms of feature reprojection error. It optimizes scene depth and camera motion together via bundle adjustment. The whole pipeline is differentiable and thus end-to-end trainable, such that the features are learned from data to facilitate structure-from-motion. The dense depth is parameterized as a linear combination of some bases depth maps generated from the network. Our BA-Net nicely combines domain knowledge (hard-coded multi-view geometry constraint) with machine learning (learned feature representation and bases depth map generator). It outperforms conventional BA and recent deep learning based methods.

[1]More comparison with DeMoN on DeMoN's data is provided in the appendix.

REFERENCES

Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 29–42, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15552-9.

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, October 2011. ISSN 0001-0782. doi: 10.1145/2001269.2001293. URL http://doi.acm.org/10.1145/2001269.2001293.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 136–145. PMLR, 2017.

Christian Beder and Richard Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In Katrin Franke, Klaus-Robert Müller, Bertram Nickolay, and Ralf Schäfer, editors, *Pattern Recognition*, pages 657–666, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-44414-5.

Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam - learning a compact, optimisable representation for dense visual SLAM. *CoRR*, abs/1804.00874, 2018. URL http://arxiv.org/abs/1804.00874.

D.C. Brown. *A Solution to the General Problem of Multiple Station Analytical Stereotriangulation*. RCA Data reducation technical report. D. Brown Associates, Incorporated, 1958. URL https://books.google.ca/books?id=FikPPwAACAAJ.

Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. URL http://arxiv.org/abs/1512.03012.

Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J. Davison. Learning to solve nonlinear least squares for monocular stereo. In *The European Conference on Computer Vision (ECCV)*, September 2018.

J. Czarnowski, S. Leutenegger, and A. J. Davison. Semantic texture for robust dense tracking. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 851–859, Oct 2017. doi: 10.1109/ICCVW.2017.105.

A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, July 2017a. doi: 10.1109/CVPR.2017.261.

Angela Dai, Matthias Niessner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(3), May 2017b. ISSN 0730-0301. doi: 10.1145/3054739. URL http://doi.acm.org/10.1145/3054739.

A. Delaunoy and M. Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, June 2014. doi: 10.1109/CVPR.2014.193.

Justin Domke. Generic methods for optimization-based modeling. In *AISTATS*, 2012.

D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, Dec 2015. doi: 10.1109/ICCV.2015.304.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2366–2374, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969091.

J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2658577.

Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proc. of ECCV*, 2014.

Jakob Engel, Vladyslav C. Usenko, and Daniel Cremers. A photometrically calibrated benchmark for monocular visual odometry. *CoRR*, abs/1607.02555, 2016. URL http://arxiv.org/abs/1607.02555.

Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

Ankur Handa, Michael Bloesch, Viorica Pătrăucean, Simon Stent, John McCormac, and Andrew Davison. *gvnn: Neural Network Library for Geometric Computer Vision*, pages 67–82. Springer International Publishing, Cham, 2016. ISBN 978-3-319-49409-8. doi: 10.1007/978-3-319-49409-8_9. URL https://doi.org/10.1007/978-3-319-49409-8_9.

R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997. ISSN 0162-8828. doi: 10.1109/34.601246.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814 vol. 2, June 2005. doi: 10.1109/CVPR.2005.56.

Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to learn using gradient descent. In *IN LECTURE NOTES ON COMP. SCI. 2130, PROC. INTL. CONF. ON ARTI NEURAL NETWORKS (ICANN-2001*, pages 87–94. Springer, 2001.

Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 577–584, New York, NY, USA, 2005. ACM. doi: 10.1145/1186822.1073232. URL http://doi.acm.org/10.1145/1186822.1073232.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

L'ubor Ladický, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 89–96, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.19. URL https://doi.org/10.1109/CVPR.2014.19.

I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248, Oct 2016. doi: 10.1109/3DV.2016.32.

T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017. doi: 10.1109/CVPR.2017.106.

M. L. A. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1526–1531 Vol. 2, Oct 2005. doi: 10.1109/ICCV.2005.128.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans. Rob.*, 31:1147–1163, 2015.

D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, June 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.17.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

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, May 2009. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.132.

Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 1161–1168, Cambridge, MA, USA, 2005. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2976248.2976394`.

U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781, June 2014. doi: 10.1109/CVPR.2014.349.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

F. Steinbruecker, J. Sturm, and D. Cremers. Real-time visual odometry from dense rgb-d images. In *Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV)*, 2011.

C. Tang, O. Wang, and P. Tan. Gslam: Initialization-robust monocular visual slam via global structure-from-motion. In *2017 Fifth International Conference on 3D Vision (3DV)*, pages 239–248, Oct 2017. doi: 10.1109/3DV.2016.32.

K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6565–6574, July 2017. doi: 10.1109/CVPR.2017.695.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK, 2000a. Springer-Verlag. ISBN 3-540-67973-1. URL `http://dl.acm.org/citation.cfm?id=646271.685629`.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK, 2000b. Springer-Verlag. ISBN 3-540-67973-1. URL `http://dl.acm.org/citation.cfm?id=646271.685629`.

Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Chaoyang Wang, Buenaposada, Miguel Jose, Rui Zhu, , and Simon Lucey. Learning depth from monocular videos using direct methods. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR 2011*, pages 3057–3064, June 2011. doi: 10.1109/CVPR.2011.5995552.

D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 161–169, July 2017. doi: 10.1109/CVPR.2017.25.

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

F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–644, July 2017. doi: 10.1109/CVPR.2017.75.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

L. Zwald and S. Lambert-Lacroix. The BerHu penalty and the grouped effect. *ArXiv e-prints*, jul 2012.

	Ours (Full)	w/o Feature Learning	w/o Joint Optimization	w/o λ Prediction
Rotation (degree)	**1.018**	2.667	1.036	3.176
Translation (cm)	**3.39**	10.8	3.91	12.83
Translation (degree)	**20.577**	31.493	26.779	31.905
abs relative difference	**0.161**	0.267	0.217	0.259
sqr relative difference	**0.092**	0.242	0.145	0.549
RMSE (linear)	**0.346**	0.481	0.428	0.557
RMSE (log)	**0.214**	0.303	0.270	0.345
RMSE (log, scale inv.)	**0.184**	0.226	0.205	0.306

Table 3: Ablation Study Comparisons by Disabling Different Components of BA-Net

APPENDIX A: ABLATION STUDIES

Learned Features vs Pre-trained Features Our learned feature pyramid improves the convexity of the objective function to facilitate the optimization. We compare our learned feature with features pre-trained on ImageNet for classification tasks. As shown in Table 3, the pre-trained feature produces larger error. This proves the discussion in Section 4.2.

Bundle Adjustment Optimization vs SE(3) Pose Estimation Our BA-Layer optimizes depth and camera poses jointly. We compare it to the SE(3) camera pose estimation with fixed depth map (generated by a monocular image depth estimation method), which is adopted in Wang et al. (2018). To make the comparison fair, we also use our learned feature for the SE(3) camera pose estimation. As shown in Table 3, without BA optimization, both the depth maps and camera poses are worse, because the errors in the depth estimation will degrades the camera pose estimation.

Differentiable LM vs Gauss Newton(with fixedλ) To make the whole pipeline end-to-end trainable, we makes the LM algorithm differentiable by learning the damping factor from neural networks. We compare our method with a fixed λ. Again, for fair comparison, we provide our learned feature pyramids to both methods, though it cannot be learned without our differentiable method. As shown in Table 3, our method works much better, because the fixed λ is not optimal for all iterations, and the update could make the solution moves to the wrong direction.

APPENDIX B: EVALUATION ON DEMON DATASET

Table 4 summarizes our results on the DeMoN dataset. For a comparison, we also cite the results from DeMoN Ummenhofer et al. (2017) and the most recent work LS-Net Clark et al. (2018). We further cite the results from some conventional approaches as reported in DeMoN, indicated as **Oracle**, **SIFT**, **FF**, and **Matlab** respectively. Here, **Oracle** uses *ground truth* camera poses to solve the multi-view stereo by SGM (Hirschmuller, 2005), while **SIFT**, **FF**, and **Matlab** further use sparse features, optical flow, and KLT tracking respectively for feature correspondence to solve camera poses by the 8-pt algorithm Hartley (1997).

Our method consistently outperforms DeMoN Ummenhofer et al. (2017) at both camera motion and scene depth, except on the 'Scenes11' data, because we enforce multi-view geometry constraint in the BA-Layer. Our results are poorer on the 'Scene11' dataset, because the images there are synthesized with random objects from the ShapeNet (Chang et al., 2015) without physically correct scale. This setting is inconsistent with real data and makes it harder for our method to learn the depth bases generator.

When compared with LS-Net Clark et al. (2018), our method achieves similar accuracy on camera poses but better scene depth. It proves our featuremetric BA with learned feature is superior than the photometric BA in the LS-Net.

APPENDIX C: MULTI-VIEW STRUCTURE-FROM-MOTION

Our method can be easily extended to reconstruct multiple images. We evaluate our method in the multi-view setting on the ScanNet (Dai et al., 2017a) dataset. To sample multi-view images for

		Depth			Motion	
	Method	L1-inv	sc-inv	L1-rel	Rotation	Translation
MVS	Oracle	0.019	0.197	0.105	0	0
	SIFT	0.056	0.309	0.361	21.180	60.516
	FF	0.055	0.308	0.322	4.834	17.252
	Matlab	-	-	-	10.843	32.736
	DeMoN	0.047	0.202	0.305	5.156	14.447
	LS-Net	0.051	0.221	0.311	4.653	**11.221**
	Ours	**0.03**	**0.15**	**0.08**	**3.499**	11.238

		Depth			Motion	
	Method	L1-inv	sc-inv	L1-rel	Rotation	Translation
Scenes11	Oracle	0.023	0.618	0.349	0	0
	SIFT	0.051	0.900	1.027	6.179	56.650
	FF	0.038	0.793	0.776	1.309	19.425
	Matlab	-	-	-	0.917	14.639
	DeMoN	0.019	0.315	0.248	**0.809**	8.918
	LS-Net	**0.010**	0.410	0.210	0.910	**8.21**
	Ours	0.08	**0.21**	**0.13**	1.298	10.37

		Depth			Motion	
	Method	L1-inv	sc-inv	L1-rel	Rotation	Translation
RGB-D	Oracle	0.026	0.398	0.336	0	0
	SIFT	0.050	0.577	0.703	12.010	56.021
	FF	0.045	0.548	0.613	4.709	46.058
	Matlab	-	-	-	12.831	49.612
	DeMoN	0.028	0.130	0.212	2.641	20.585
	LS-Net	0.019	0.09	0.301	**1.01**	22.1
	Ours	**0.008**	**0.087**	**0.05**	2.459	**14.90**

		Depth			Motion	
	Method	L1-inv	sc-inv	L1-rel	Rotation	Translation
Sun3D	Oracle	0.020	0.241	0.220	0	0
	SIFT	0.029	0.290	0.286	7.702	41.825
	FF	0.029	0.284	0.297	3.681	33.301
	Matlab	-	-	-	5.920	32.298
	DeMoN	0.019	0.114	0.172	1.801	18.811
	LS-Net	**0.015**	0.189	0.650	**1.521**	14.347
	Ours	**0.015**	**0.11**	**0.06**	1.729	**13.26**

Table 4: Quantitative comparisons on the DeMoN datasets.

	Ours(2-views)	Ours(3-views)	Ours(5-views)
Rotation (degree)	1.018	1.013	**1.009**
Translation (cm)	3.391	2.852	**2.365**
Translation (degree)	20.577	16.423	**14.626**
abs relative difference	0.161	0.111	**0.091**
sqr relative difference	0.092	0.087	**0.068**
RMSE (linear)	0.346	0.288	**0.223**
RMSE (log)	0.214	0.179	**0.147**
RMSE (log, scale inv.)	0.184	0.168	**0.137**

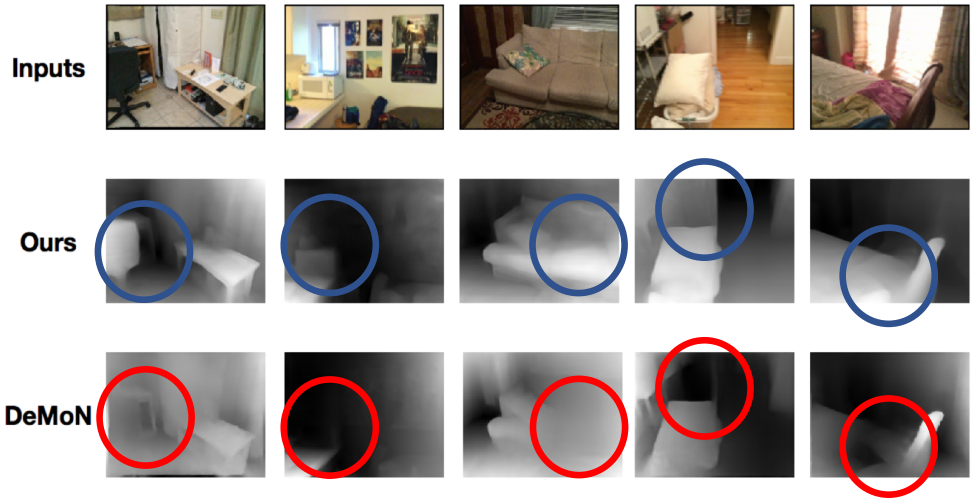
Table 5: Quantitative comparisons on multi-view reconstruction on ScanNet.

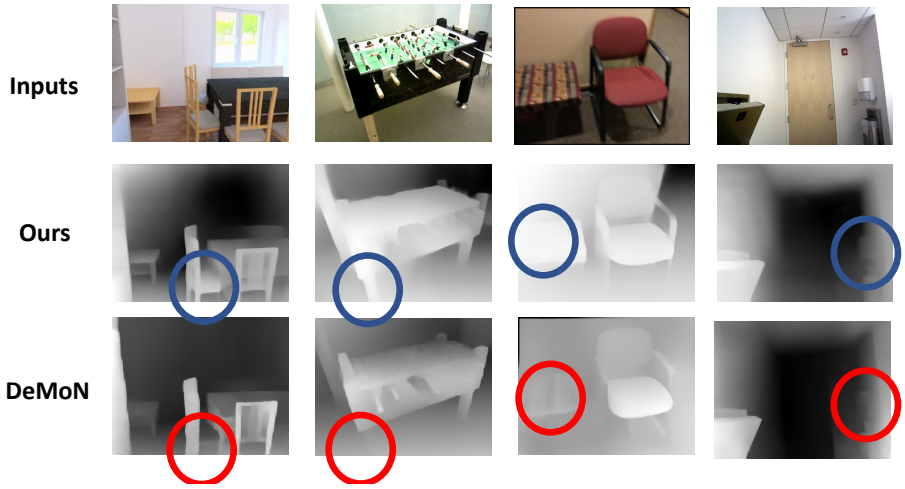
training, we randomly select two-view image pairs that shares a common image to construct N-view sequences. Due to the limited GPU memory (12G), we limit N to 5.

As shown in the Table 5, the accuracy is consistently improved when more views are included, which demonstrated the strength of the multi-view geometry constraint. Instead, most existing deep learning approaches can only handle two views at a time, which is sub-optimal as known in structure-from-motion literature.

APPENDIX D: QUALITATIVE COMPARISONS WITH OTHER METHODS

Finally, we show some qualitative comparison with the previous methods. Figure 5 shows the recovered depth map by our method and DeMoN Ummenhofer et al. (2017) on the ScanNet data. As we can see from the regions highlighted with a red circle, our method recovers more shape details. This is consistent with the quantitative results in Table 1. Figure 6 reflects the results in Table 4. Figure 7 shows the recovered depth maps by our method, Wang et al. (2018), and Godard et al. (2017) respectively. Similarly, we observe more shape details in our results, as reflected in the quantitative results in Table 2.





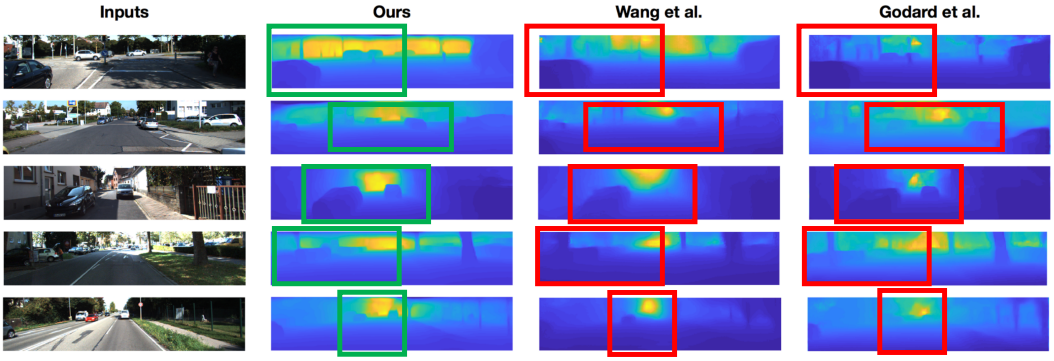


Figure 7: Qualitative Comparisons with Wang et al. (2018) and Godard et al. (2017).