DELTAS: Depth Estimation by Learning Triangulation And densification of Sparse points

Ayan Sinha[1], Zak Murez[1], James Bartolozzi*, Vijay Badrinarayanan[2]*, and Andrew Rabinovich[3]*

[1] Magic Leap Inc., CA, USA {asinha,zmurez}@magicleap.com
*Work done at Magic Leap bartolozzij@gmail.com
[2] Wayve.ai, London, UK vijay@wayve.ai
[3] InsideIQ Inc., CA, USA andrew@insideiq.team

Abstract. Multi-view stereo (MVS) is the golden mean between the accuracy of active depth sensing and the practicality of monocular depth estimation. Cost volume based approaches employing 3D convolutional neural networks (CNNs) have considerably improved the accuracy of MVS systems. However, this accuracy comes at a high computational cost which impedes practical adoption. Distinct from cost volume approaches, we propose an efficient depth estimation approach by first (a) detecting and evaluating descriptors for interest points, then (b) learning to match and triangulate a small set of interest points, and finally (c) densifying this sparse set of 3D points using CNNs. An end-to-end network efficiently performs all three steps within a deep learning framework and trained with intermediate 2D image and 3D geometric supervision, along with depth supervision. Crucially, our first step complements pose estimation using interest point detection and descriptor learning. We demonstrate state-of-the-art results on depth estimation with lower compute for different scene lengths. Furthermore, our method generalizes to newer environments and the descriptors output by our network compare favorably to strong baselines.

Keywords: 3D from Multi-view and Sensors, Stereo Depth Estimation, Multi-task learning

1 Motivation

Depth sensing is crucial for a wide range of applications ranging from Augmented Reality (AR)/ Virtual Reality (VR) to autonomous driving. Estimating depth can be broadly divided into classes: active and passive sensing. Active sensing techniques include LiDAR, structured-light and time-of-flight (ToF) cameras, whereas depth estimation using a monocular camera or stereopsis of an array of cameras is termed passive sensing. Active sensors are currently the de-facto standard of applications requiring depth sensing due to good accuracy and low latency in varied environments [48]. However, active sensors have their own of limitation. LiDARs are prohibitively expensive and provide sparse measurements.

Structured-light and ToF depth cameras have limited range and completeness due to the physics of light transport. Furthermore, they are power hungry and inhibit mobility critical for AR/VR applications on wearables. Consequently, computer vision researchers have pursued passive sensing techniques as a ubiquitous, cost-effective and energy-efficient alternative to active sensors [31].

Passive depth sensing using a stereo cameras requires a large baseline and careful calibration for accurate depth estimation [3]. A large baseline is infeasible for mobile devices like phones and wearables. An alternative is to use MVS techniques for a moving monocular camera to estimate depth. MVS generally refers to the problem of reconstructing 3D scene structure from multiple images with known camera poses and intrinsics [14]. The unconstrained nature of camera motion alleviates the baseline limitation of stereo-rigs, and the algorithm benefits from multiple observations of the same scene from continuously varying viewpoints [17]. However, camera motion also makes depth estimation more challenging relative to rigid stereo-rigs due to pose uncertainty and added complexity of motion artifacts. Most MVS approaches involve building a 3D cost volume, usually with a plane sweep stereo approach [45, 18]. Accurate depth estimation using MVS rely on 3D convolutions on the cost volume, which is both memory as well as computationally expensive, scaling cubically with the resolution. Furthermore, redundant compute is added by ignoring useful image-level properties such as interest points and their descriptors, which are a necessary precursor to camera pose estimation, and hence, any MVS technique. This increases the overall cost and energy requirements for passive sensing.

Passive sensing using a single image is fundamentally unreliable due to scale ambiguity in 2D images. Deep learning based monocular depth estimation approaches formulate the problem as depth regression [10, 11] and have reduced the performance gap to those of active sensors [26, 24], but still far from being practical. Recently, sparse-to-dense depth estimation approaches have been proposed to remove the scale ambiguity and improve robustness of monocular depth estimation [31]. Indeed, recent sparse-to-dense approaches with less than 0.5% depth samples have accuracy comparable to active sensors, with higher range and completeness [6] . However, these approaches assume accurate or seed depth samples from an active sensor which is limiting. The alternative is to use the sparse 3D landmarks output from the best performing algorithms for Simultaneous Localization and Mapping (SLAM) [32] or Visual Inertial Odometry (VIO) [34]. However, using depth evaluated from these sparse landmarks in lieu of depth from active sensors, significantly degrades performance [47]. This is not surprising as the learnt sparse-to-dense network ignores potentially useful cues, structured noise and biases present in SLAM or VIO algorithm.

Here we propose to learn the sparse 3D landmarks in conjunction with the sparse to dense formulation in an end-to-end manner so as to (a) remove dependence on a cost volume in the MVS technique,thus, significantly reducing compute, (b) complement camera pose estimation using sparse VIO or SLAM by reusing detected interest points and descriptors, (c) utilize geometry-based MVS concepts to guide the algorithm and improve the interpretability, and (d) benefit from the accuracy and efficiency of sparse-to-dense techniques. Our network is a multitask model [22], comprised of an encoder-decoder structure composed on two encoders, one for RGB image and one for sparse depth image, and three decoders: one for interest point detection, one for descriptors and one for the dense depth prediction. We also contribute a differentiable module that efficiently triangulates points using geometric priors and forms the critical link between the interest point decoder, descriptor decoder, and the sparse depth encoder enabling end-to-end training.

The rest of the paper is organized as follows. Section 2 discussed related work and Section 3 describes our approach. We perform experimental evaluation in Section 4, and finally conclusions and future work are presented in Section 5.

2 Related Work

Interest point detection and description: Sparse feature based methods are standard for SLAM or VIO techniques due to their high speed and accuracy. The detect-then-describe approach is the most common approach to sparse feature extraction, wherein, interest points are detected and then described for a patch around the point. The descriptor encapsulates higher level information, which are missed by typical low-level interest points such as corners, blobs, etc. Prior to the deep learning revolution, classical systems like SIFT [28] and ORB [38] were ubiquitously used as descriptors for feature matching for low level vision tasks. Deep neural networks directly optimizing for the objective at hand have now replaced these hand engineered features across a wide array of applications. However, such an end-to-end network has remained elusive for SLAM [33] due to the components being non-differentiable. General purpose descriptors learnt by methods such as SuperPoint [9], LIFT [46], GIFT [27] aim to bridge the gap towards differentiable SLAM.

MVS: MVS approaches either directly reconstruct a 3D volume or output a depth map which can be flexibly used for 3D reconstruction or other applications. Methods reconstructing 3D volumes [45, 5] are restricted to small spaces or isolated objects either due to the high memory load of operating in a 3D voxelized space [36, 40], or due to the difficulty of learning point representations in complex environments [35]. Here, we use multi-view images captured in indoor environments for depth estimation due to the versatility of depth map representation. This area has lately seen a lot of progress starting with DeepMVS [18] which proposed a learnt patch matching approach. MVDepthNet [44], and DPSNet [19] build a cost volume for depth estimation. GP-MVSNet [17] built upon MVDepthNet to coherently fuse temporal information using gaussian processes. All these methods utilize the plane sweep algorithm during some stage of depth estimation, resulting in an accuracy vs efficiency trade-off.

Sparse to Dense Depth prediction: Sparse-to-dense depth estimation has recently emerged as a way to supplement active depth sensors due to their range

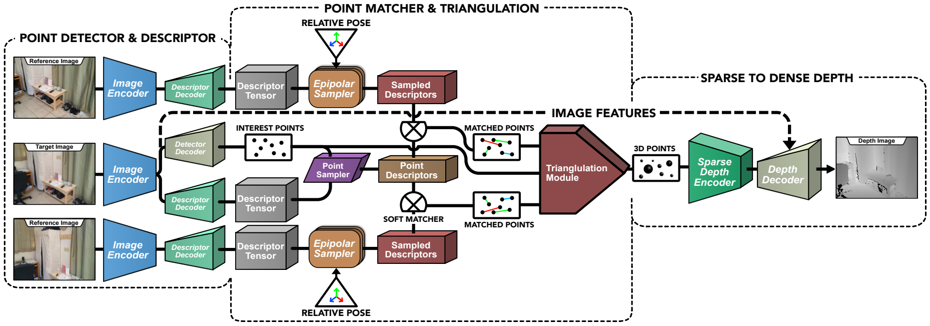

Fig. 1. End-to-end network for detection and description of interest points, matching and triangulation of the points and densification of 3D points for depth estimation.

limitations when operating on a power budget, and to fill in depth in hard to detect regions such as dark or reflective objects. The first such approach was proposed by Ma et.al[31], and following work by Chen et. al. [6] and [47] introduced innovations in the representation and network architecture. A convolutional spatial propagation module is proposed in [7] to in-fill the missing depth values. Self-supervised approaches [13, 12] have concurrently been explored for the sparse-to-dense problem [30]. Recently, a learnable triangulation technique was proposed to learn human pose key-points [21]. We leverage their algebraic triangulation module for the purpose of sparse reconstruction of 3D points.

3 Method

Our method can be broadly sub-divided into three steps as illustrated in Figure 1 for a prototypical target image and two view-points. In the first step, the target or anchor image and the multi-view images are passed through a shared RGB encoder and descriptor decoder to output a descriptor field for each image. Interest points are also detected for the target or the anchor image. In the second step, the interest points in the anchor image in conjunction with the relative poses are used to determine the search space in the reference or auxiliary images from alternate view-points. Descriptors are sampled in the search space and are matched with descriptors for the interest points. Then, the matched keypoints are triangulated using SVD and the output 3D points are used to create a sparse depth image. In the third and final step, the output feature maps for the sparse depth encoder and intermediate feature maps from the RGB encoder are collectively used to inform the depth decoder and output a dense depth image. Each of the three steps are described in greater detail below.

3.1 Interest point detector and descriptor

We adopt SuperPoint-like [9] formulation of a fully-convolutional neural network architecture which operates on a full-resolution image and produces in-

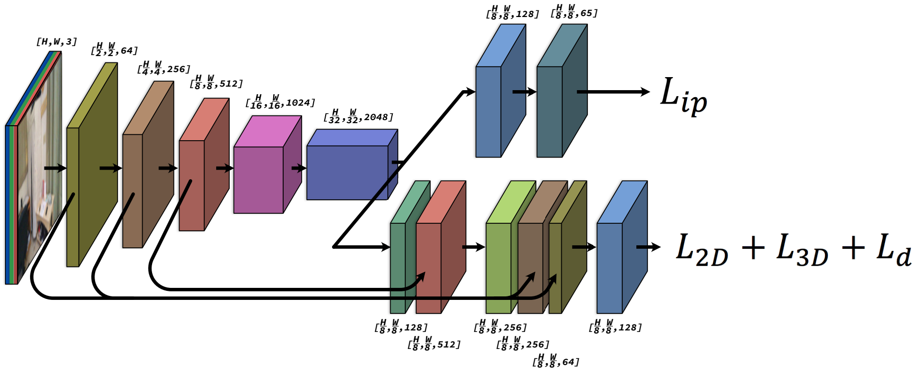

Fig. 2. SuperPoint-like network with detector and descriptor heads.

terest point detection accompanied by fixed length descriptors. The model has a single, shared encoder to process and reduce the input image dimensionality. The feature maps from the encoder feed into two task- specific decoder "heads", which learn weights for interest point detection and interest point description. This joint formulation of interest point detection and description in SuperPoint enables sharing compute for the detection and description tasks, as well as the down stream task of depth estimation. However, SuperPoint was trained on grayscale images with focus on interest point detection and description for continuous pose estimation on high frame rate video streams, and hence, has a relatively shallow encoder. On the contrary, we are interested in image sequences with sufficient baseline, and consequently longer intervals between subsequent frames. Furthermore, SuperPoint's shallow backbone suitable for sparse point analysis has limited capacity for our downstream task of dense depth estimation. Hence, we replace the shallow backbone with a ResNet-50 [16] encoder which balances efficiency and performance. The output resolution of the interest point detector decoder is identical to that of SuperPoint. In order to fuse fine and coarse level image information critical for point matching, we use a U-Net [37] like architecture for the descriptor decoder. This decoder outputs an N-dimensional descriptor tensor at $1/8^{th}$ the image resolution, similar to SuperPoint. The architecture is illustrated in Figure 2. We train the interest point detector network by distilling the output of the original SuperPoint network and the descriptors are trained by the matching formulation described below.

3.2 Point matching and triangulation

The previous step provides interest points for the anchor image and descriptors for all images, i.e., the anchor image and full set of auxiliary images. A naive approach will be to match descriptors of the interest points sampled from the descriptor field of the anchor image to all possible positions in each auxiliary image. However, this is computationally prohibitive. Hence, we invoke geometrical

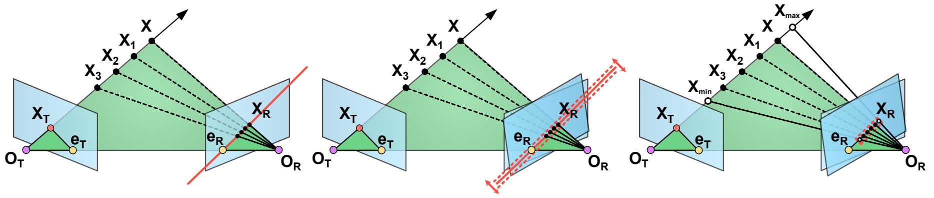

Fig. 3. Left: Epipolar sampling; Middle: Offset sampling due to relative pose error; Right: Constrained depth range sampling

constraints to restrict the search space and improve efficiency. Using concepts from multi-view geometry, we only search along the epipolar line in the auxiliary images [14]. The epipolar line is determined using the fundamental matrix, F, using the relation $xFx^T = 0$, where x is the set of points in the image. The matched point is guaranteed to lie on the epipolar line in an ideal scenario as illustrated in Figure 3 (Left). However, practical limitations to obtain perfect pose lead us to search along the epipolar line with a small fixed offset on either side; Figure 3 (Middle). Furthermore, the epipolar line stretches for depth values from $-\infty$ to ∞. We clamp the epipolar line to lie within feasible depth sensing range, and vary the sampling rate within this restricted range in order to obtain descriptor fields with the same output shape for implementation purposes, shown in Figure 3 (Right). We use bilinear sampling to obtain the descriptors at the desired points in the descriptor field. The descriptor of each interest point is convolved with the descriptor field along its corresponding epipolar line for each image view-point:

$$C_{j,k} = \hat{D}_j * D_j^k, \forall x \in \mathcal{E}, \tag{1}$$

where $\hat{D}$ is the descriptor field of the anchor image, D^k is the descriptor field of the k^{th} auxiliary image, and convolved over all sampled points x along the clamped epipolar line $\mathcal{E}$ for point j. This effectively provides a cross-correlation map [2] between the descriptor field and interest point descriptors. High values in this map indicate potential key-point matches in the auxiliary images to the interest points in the anchor image. In practice, we add batch normalization [20] and ReLU non-linearity [23] to output $C_{j,k}$ in order to ease training.

To obtain the 3D points, we follow the algebraic triangulation approach proposed in [21]. We process each interest point j independently of each other. The approach is built upon triangulating the 2D interest points along with the 2D positions obtained from the peak value in each cross correlation map. To estimate the 2D positions we first compute the softmax across the spatial axes:

$$C^{'}_{j,k} = \exp(C_{j,k})/(\sum_{r_x=1}^{W}\sum_{r_y=1}^{H}\exp(C_{j,k}(r_x, r_y))), \tag{2}$$

where, $C_{j,k}$ indicates the cross-correlation map for the j^{th} inter-point and k^{th} view, and W, H are spatial dimensions of the epipolar search line. Then we calculate the 2D positions of the points as the center of mass of the corresponding cross-correlation maps, also termed soft-argmax operation:

$$x_{j,k} = \sum_{r_x=1}^{W} \sum_{r_y=1}^{H} r(x,y)(C^{'}_{j,k}(r(x,y))). \tag{3}$$

The soft-argmax operation enables differentiable routing between the 2D position of the matched points $x_{j,k}$ and the cross-correlation maps $C_{j,k}$. We use the linear algebraic triangulation approach proposed in [21] to estimate the 3D points from the matched 2D points $x_{j,k}$. Their method reduces the finding of the 3D coordinates of a point z_j to solving the over-determined system of equations on homogeneous 3D coordinate vector of the point $\bar{z}$:

$$A_j \bar{z_j} = 0, \tag{4}$$

where $A_j \in \mathcal{R}^{2k,4}$ is a matrix composed of the components from the full projection matrices and $x_{j,k}$. Different view-points may contribute unequally to the triangulation of a point due to occlusions and motion artifacts. Weighing the contributions equally leads to sub-optimal performance. The problem is solved in a differentiable way by adding weights w_k to the coefficients of the matrix corresponding to different views:

$$(w_j A_j)\bar{z_j} = 0. \tag{5}$$

The weights w are set to be the max value in each cross-correlation map. This allows the contribution of the each camera view to be controlled by the quality of match, and low-confidence matches to be weighted less while triangulating the interest point. Note the confidence value of the interest points are set to be 1. The above equation is solved via differentiable Singular Value Decomposition (SVD) of the matrix $B = UDV^T$, from which $\bar{z}$ is set as the last column of V. The final non-homogeneous value of z is obtained by dividing the homogeneous 3D coordinate vector $\bar{z}$ by its fourth coordinate: $z = \bar{z}/(\bar{z})_4$ [21].

3.3 Densification of sparse depth points

The interest-point detector network provides the 2D position of the points. The z coordinate of the triangulated points provides the depth. We impute a sparse depth image of the same resolution as the input image with depth of these sparse points. Note that the gradients can propagate from the sparse depth image back to the 3D key-points all the way to the input image. This is akin to switch unpooling in SegNet [1]. We pass the sparse depth image through an encoder network which is a narrower version of the image encoder network. Specifically, we use a ResNet-50 encoder with the channel widths after each layer to be $1/4^{th}$ of the image encoder. We concatenate these features with the features obtained from the image encoder. We use a U-net style decoder with intermediate feature maps from both the image as well as sparse depth encoder concatenated with

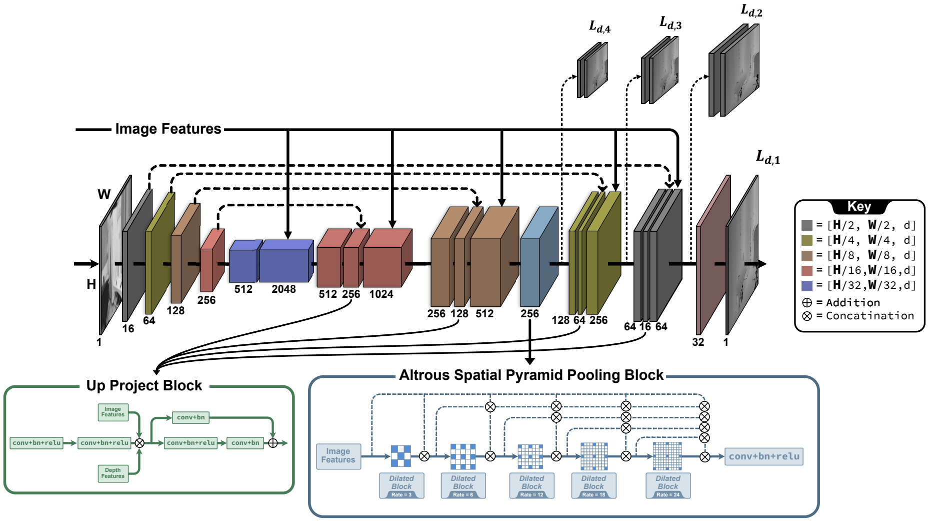

Fig. 4. Proposed sparse-to-dense network architecture showing the concatenation of image and sparse depth features. We use deep supervision over 4 image scales. The blocks below illustrate the upsampling and the altrous spatial pyramid pooling (ASPP) block.

the intermediate feature maps of the same resolution in the decoder, similar to [6]. We provide deep supervision over 4 scales [25]. We also include a spatial pyramid pooling block to encourage feature mixing at different receptive field sizes [15, 4]. The details of the architecture are shown in the Figure 4.

3.4 Overall training objective

The entire network is trained with a combination of (a) cross entropy loss between the output tensor of the interest point detector decoder and ground truth interest point locations obtained from SuperPoint, (b) a smooth-L1 loss between the 2D points output after soft argmax and ground truth 2D point matches, (c) a smooth-L1 loss between the 3D points output after SVD triangulation and ground truth 3D points, (d) an edge aware smoothness loss on the output dense depth map, and (e) a smooth-L1 loss over multiple scales between the predicted dense depth map output and ground truth 3D depth map. The overall training objective is:

$$L = w_{ip}L_{ip} + w_{2d}L_{2d} + w_{3d}L_{3d} + w_{sm}L_{sm} + \sum_{i} w_{d,i}L_{d,i}, \tag{6}$$

where L_{ip} is the interest point detection loss, L_{2d} is the 2D matching loss, L_{3d} is the 3D triangulation loss, L_{sm} is the smoothness loss, and $L_{d,i}$ is the depth estimation loss at scale i for 4 different scales ranging from original image resolution to $1/16^{th}$ the image resolution.

4 Experimental Results

4.1 Implementation Details

Training: Most MVS approaches are trained on the DEMON dataset [43]. However, the DEMON dataset mostly contains pairs of images with the associated depth and pose information. Relative confidence estimation is crucial to accurate triangulation in our algorithm, and needs sequences of length three or greater in order to estimate the confidence accurately and holistically triangulate an interest point. Hence, we diverge from traditional datasets for MVS depth estimation, and instead use ScanNet [8]. ScanNet is an RGB-D video dataset containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations. Three views from a scan at a fixed interval of 20 frames along with the pose and depth information forms a training data point in our method. The target frame is passed through SuperPoint in order to detect interest points, which are then distilled using the loss L_{ip} while training our network. We use the depth images to determine ground truth 2D matches, and unproject the depth to determine the ground truth 3D points. We train our model for 100K iterations using PyTorch framework with batch-size of 24 and ADAM optimizer with learning rate 0.0001 ($\beta_1 = 0.9$, $\beta_2 = 0.999$), which takes about 3 days across 4 Nvidia Titan RTX GPUs. . We fix the resolution of the image to be qVGA (240×320) and number of interest points to be 512 in each image with at most half the interest points chosen from the interest point detector thresholded at 0.0005, and the rest of the points chosen randomly from the image. Choosing random points ensures uniform distribution of sparse points in the image and helps the densification process. We set the length of the sampled descriptors along the epipolar line to be 100, albeit, we found that the matching is robust even for lengths as small as 25. We set the range of depth estimation to be between 0.5 and 10 meters, as common for indoor environments. We empirically set the weights to be [0.1,1.0,2.0,1.0,2.0] for $w_{ip}, w_{2d}, w_{3d}, w_{sm}, w_{d,1}$, respectively. We damp $w_{d,1}$ by a factor of 0.7 for each subsequent scale.

Evaluation: The ScanNet test set consists of 100 scans of unique scenes different for the 707 scenes in the training dataset. We first evaluate the performance of our detector and descriptor decoder for the purpose of pose estimation on ScanNet. We use the evaluation protocol and metrics proposed in SuperPoint, namely the mean localization error (MLE), the matching score (MScore), repeatability (Rep) and the fraction of correct pose estimated using descriptor matches and PnP algorithm at 5° threshold for rotation and and 5 cm for translation. We compare against SuperPoint, SIFT, ORB and SURF at a NMS threshold of 3 pixels for Rep, MLE, and MScore as suggested in the SuperPoint paper. Next, we use standard metrics to quantitatively measure the quality of our estimated depth: : absolute relative error (Abs Rel), absolute difference error (Abs diff), square relative error (Sq Rel), root mean square error and its log scale (RMSE and RMSE log) and inlier ratios ($\delta < 1.25^i$ where $i \in 1, 2, 3$). Note higher values for inlier ratios are desirable, whereas all other metrics warrant lower values.

We compare our method to recent deep learning approaches for MVS: (a) DPSNet: Deep plane sweep approach, (b) MVDepthNet: Multi-view depth net, and (c) GPMVSNet temporal non-parametric fusion approach using Gaussian processes. Note that these methods perform much better than traditional geometry-based stereo algorithms. Our primary results are on sequences of length 3, but we also report numbers on sequences of length 2,4,5 and 7 in order to understand the performance as a function of scene length. We evaluate the methods on Sun3D dataset, in order to understand the generalization of our approach to other indoor scenes. We also discuss the multiply-accumuate operations (MACs) for the different methods to understand the operating efficiency at run-time.

4.2 Detector and Descriptor Quality

Table 1 shows the results of the our detector and descriptor evaluation. Note that MLE and repeatability are detector metrics, MScore is a descriptor metric, and rotation@5° and translation@5cm are combined metrics. We set the threshold for our detector at 0.0005, the same as that used during training. This results in a large number of interest points being detected (Num) which artificially inflates the repeatability score (Rep) in our favour, but has poor localization performance as indicated by MLE metric. However, our MScore is comparable to SuperPoint although we trained our network to only match along the epipolar line, and not for the full image. Furthermore, we have the best rotation@5° and translation@5cm metric indicating that the matches found using our descriptors help accurately determine rotation and translation, i.e., pose. These results are indicative that our training procedure can complement the homographic adaptation technique of SuperPoint and boost the overall performance. Incorporation of evaluated pose using ideas discussed in [39], in lieu of ground truth pose to train our network is left for future work.

Table 1. Performance of different descriptors on ScanNet.

	MLE	**MScore**	**Num**	**Rep**	**rot@5°**	**trans@5cm**
ORB	2.584	0.194	401	0.613	0.142	0.064
SIFT	**2.327**	0.201	203	0.496	0.311	0.148
SURF	2.577	0.198	268	0.460	0.303	0.134
SuperPoint	2.545	**0.375**	129	0.519	0.489	0.244
Ours	3.101	0.329	1511	**0.738**	**0.518**	**0.254**

4.3 Depth Results

We set the same hyper-parameters for evaluating our network for all scenarios and across all datasets, i.e., fix the number of points detected to be 512, length

Table 2. Performance of depth estimation on ScanNet. We use sequences of length 3 and sample every 20 frames. FT indicates fine-tuned on ScanNet.

	Abs Rel	Abs	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
GPMVS	0.1306	0.2600	0.0944	0.3451	0.1881	0.8481	0.9462	0.9753
GPMVS-FT	0.1079	0.2255	0.0960	0.4659	0.1998	0.8905	0.9591	0.9789
MVDepth	0.1191	0.2096	0.0910	0.3048	0.1597	0.8690	0.9599	0.9851
MVDepth-FT	0.1054	0.1911	0.0970	0.3053	0.1553	0.8952	0.9707	0.9895
DPS	0.1470	0.2248	0.1035	0.3468	0.1952	0.8486	0.9474	0.9761
DPS-FT	0.1025	0.1675	0.0574	0.2679	0.1531	0.9102	0.9708	0.9872
Ours	**0.0932**	**0.1540**	**0.0506**	**0.2505**	**0.1426**	**0.9287**	**0.9767**	**0.9893**

of the sampled descriptors to be 100, and the detector threshold to be 5e-4. In order to ensure uniform distribution of the interest points and avoid clusters, we set a high NMS value of 9 as suggested in [9]. The supplement has analysis of the sparse depth output from our network and ablation study over different choices of hyper parameters. Table 2 shows the performance of depth estimation on sequences of length 3 and gap 20 as used in the training set. For fair comparison, we evaluate two versions of the competing approaches (1) The author provided open source trained model, (2) The trained model fine-tuned on ScanNet for 100K iterations with the default training parameters as suggested in the manuscript or made available by the authors. We use a gap of 20 frames to train each network, similar to ours. The fine-tuned models are indicated by the suffix *FT* in the table. Unsurprisingly, the fine-tuned models fare much better than the original models on ScanNet evaluation. MVDepthNet has least improvement after fine-tuning, which can be attributed to the heavy geometric and photometric augmentation used during training, hence making it generalize well. DPSNet benefits maximally from fine-tuning with over 25% drop in absolute error. However, our network outperforms all methods across all metrics. Figure 6 shows qualitative comparison between the different methods and Figure 5 show sample 3D reconstructions of the scene from the estimated depth maps. In Figure 6, we see that MVDepthNet has gridding artifacts, which are removed by GPMVSNet. However, GPMVSNet has poor metric performance. DPSNet washes away finer details and also suffers from gridding artifacts. Our method preserves finer details while maintaining global coherence compared to all other methods. As we use geometry to estimate sparse depth, and the network in-fills the missing values, we retain metric performance while leveraging the generative ability of CNNs with sparse priors. In Figure 5 we see our method consistently output less noisy scene reconstructions compared to MVDepthNet and DPSNet. Moreover, we see planes and corners being respected better than the other methods.

An important feature of any multiview stereo method is the ability to improve with more views. Table 3 shows the performance for different number of images. We set the frame gap to be 20, 15, 12 and 10 for 2,4,5 and 7 frames respectively. These gaps ensure that each set approximately span similar volumes in 3D space, and any performance improvement emerges from the network better using the available information as opposed to acquiring new information. We again see

Table 3. Performance of depth estimation on ScanNet. Results on sequences of various lengths are presented. GPN: GPMVSNet, MVN: MVDepthNet, DPS: DPSNet. AbR: Abolute Relative, Abs: Absolute difference, SqR: Square Relative.

Method	**2 Frames**			**4 Frames**			**5 Frames**			**7 Frames**		
	AbR	Abs	SqR	AbR	Abs	SqR	AbR	Abs	SqR	AbR	Abs	SqR
GPN	0.112	0.233	0.101	0.109	0.226	0.100	0.107	0.226	0.112	0.109	0.230	0.116
MVN	0.126	0.238	0.471	0.105	0.191	0.078	0.106	0.192	0.071	0.108	0.195	0.067
DPS	**0.099**	0.181	0.062	0.102	0.168	0.057	0.102	0.168	0.057	0.102	0.167	0.057
Ours	0.106	**0.173**	**0.057**	**0.090**	**0.150**	**0.049**	**0.088**	**0.147**	**0.048**	**0.087**	**0.144**	**0.043**

Table 4. Performance of depth estimation on Sun3D. We use sequences of length 2.

	Abs Rel	Abs	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
MVDepth	0.1377	0.3199	0.1564	0.4523	0.1853	0.8245	0.9601	0.9851
MVDepth-FT	0.3092	0.7209	4.4899	1.718	0.319	0.7873	0.9117	0.9387
DPS	0.1590	0.3341	0.1564	0.4516	0.1958	0.8087	0.9363	0.9787
DPS-FT	0.1274	0.2858	0.0855	0.3815	0.1768	0.8396	0.9459	0.9866
Ours	**0.1245**	**0.2662**	**0.0741**	**0.3602**	**0.1666**	**0.8551**	**0.9728**	**0.9902**

that our method outperforms all other methods on all three metrics for different sequence lengths. Closer inspection of the values indicate that the DPSNet and GPMVSNet do not benefit from additional views, whereas, MVDepthNet benefits from a small number of additional views but stagnates for more than 4 frames. On the contrary, we show steady improvement in all three metrics with additional views. This can be attributed to our point matcher and triangulation module which naturally benefits from additional views.

As a final experiment, we test our network on Sun3D test dataset consisting of 80 pairs of images. Sun3D also captures indoor environments, albeit at a much smaller scale compared to ScanNet. Table 4 shows the performance for the two versions of DPSNet and MVDepthNet discussed previously, and our network. Note DPSNet and MVDepthNet were originally trained on the Sun3D training database. The fine-tuned version of DPSNet performs better than the original network on the Sun3D test set owing to the greater diversity in ScanNet training database. MVDepthNet on the contrary performs worse, indicating that it overfit to ScanNet and the original network was sufficiently trained and generalized well. Remarkably, we again outperform both methods although our trained network has never seen any image from the Sun3D database. This indicates that our principled way of determining sparse depth, and then densifying has good generalizability. The supplement shows additional qualitative results.

We evaluate the total number of multiply-accumulate operations (MACs) needed for our approach. For a 2 image sequence, we perform 16.57 Giga Macs (GMacs) for the point detector and descriptor module, less than 0.002 GMacs for the matcher and triangulation module, and 67.90 GMacs for the sparse-to-dense module. A large fraction of this is due to the U-Net style feature tensors connecting the image and sparse depth encoder to the decoder. We perform a total of 84.48 GMacs to estimate the depth for a 2 image sequence. This is considerably lower than DPSNet which performs 295.63 GMacs for a 2 image sequence, and also less than the real-time MVDepthNet which performs 134.8 GMacs for a pair of images to estimate depth. It takes 90 milliseconds to estimate depth on Nvidia Titan RTX GPU, which we evaluated to be 2.5 times faster than DPSNet. Inference time for MVDepthNet and GPMVSNet is ≈ 60 milliseconds. We believe our method can be further sped up by replacing Pytorch's native SVD with a custom implementation for triangulation. Furthermore, as we do not depend on a cost volume, compound scaling laws as those derived for image [41] and object [42] recognition can be straightforwardly extended to our method.

5 Conclusion

In this work we developed an efficient depth estimation algorithm by learning to triangulate and densify sparse points in a multi-view stereo scenario. On all of the existing benchmarks, we have exceeded the state-of-the-art results, and demonstrated computation efficiency over competitive methods. In future work, we will expand on incorporating more effective attention mechanisms for interest point matching, and more anchor supporting view selection. Jointly learning depth and the full scene holistically using truncated signed distance function (TSDF) or similar representations is another promising direction. Video depth estimation approaches such as [29] are closely related to MVS, and our approach can be readily extended to predict consistent and efficient depth for videos. Finally, we look forward to deeper integration with the SLAM problem, as depth estimation and SLAM are duals of each other. Overall, we believe that our approach of coupling geometry with the power of conventional 2D CNNs is a promising direction for learning 3D Vision.

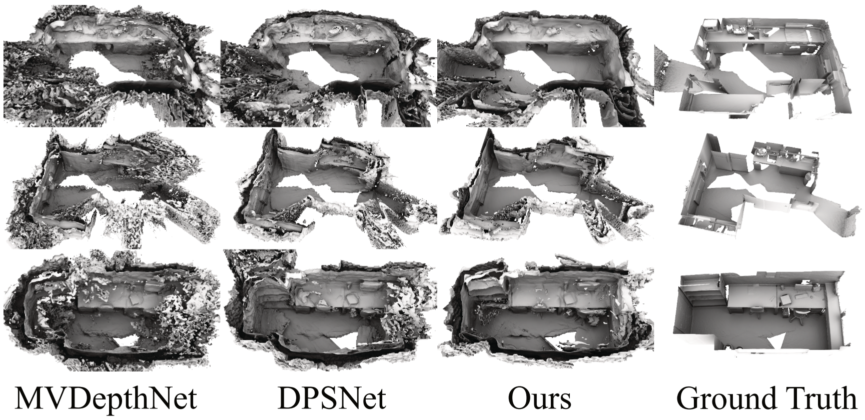

Fig. 5. 3D scene reconstruction using predicted depth over the full sequence.

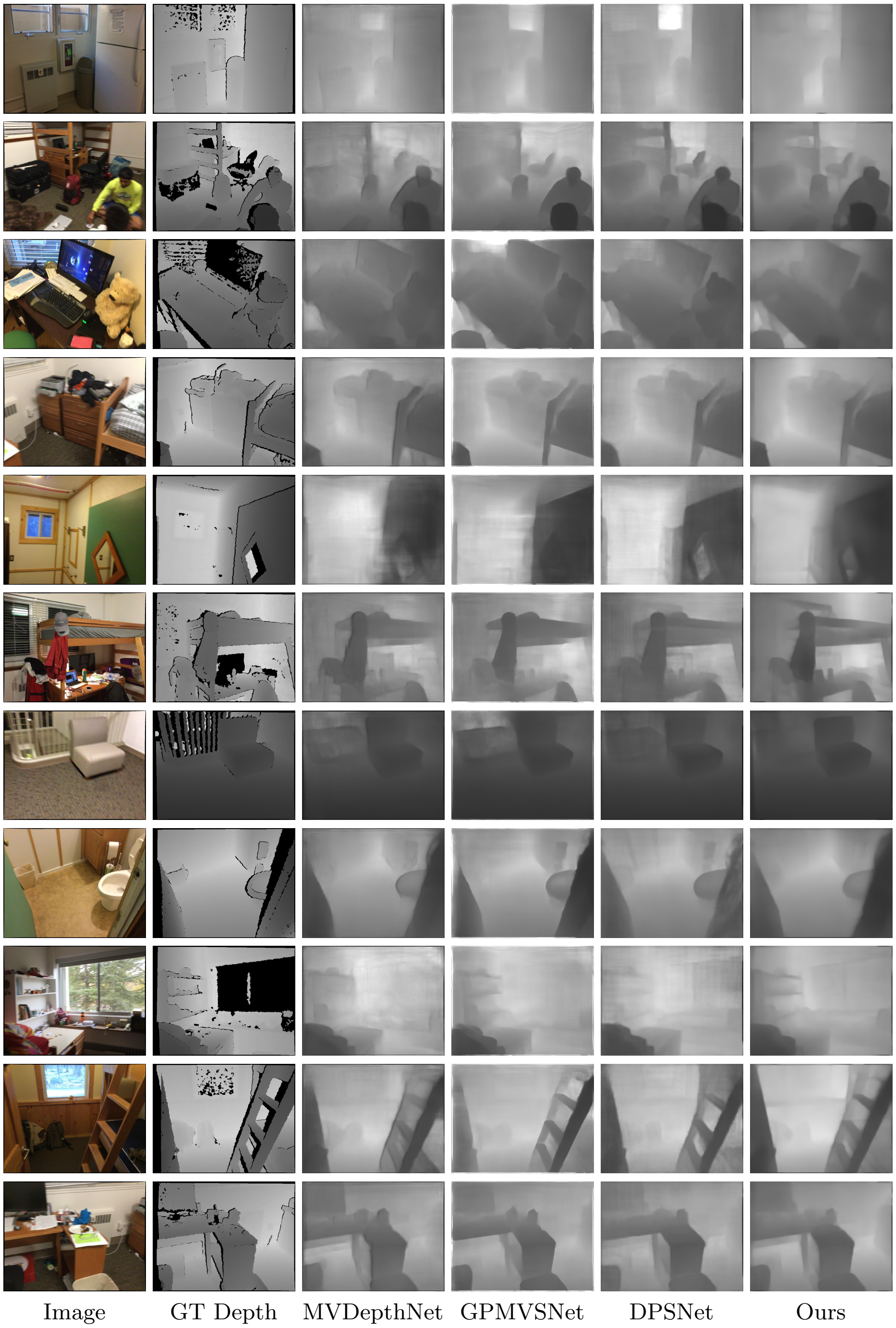

Fig. 6. Qualitative Performance of our networks on sampled images from ScanNet.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation (2015)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016)
3. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5410–5418 (2018)
4. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
5. Chen, R., Han, S., Xu, J., Su, H.: Point-based multi-view stereo network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1538–1547 (2019)
6. Chen, Z., Badrinarayanan, V., Drozdov, G., Rabinovich, A.: Estimating depth from rgb and sparse sensing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 167–182 (2018)
7. Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 103–119 (2018)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 337–33712 (June 2018). https://doi.org/10.1109/CVPRW.2018.00060
10. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision. pp. 2650–2658 (2015)
11. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2002–2011 (2018)
12. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision. pp. 740–756. Springer (2016)
13. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 270–279 (2017)
14. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
15. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Hou, Y., Kannala, J., Solin, A.: Multi-view stereo by temporal nonparametric fusion. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2651–2660 (2019)

18. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2821–2830 (2018)
19. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: End-to-end deep plane sweep stereo. In: 7th International Conference on Learning Representations, ICLR 2019. International Conference on Learning Representations, ICLR (2019)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
21. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7718–7727 (2019)
22. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7482–7491 (2018)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
24. Lasinger, K., Ranftl, R., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv preprint arXiv:1907.01341 (2019)
25. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial intelligence and statistics. pp. 562–570 (2015)
26. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 (2019)
27. Liu, Y., Shen, Z., Lin, Z., Peng, S., Bao, H., Zhou, X.: Gift: Learning transformation-invariant dense visual descriptors via group cnns. In: Advances in Neural Information Processing Systems. pp. 6990–7001 (2019)
28. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
29. Luo, X., Huang, J., Szeliski, R., Matzen, K., Kopf, J.: Consistent video depth estimation **39**(4) (2020)
30. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3288–3295. IEEE (2019)
31. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image (2018)
32. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
33. Murthy Jatavallabhula, K., Iyer, G., Paull, L.: gradslam: Dense slam meets automatic differentiation. arXiv preprint arXiv:1910.10672 (2019)
34. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–I. Ieee (2004)
35. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
36. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3577–3586 (2017)

37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
38. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011)
39. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. arXiv preprint arXiv:1911.11763 (2019)
40. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6040–6049 (2017)
41. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
42. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. arXiv preprint arXiv:1911.09070 (2019)
43. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5038–5047 (2017)
44. Wang, K., Shen, S.: Mvdepthnet: real-time multiview depth estimation neural network. In: 2018 International Conference on 3D Vision (3DV). pp. 248–257. IEEE (2018)
45. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 767–783 (2018)
46. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision. pp. 467–483. Springer (2016)
47. Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 175–185 (2018)
48. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE multimedia **19**(2), 4–10 (2012)