DepthInSpace: Exploitation and Fusion of Multiple Video Frames for Structured-Light Depth Estimation

Mohammad Mahdi Johari
Idiap Reserach Institute, EPFL
mohammad.johari@idiap.ch

Camilla Carta
ams OSRAM
camilla.carta@ams.com

François Fleuret
University of Geneva, EPFL
francois.fleuret@unige.ch

Abstract

We present DepthInSpace, a self-supervised deep-learning method for depth estimation using a structured-light camera. The design of this method is motivated by the commercial use case of embedded depth sensors in nowadays smartphones. We first propose to use estimated optical flow from ambient information of multiple video frames as a complementary guide for training a single-frame depth estimation network, helping to preserve edges and reduce over-smoothing issues. Utilizing optical flow, we also propose to fuse the data of multiple video frames to get a more accurate depth map. In particular, fused depth maps are more robust in occluded areas and incur less in flying pixels artifacts. We finally demonstrate that these more precise fused depth maps can be used as self-supervision for fine-tuning a single-frame depth estimation network to improve its performance. Our models' effectiveness is evaluated and compared with state-of-the-art models on both synthetic and our newly introduced real datasets. The implementation code, training procedure, and both synthetic and captured real datasets are available at https://www.idiap.ch/paper/depthinspace.

1. Introduction

With the advent of structured-light cameras, depth-sensing became conceivable with basic algorithms implementable on devices with computational constraints in real-time. For instance, Kinect V1 uses a correlation-based block matching technique [36], and Intel RealSense [22] employs a semi-global matching scheme [16]. However, learning-based approaches in this field are relatively limited. Fanello *et al*. [35] propose a computationally efficient feature matching method. Projecting image patches to compact binary representation is proposed in UltraStereo [10] to achieve a low complex matching scheme. HyperDepth [34] casts the problem of depth estimation as a classification-regression task, which it solves using an ensemble of cascaded random forests. However, HyperDepth assumes the availability of ground-truth labels either from high-accuracy sensors or exhaustive stereo-matching search algorithms.

Due to the lack of large-scale, precise ground-truth data, an end-to-end training of a deep neural network in a self-supervised manner has been at the center of attention recently. ActiveStereoNet [49] uses Siamese networks for predicting disparity and proposes a novel photometric loss function based on a Local Contrast Normalization (LCN) scheme for training. A separate color sensor is used in [24] to enhance the performance of [49]. Riegler *et al*. [33] exploit the photometric loss function of [49] and propose an edge-detection network along with an edge-aware smoothness loss function to overcome the issue of edge fattening. They also introduce another loss function that leverages the information of other video frames to supervise the disparity estimation network's training. To do so, they use the estimated disparity and camera pose parameters to transform pixels into a 3D point cloud and apply the consistency of predicted depth of matched pixels across multiple frames.

We take the work in [33] as the baseline, and our contributions in this article are as follows:

- We propose a novel training scheme that uses optical flow predictions from ambient images to find matched pixels independently of the estimated disparities, which stabilizes the training and enhances accuracy. Our sensor can capture ambient images conveniently, and we exploit this feature in this regard.
- We extend this model to fuse information from multiple video frames to obtain more precise disparity maps with sharper edges and fewer artifacts.
- We finally propose to exploit the resulting fused disparity maps to fine-tune a single-frame disparity estimation network.

2. Related Works

Active Depth Estimation: The setup usually consists of a camera and a projector which projects a random but known pattern of dots into the scene. Dependent on the depth of objects in the environment, the camera receives a deformed shape of the projected pattern, and this phenomenon could be used in depth estimation algorithms. Such algorithms include basic searching for correspondences in Kinect V1 [28], computationally efficient learning-based techniques [10, 34, 7], and a deep neural network trained end-to-end to estimate disparity map directly [49, 24, 33].

Leveraging Multiple Frames: Utilizing multiple frames for depth estimation includes but is not limited to to structured-light sensors [33]. In [11, 45, 25], the second image of a stereo camera is regarded as another video frame. Explicit utilization of multiple video frames of a conventional camera for self-supervision is proposed in [48, 50, 2, 12, 13, 31, 5]. Fusing the information of multiple frames during inference is employed in RGB depth estimation models like DeepV2D [38], DeepMVS [17], DeepSFM [42], and DPSNet [20] in the form of aggregating volume cost representations. In these papers, the aggregation is done by simple pooling operations (DeepV2D and DeepMVS) or performing convolution on the 2D grid (DeepSFM and DPSNet). Such approaches would fail in the context of structured-light images, where the projector also moves with the camera. As a result of the moving projector, the scene is textured with the projected dots differently, and the camera captures an entirely new scene at each frame. Simply warping frames together and aggregating on the 2D grid will limit the performance since the dots information is meaningless in the warped frames and interferes with the fusion process. We tackle this issue in Section 3.2, where we perform fusion and convolution in the continuous 3D space to leverage the consistency of geometry there maximally. Unfortunately, all the aforementioned models are designed to work with RGB images, and we cannot evaluate them for structured-light images through experiments. However, we examine how the aggregation of frames on the 2D grid would fail for these images in the supplementary material.

Optical Flow and Depth Estimation: Numerous researches in passive depth estimation suggest taking advantage of consistency between optical flow prediction and camera ego-motion between consecutive video frames. The authors in [41, 47, 51, 32] claim that simultaneously training an optical flow network and a depth estimation network can benefit both tasks and result in a better performance than training those individually. The work in [27] proposes a novel framework capable of fine-tuning a general monocular depth estimation network during test time by leveraging a pre-trained optical flow estimation network. Although it is not common in the context of active stereo depth sensing, there is adequate ambient information in captured images to exploit and predict optical flow between frames and improve the quality of depth estimation accordingly.

Convolution in Point Cloud: In the context of point cloud processing, some novel techniques are proposed that perform convolution on points in the continuous 3D space resembling convolutional neural networks of regular grid structures. Models in [39, 26, 46, 43, 3, 40] are shown to be capable of applying convolution on unstructured and unordered data and work well on point cloud benchmark tasks and datasets. For 2D grid-style data, when depth information is available, it is plausible to transform points into the 3D space and leverage such continuous convolutions. Such an approach is presented in [9], where the authors jointly benefit from conventional 2D convolution and parametric continuous convolution introduced in [40].

3. Method

We build DepthInSpace (DIS) model upon the Connecting the Dots (CTD) model in [33]. CTD suggests using two separate networks, one for estimating the disparity, and the other for detecting the edges in the images. The edge detector is weakly supervised with the ambient images, which are the same as dot images except that the projector is off during photo capture. Obtaining ambient data is considerably cheaper than the ground-truth depth data; however, the edge detection network is proposed to reduce the number of ambient images required for training.

We claim ambient images contain more valuable information than only the objects' edges. The sensor that we use is equipped with a programmable switch that can capture both dot images and ambient images with no additional cost. Accordingly, we discard the edge detection network and replace the CTD's smoothing loss function with a loss that directly extracts edges from ambient images. Also, we predict the optical flow from ambient images to find the matched pixels and introduce a new loss which encourages geometric consistency between them. Our proposed loss replaces the geometric loss in CTD and is preferable in two regards. First, CTD uses the momentary predicted depth and ego-motion of the camera to find the matched pixels. As a result, the optimization landscape changes rapidly during training and could result in instability of training. Secondly, the error in momentary predicted depths participates in the procedure of finding matched pixels and leads to degraded performance. In addition, the matching scheme with optical flow provides more flexibility to detect mistakenly matched pixels and exclude them from contributing to the loss function. We use LiteFlowNet [18] pre-trained on MPI Sintel [4] for optical flow, which is a lightweight and fast model, but it has comparable performance to computational and memory resource expensive models like FlowNet2 [19].

3.1. Single-Frame Disparity Estimation

Our DepthInSpace Single-Frame (DIS-SF) model takes the CTD model [33] as a baseline and modifies two of its

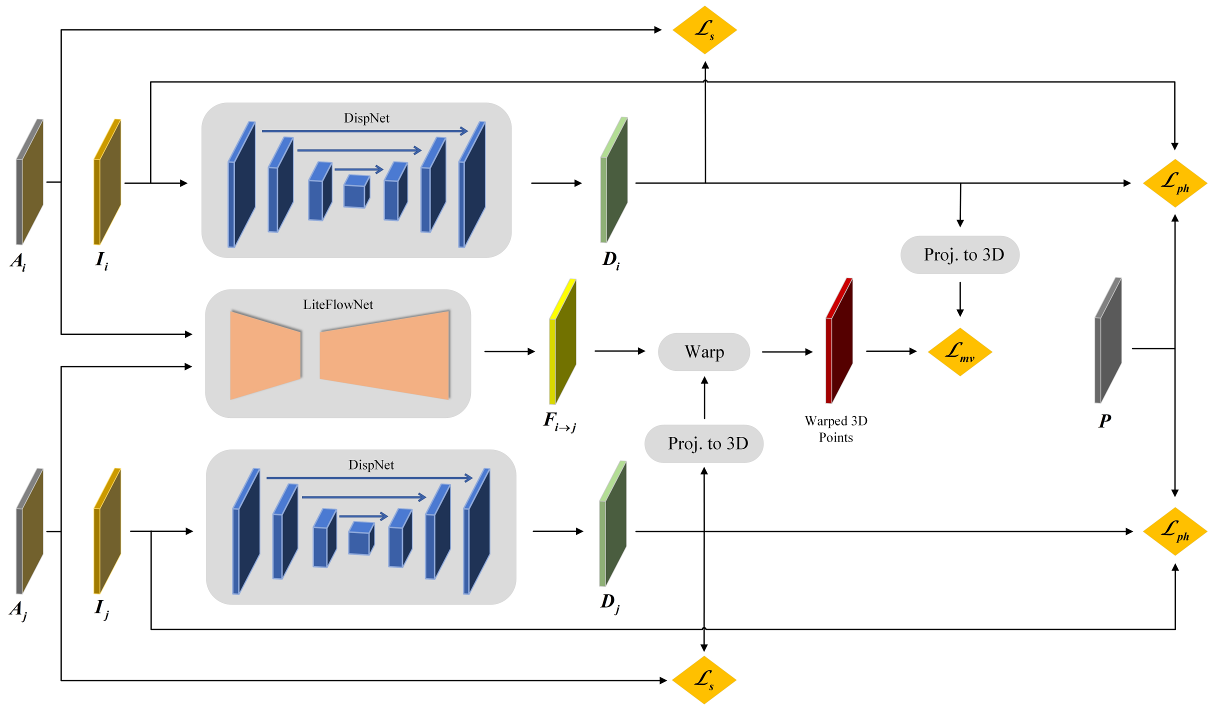

Figure 1. The training scheme of our DIS-SF model for a sample pair of frames i and j, and a reference pattern $\boldsymbol{P}$. The dot images $\boldsymbol{I_i}$ and $\boldsymbol{I_j}$ are fed to the DispNet [29] separately to predict disparities $\boldsymbol{D_i}$ and $\boldsymbol{D_j}$. On another path, LiteFlowNet [18] generates optical flow of these two frames $\boldsymbol{F_{i\to j}}$ exploiting ambient images $\boldsymbol{A_i}$ and $\boldsymbol{A_j}$ jointly. The photometric loss $\boldsymbol{\mathcal{L}_{ph}}$ and the smoothness loss $\boldsymbol{\mathcal{L}_s}$ are applied to images separately, whereas the multi-view loss $\boldsymbol{\mathcal{L}_{mv}}$, which imposes consistency of predicted depths between two frames, is applied pairwise (see Section 4). This scheme is employed for every pair of images from the same scene. The block **Warp** denotes bilinear 2D warping via optical flow and the block **Proj. to 3D** means projecting points into 3D space using the disparities and the camera's intrinsic parameters and adjusting the view angle of points using the camera's extrinsic parameters. After training and for disparity inference, DispNet [29] takes a single dot image $\boldsymbol{I}$ and estimates a disparity map $\boldsymbol{D}$ as output.

loss functions: we incorporate a novel multi-view loss function leveraging optical flow predictions and an improved edge-aware smoothness loss. The training scheme of our DIS-SF model is presented in Figure 1. The photometric loss $\boldsymbol{\mathcal{L}_{ph}}$ enforces consistency between the input image and the warped reference pattern via the estimated disparity map. For smoothness loss $\boldsymbol{\mathcal{L}_s}$, we propose using an edge-aware one similar to [11, 12, 31], except that we extract the edge information directly from the ambient images.

Furthermore, we introduce a novel multi-view loss $\boldsymbol{\mathcal{L}_{mv}}$, which enforces the consistency of the estimated depths between two different views with the help of bilinear warping via optical flow predictions. Note that the photometric loss and smoothness loss apply to each image individually, whereas the multi-view loss applies to all possible permutations of image pairs from the same scene. For more details about the loss functions, refer to Section 4.

We use DispNet [29] for inferring disparity. We also apply Local Contrast Normalization (LCN) preprocessing, suggested in [49, 33], to both dot images $\boldsymbol{I}$ and the reference pattern $\boldsymbol{P}$. Although we use ambient images $\boldsymbol{A}$ in our training scheme, we do not directly employ them as DispNet's input. This makes data preparation more convenient during inference, and DispNet [29] predicts disparity maps $\boldsymbol{D}$ only based on dot images $\boldsymbol{I}$. Instead, the pairs of ambient images are exploited as the input of LiteFlowNet [18] to predict the optical flow map $\boldsymbol{F}$. More discussion on how we use pre-trained LiteFlowNet with ambient images, while it is designed to work with RGB images, as well as an ablation study are provided in the supplementary.

3.2. Multi-Frame Disparity Estimation

Our Multi-Frame (DIS-MF) model combines the information of other frames from the same scene into one frame and generates more accurate disparities. We assume an initial imperfect disparity map is available for each frame beforehand, and we attempt to increase the quality of the disparities by fusing the frames. In this regard, we take the outputs of our DIS-SF model as the imperfect disparities. Compared to traditional RGB depth estimation, aggregating data of multiple frames is more efficacious in structured-light setup because the performance of depth sensing depends on how the dots touch the objects in the environment. Thus, the data contained in the frames are less correlated.

Let $\boldsymbol{\phi} \in \mathbb{R}^{C\times H\times W}$ denote a feature map of size $H \times W$ with C channels, and $\boldsymbol{X} \in \mathbb{R}^{3\times H\times W}$ denote the corresponding 3D points obtained using the imperfect disparities and camera projection matrix $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$. Let us assume we have a pair of images with feature maps of $(\boldsymbol{\phi_i}, \boldsymbol{\phi_j})$ and 3D points of $(\boldsymbol{X_i}, \boldsymbol{X_j})$. Frame i is assumed as the target frame, and we want to fuse the information of $\boldsymbol{\phi_j}$ into $\boldsymbol{\phi_i}$. Our model's first step is warping both feature map $\boldsymbol{\phi}$ and 3D points $\boldsymbol{X}$ on the 2D grid via optical flow predictions $\boldsymbol{F_{i\to j}}$ and $\boldsymbol{F_{j\to i}}$. Optical flow warping places the data of the frames on the 2D grid such that corresponding data of the frames appear in each other's neighborhood on the 2D grid.

Let $\boldsymbol{\phi_{j\to i}} = w^{j\to i}(\boldsymbol{\phi_j})$ and $\boldsymbol{X_{j\to i}} = w^{j\to i}(\boldsymbol{X_j})$ denote warped features and warped points, where $w^{j\to i}(\cdot)$ stands for bilinear 2D warping via the optical flow $\boldsymbol{F_{i\to j}}$. We also define a binary mask map $\boldsymbol{M_{j\to i}} \in \{0,1\}^{1\times H\times W}$ which indicates if the warped data is valid and should be allowed to participate in our fusion framework. We construct $\boldsymbol{M_{j\to i}}$ by evaluating the forward-backward consistency of optical flow predictions, similar to [51, 30]:

$$\begin{aligned}\boldsymbol{M_{j\to i}} &= |\boldsymbol{F_{i\to j}} + w^{j\to i}(\boldsymbol{F_{j\to i}})|^2 \\ &< 0.01 \times (|\boldsymbol{F_{i\to j}}|^2 + |w^{j\to i}(\boldsymbol{F_{j\to i}})|^2) + 0.5 \quad (1)\end{aligned}$$

Despite having all warped data and their validation mask map on the same 2D grid, we do not perform fusion naively on the grid space. As we already mentioned in Section 2, warped features in the structured-light setup contain interfering data of warped dots that make the fusion task complicated. Instead, we propose a fusion block that performs fusion and convolution in the continuous 3D space. Our fusion block also has a sense of faulty imperfect disparities and can prevent those points from contributing to the aggregation. The details of our fusion block and its utilization in our DIS-MF network architecture are as follows.

Fusion Block: Chen *et al.* [9] suggest when depth information of a 2D image is available, it is conceivable to exploit continuous convolution in the 3D space and benefit from both 2D and 3D data processing simultaneously. Such a proposal is consistent with the idea of merging the data of multiple frames as the projected points in the 3D space could be processed regardless of their camera pose. Inspired by them, we propose a fusion block capable of fusing several feature maps originating from different frames into the target frame's feature map. For the sake of simplicity, let us assume we only have two frames and intend to merge the feature map $\boldsymbol{\phi_j}$ into the target feature map $\boldsymbol{\phi_i}$. The functionality of the fusion block is illustrated in Figure 2. We use the continuous 3D convolution [40] as the core element of our fusion block. Most architectures that exploit 3D convolution on the point cloud require running exhaustive search algorithms to find points in the neighborhood [9, 26, 46, 43, 3, 40], which is infeasible to perform on dense data such as ours. For instance, Chen *et al.* [9] pre-compute the indices of nearest neighbors for all points. To mitigate the issue, we propose a novel technique that is practical in real-time processing. Since our data is not fully unstructured, we suspect points that are close in 3D space will be close on the 2D grid map if they are warped to the same camera perspective, but not vice versa.

Accordingly, we form the concatenated feature map $[\boldsymbol{\phi_{j\to i}}, \boldsymbol{\phi_i}]$ and point map $[\boldsymbol{X_{j\to i}}, \boldsymbol{X_i}]$ and slide a 3×3 window over each 2D grid map simultaneously and perform convolution only on points inside the sliding window similarly to a conventional CNN. The difference is, instead of performing a weighted sum with learnable parameters, we search for the nearest points and perform continuous convolution. For simplifying the equations, let $\boldsymbol{\phi_{i\to i}} = \boldsymbol{\phi_i}$, $\boldsymbol{X_{i\to i}} = \boldsymbol{X_i}$, and $\boldsymbol{M_{i\to i}} = \vec{\boldsymbol{1}}$. Also, let $\boldsymbol{\phi}(h,w)$ and $\boldsymbol{X}(h,w)$ represent the features and the coordinate of the position (h,w) on the grid map where $0 \le h < H$ and $0 \le w < W$. We first search for the nearest points to the center point of the sliding window on the target frame i:

$$\begin{aligned}&l^*(h,w), m^*(h,w), n^*(h,w) \\ &= k\text{-}\arg\min_{\substack{l\in\{i,j\}\\ -1\le m\le +1\\ -1\le n\le +1}} \frac{\left|\boldsymbol{X_{l\to i}}(h+m, w+n) - \boldsymbol{X_i}(h,w)\right|}{\boldsymbol{M_{l\to i}} + \epsilon} \quad (2)\end{aligned}$$

where k-$\arg\min g(\cdot)$ returns the k indices that minimize the function $g(\cdot)$, and ϵ is a small constant. $\boldsymbol{M_{l\to i}}$ is used in the denominator to exclude invalid points, and we set $k = 9$ to ensure all returned indices correspond to valid pixels due to the window size 3×3. To extend the model to fuse more than two frames, l in Equation 2 should span all available frames rather than only $\{i,j\}$. The convolution's result is:

$$\begin{aligned}\boldsymbol{\phi'_i}(h,w) = \Psi \times &\sum_{l^*,m^*,n^*} \Big(\boldsymbol{\phi_{l^*\to i}}(h+m^*, w+n^*) \\ &\odot \text{MLP}\big(\boldsymbol{X_{l^*\to i}}(h+m^*, w+n^*) - \boldsymbol{X_i}(h,w)\big)\Big) \quad (3)\end{aligned}$$

where MLP is a multi-layer perceptron mapping 3D vectors to C-dimensional weights, $\odot$ denotes element-wise product, and Ψ is a $C \times C$ learnable weight matrix. This implementation can be regarded as a continuous version of separable convolution. The MLP and weighted sum perform depth-wise convolution, while the linear transformation resembles 1×1 convolution [9].

As shown in Figure 2, we adopt two 3D convolutions in each fusion block. Accordingly, we warp the other frames' outputs of the first 3D convolution to the target frame $\boldsymbol{\phi'_{j\to i}}$ and fuse them into the second 3D convolution as well. We also employ traditional 2D CNNs in the fusion block because there are some shortcomings to 3D convolution, such

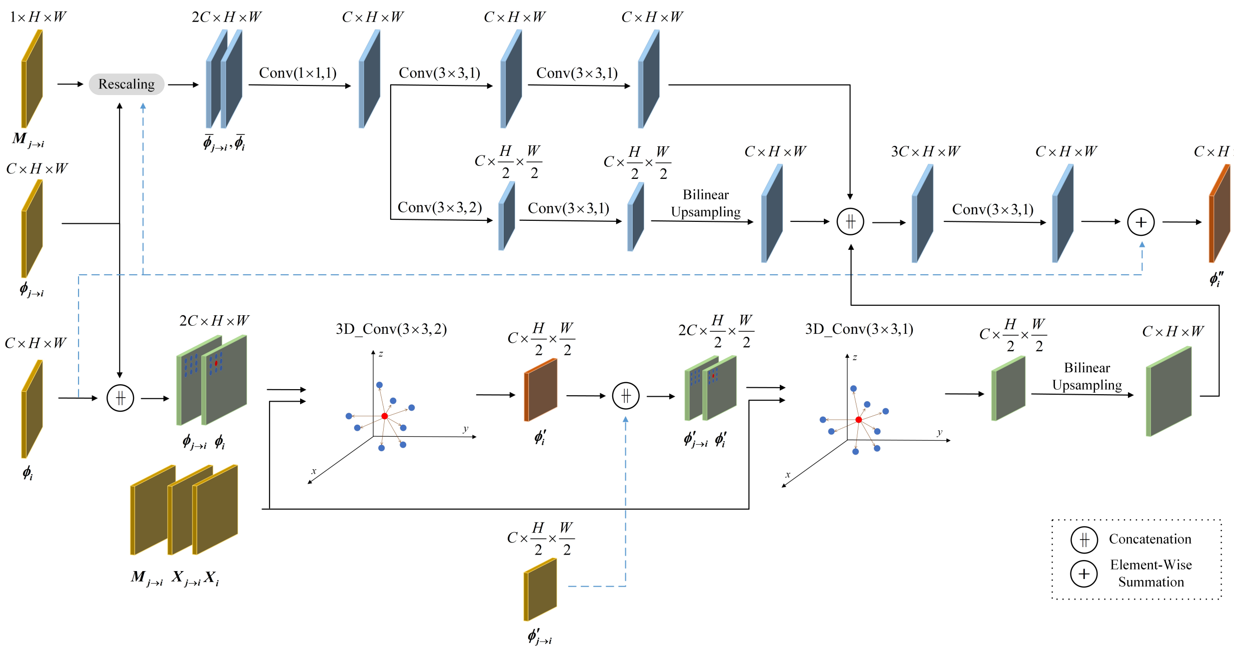

Figure 2. Internal architecture of our proposed fusion block, whose details of utilization in our DIS-MF model are illustrated in Section 3.2 and Figure 3. We depict how features of an auxiliary frame $\phi_{j\to i}$ are being fused into the target frame's features ϕ_i. Binary mask map $M_{j\to i}$, 3D points of the target frame X_i and warped frame $X_{j\to i}$, and the warped result of the first 3D convolution of the auxiliary frame $\phi'_{j\to i}$ are also inputs of this block. ϕ''_i stands for the output of this block, and ϕ'_i represents the output of the first 3D convolution required for fusing into other frames' fusion blocks. Conv$(k \times k, s)$ and 3D_Conv$(k \times k, s)$ denote 2D and continuous 3D convolution respectively, with kernel size of k and stride s, and the block **Rescaling** denotes the operations described in Equation (4).

as edge fattening near the boundaries of objects and background. To merge the feature maps in 2D CNNs, we handle invalid points differently by proposing a scheme similar to dropout [37]. To do so, we first zero out features of invalid points, and then rescale the remaining valid features inversely proportionally to the number of valid frames for each point on the 2D grid:

$$\forall l \in \{i, j\} : \bar{\phi}_{l\to i} = \frac{\phi_{l\to i} \times M_{l\to i}}{\sum_{p\in\{i,j\}} M_{p\to i}} \tag{4}$$

The 3D convolutions along with 2D CNNs jointly construct the fusion block, which is capable of processing high-resolution feature maps and effectively benefits from the information of other frames from the same scene. SELU nonlinearity [23] and Group Norm [44] are used after each convolution. We prefer Group Norm to Batch Norm [21] in our model because Group Norm statistics are independent of the number of samples in a batch and make training large networks feasible with smaller batch sizes.

Network Architecture: Figure 3 illustrates the network architecture of our DIS-MF model. The architecture includes three sections as follows. The preprocessing section takes the images (I, A) and the imperfect disparity D as input and generates high-level feature maps for each frame individually. Next, the feature maps are fed into cascaded series of fusion blocks, along with their corresponding 3D points X and binary masks M required for merging and 3D convolutions to obtain fused feature maps. Warping with the optical flow is employed whenever any data on a 2D grid map is needed to be warped to another frame's 2D grid.

Lastly, the fused feature maps go through a refinement structure to preserve high-resolution details such as edges and reduce distortions resulting from combining frames. Our refinement section is inspired by the one in [49], but takes the upsampled fused features and the ambient image as inputs. In both the preprocessing and refinement sections, we exploited residual blocks introduced in [15] to promote gradient backpropagation and expedite the training process.

An ablation study of design choices for the DIS-MF network architecture is provided in the supplementary.

3.3. Fine-Tuning the Single-Frame Model

For purposes where resources are limited during inference, we propose an alternative approach to exploit the scheme of fusing image frames. We suggest that after training the DIS-MF model, the produced disparities can be used as an auxiliary loss function to supervise and fine-tune the

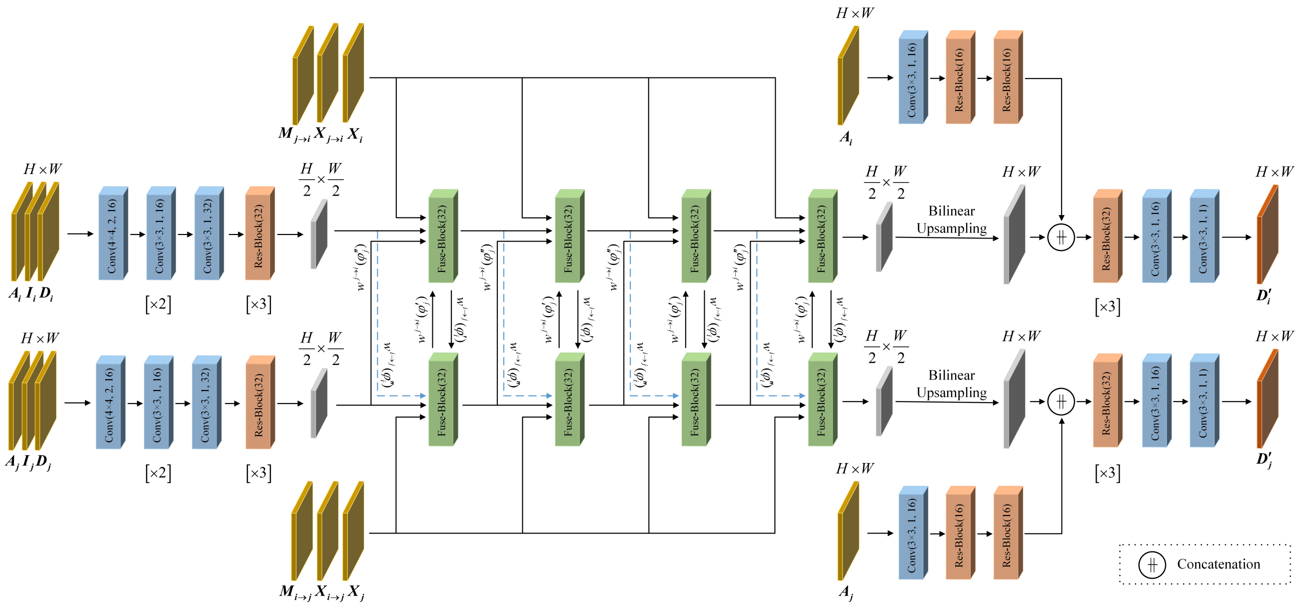

Figure 3. Our DIS-MF network architecture when only two frames i and j are combined. Warping the first 3D convolution output $\boldsymbol{\phi'}$ and the final output of each fusion block $\boldsymbol{\phi''}$ using the relative optical flows are denoted by $w^{i\to j}(\cdot)$ and $w^{j\to i}(\cdot)$. Note that $\boldsymbol{D}$ stands for imperfect disparity participating as one of the inputs, and $\boldsymbol{D'}$ represents the final predicted disparity of the model. This figure depicts the inference network of our DIS-MF model. For training the DIS-MF model, this network replaces those individual DispNet [29] networks in the DIS-SF model in Figure 1, and the same scheme and loss functions (see Section 4) are adopted.

single-frame network. The resulting model, DepthInSpace Fine-Tuned Single-Frame (DIS-FTSF), can yield more accurate disparity maps with no additional memory or computation cost during inference compared with DIS-SF.

4. Loss Functions

Here we introduce our loss functions employed in our models. Let $\Gamma = \{\boldsymbol{I_i}, \boldsymbol{A_i}\}_{i=0}^{N-1}$ denote the image samples from the same scene. The overall loss function consists of a photometric loss $\boldsymbol{\mathcal{L}_{ph}}$, a smoothness loss $\boldsymbol{\mathcal{L}_{s}}$, a multi-view loss $\boldsymbol{\mathcal{L}_{mv}}$, and a pseudo-ground truth loss $\boldsymbol{\mathcal{L}_{pgt}}$:

$$\boldsymbol{\mathcal{L}} = \frac{1}{N}\sum_{i\in\Gamma}(\boldsymbol{\mathcal{L}^i_{ph}} + \lambda_1\boldsymbol{\mathcal{L}^i_{s}} + \lambda_2\boldsymbol{\mathcal{L}^i_{pgt}}) + \frac{1}{N(N-1)}\sum_{i,j\in\Gamma}\lambda_3\boldsymbol{\mathcal{L}^{ij}_{mv}} \quad (5)$$

where $\{\lambda_k\}_{k=1}^3$ are weighting constants, which do not necessarily take the same value in all of our models.

Let $\boldsymbol{D}$ denote the disparity map, $\boldsymbol{\tilde{I}}$ denote the local contrast normalized input image, and $\boldsymbol{P}$ denote the local contrast normalized reference dot pattern. Similarly to CTD, we employ the smooth Census transform [14], represented by $\|\cdot\|_C$, in our photometric loss:

$$\boldsymbol{\mathcal{L}^i_{ph}} = \sum_{h,w}\|\boldsymbol{\tilde{I}_i}(h,w) - \boldsymbol{P}\big(h, w - \boldsymbol{D_i}(h,w)\big)\|_C \quad (6)$$

Since we assume the availability of ambient images, we introduce an edge-aware smoothness loss similar to [11, 12]. The smoothness loss imposes consistency between disparity map discontinuities and edges in the ambient image:

$$\boldsymbol{\mathcal{L}^i_{s}} = |\nabla_h \boldsymbol{D_i}|e^{-\beta|\nabla_h \boldsymbol{A_i}|} + |\nabla_w \boldsymbol{D_i}|e^{-\beta|\nabla_w \boldsymbol{A_i}|} \quad (7)$$

where ∇_h and ∇_w stand for 2D spatial gradients and β is a constant. Moreover, we impose the consistency between the predicted depths in each pair of images from the same scene. Let $\boldsymbol{X_i}$ and $\boldsymbol{X_j}$ denote the 3D point clouds of the two frames obtained using the momentary predicted disparities and camera intrinsic matrix. Our multi-view loss is:

$$\boldsymbol{\mathcal{L}^{ij}_{mv}} = \left|\left\langle \boldsymbol{X_i} - w^{j\to i}(\boldsymbol{T_{j\to i}} \times [\boldsymbol{X_j}, \vec{\boldsymbol{1}}])\right\rangle_z\right| \times \boldsymbol{M'_{j\to i}} \quad (8)$$

where $\boldsymbol{T_{j\to i}} \in \mathbb{R}^{3\times4}$ is the transformation matrix consisting of ego motion parameters, $\vec{\boldsymbol{1}}$ is an all one matrix, and $\langle\cdot\rangle_z$ operator returns the depth z of its input 3D vector. $\boldsymbol{M'_{j\to i}}$ is a binary mask map validating warped points similarly to $\boldsymbol{M_{j\to i}}$ in Section 3.2, but it strictly excludes low confidence points from supervising the training. For more details regarding $\boldsymbol{M'_{j\to i}}$, refer to the supplementary.

Lastly, only in our DIS-FTSF model, we use the more accurate fused disparity $\boldsymbol{D'}$ as pseudo-ground truth to improve the quality of the imperfect disparity $\boldsymbol{D}$. We impose the L1 consistency between $\boldsymbol{D}$ and $\boldsymbol{D'}$ as an auxiliary loss:

$$\boldsymbol{\mathcal{L}^i_{pgt}} = |\boldsymbol{D_i} - \boldsymbol{D'_i}| \quad (9)$$

5. Experiments

Datasets: To evaluate our models and compare them with existing methods, we examine the accuracy of depth estimation on three synthetic datasets and one real dataset. We used the tool provided by CTD [33] to render the synthetic data. Rendering is done in the same experimental setup as CTD with the same objects of the ShapeNet Core dataset [6], but the images are captured by a sensor whose parameters are set similar to our own hardware. One dataset is rendered using the Kinect dot pattern for projection, and the second dataset is generated utilizing our own theoretical dot pattern for the projector. For the last synthetic dataset, we projected and captured the dot pattern in a real laboratory environment and used the observed pattern for rendering the dataset. In this regard, we use a virtual projector with the same parameters of the capturing camera.

We incorporated multiple datasets because different dot patterns could lead to different depth sensing performances. The denser the dots are, the better the performance is. However, choosing a dot pattern could be restricted by hardware limitations or available illumination power. That is why we examine the models' performances over different projected dot patterns. For each synthetic dataset, we create 8192 sequences for training, 512 sequences for validation, and 512 sequences for testing. Each sequence contains 4 pair of dot images and ambient images from the same scene.

We also evaluate the models on a smaller real dataset to show the generalization of our method in an actual setup. The data include 148 sequences of 4 pairs of dot images and ambient images captured from 4 different scenes. The sensor we use is equipped with a programmable switch, enabling the projector to be on and off, so it can capture dot images and ambient images alternately at the rate of 15 fps each. Given the capturing rate, each pair of dot image and ambient image captures the same scene approximately. We put aside 18 sequences for validating and testing and utilized 130 sequences in training. To obtain accurate ground truth we used a 3D scanner, the data of which is only used for evaluation. Due to the scanner limitations, we take a set of partial scans that best cover the scene. These are fused together to create a 3D model using the point-to-plane variant of the ICP algorithm [8]. A 3D mesh is then produced using the Ball-Pivoting algorithm [1]. For estimating the camera motion parameters, the same ICP variant is used to align the ground truth 3D model and the 3D model obtained from the structured-light sensor via the block matching technique.

More details of the datasets and also implementing our models are provided in the supplementary.

Metrics: We use the percentage of outliers $o(t)$ as in [33] for quantitative evaluation, which is the percentage of pixels where the difference between the estimated and the ground truth disparities is greater than t.

Comparison with existing methods: We compare our models with Semi-Global Matching (SGM) algorithm [16], HyperDepth [34], and CTD [33]. We observed through experiments that the window size of 13 for the SGM algorithm best suits our dataset. For HyperDepth, we used the same reimplementation code provided by [33] with the hyperparameters that yield the best results in the original paper [34]. Since HyperDepth is a supervised method, we used the ground truth depth maps for training this model.

When training either CTD or our models on the real dataset, we use the pre-trained weights obtained from the

Data	Method	$o(0.5)$	$o(1)$	$o(2)$	$o(5)$
Synthetic (Kinect Patt.)	SGM	10.36	9.13	8.76	2.45
	HyperDepth[a]	4.38	3.22	2.69	2.39
	CTD	2.74	1.45	0.77	0.24
	DIS-SF	2.11	1.13	0.59	0.16
	DIS-FTSF	**1.92**	**1.00**	**0.51**	**0.14**
	DIS-MF	1.59	0.72	0.33	0.10
Synthetic (Our Patt.)	SGM	12.93	11.64	11.22	4.06
	HyperDepth[a]	7.35	6.48	6.11	5.86
	CTD	3.38	1.71	0.85	0.28
	DIS-SF	2.31	1.24	0.62	0.19
	DIS-FTSF	**1.96**	**0.95**	**0.45**	**0.12**
	DIS-MF	1.58	0.71	0.32	0.10
Synthetic (Observed Patt.)	SGM	12.45	10.37	9.55	4.83
	HyperDepth[a]	6.13	4.92	4.34	4.00
	CTD	3.76	2.25	1.03	0.37
	DIS-SF	3.66	2.16	1.00	0.23
	DIS-FTSF	**2.87**	**1.48**	**0.66**	**0.17**
	DIS-MF	2.46	1.24	0.54	0.14
Real	SGM[b]	25.54	19.23	17.75	16.96
	HyperDepth[a]	34.62	25.09	22.49	21.77
	CTD	22.74	9.26	3.79	**1.00**
	DIS-SF	17.95	7.93	3.59	1.14
	DIS-FTSF	**17.06**	**7.48**	**3.47**	1.11
	DIS-MF	16.07	7.14	3.41	1.09

[a] HyperDepth is a supervised model trained with ground truth.
[b] We evaluated all models on the full image. SGM performs poorly on the real data due to large disparities in the dataset and its incapability of predicting valid depths on a large portion of the image (whereas learning models extrapolate in those areas). As an example, if we evaluated models on a cropped area of the depth maps, $o(0.5)$ and $o(1)$ would drop to 15.56 and 8.81 for SGM, and 13.06 and 5.08 for DIS-FTSF.

Table 1. Quantitative comparison of the SGM algorithm [16], HyperDepth [34], and CTD [33] versus our DIS-SF, DIS-FTSF, and DIS-MF models. Numbers are percentages of outliers $o(t)$, that is the fraction of pixels for which the estimated disparity is more than t away from ground truth. We indicate in bold the best performance among single-frame methods (*i.e.* all but our DIS-MF model, which, as expected, performs the best).

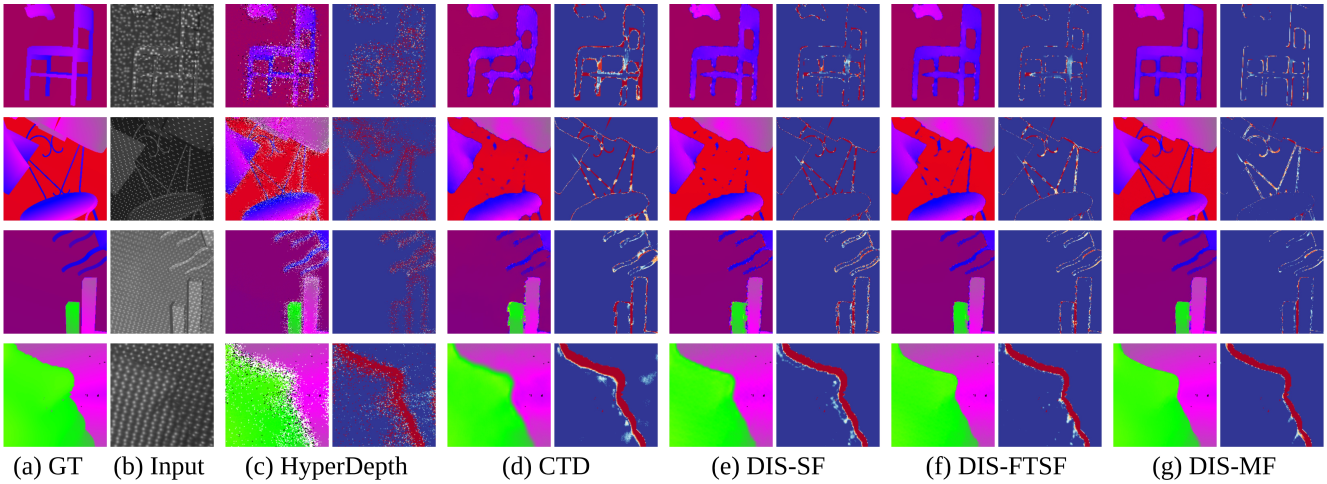

Figure 4. Qualitative results of the methods and their corresponding error maps. (a) Ground truth disparity map. (b) Input dot image with projected pattern. (c) HyperDepth [34]. (d) CTD [33]. (e) Our DIS-SF model. (f) Our DIS-FTSF model. (g) Our DIS-MF Model. Each row represents a sample corresponding to each dataset in Table 1. Points for which the ground truth data is unavailable are excluded from evaluation. For more sample images and extended qualitative evaluations, refer to the supplementary material.

synthetic data in order to speed up the training process. Moreover, due to the limitations of the 3D scanner we used to capture ground truth, we had to put objects very close to the camera, resulting in very large values of disparities. Therefore, the statistics of disparities between the real dataset and the synthetic dataset are different, causing networks to get stuck in local minima when they are fine-tuned on the real data. We handled this issue by incorporating an additional loss function and using the SGM algorithm's valid outputs as pseudo-ground truth during the first few epochs of training. This loss function warms up the training process and resembles a coarse estimation of the ground truth at the beginning of the training. This stratagem prevents the networks from getting stuck in local minima and is used for both CTD and our models.

Qualitative comparison of the estimated disparities of the models on different datasets is depicted in Figure 4. It is notable that all of our models produce sharper edges than the baseline model, CTD. Remarkably, our DIS-MF model best preserves the edges and is also capable of retaining high-resolution details. On the other hand, HyperDepth shows poor performance at discontinuities despite its accuracy in smooth regions. The figure also contrasts the quality of our DIS-SF and DIS-FTSF models and exhibits the usefulness of exploiting the DIS-MF model outputs to improve the accuracy of the DIS-SF model. Extended qualitative evaluations are provided in the supplementary material.

Table 1 provides the quantitative evaluation of the discussed models and shows the outcomes are consistent with the qualitative results. Table 1 also reflects the effect of the dot pattern on the performance of algorithms, where most models have the best accuracy in the experiment with the denser Kinect dot pattern. However, our models show robustness in all experiments. Particularly, DIS-MF yields overall the best results in all the experiments. Also, among the methods that predict disparities based on a single image, our DIS-FTSF model outperforms others overall.

For further experiments and ablation studies of the loss functions, validation masks, components of DIS-MF network, effect of imperfect disparities, utilized optical flow network, and extended qualitative analysis, refer to the supplementary material.

6. Conclusion

We proposed DepthInSpace (DIS), which includes three self-supervised deep learning models to estimate depth from structured-light sensor data. Leveraging optical flow, we utilize information from multiple video frames from the same scene to improve depth estimation accuracy in three different self-supervised fashions. We qualitatively and quantitatively evaluated our models over four datasets: a publicly available synthetic dataset, two synthetic datasets customized with our setup parameters and dot pattern, and a real dataset that we made publicly available. The experiments validate the superiority of our models over the existing state-of-the-art methods.

The natural extension for future work will be on the one hand to apply our method to active stereo setup, combining the strengths of both sources of information, and on the other hand to deal with a simplified setup, for instance with a sparser less energy-hungry pattern of illumination.

Acknowledgement
This research was supported by ams OSRAM.

References

[1] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 7

[2] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in neural information processing systems*, pages 35–45, 2019. 2

[3] Alexandre Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020. 2, 4

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 2

[5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

[7] Qifeng Chen and Vladlen Koltun. Fast mrf optimization with application to depth reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3914–3921, 2014. 2

[8] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vis. Comput.*, 10:145–155, 1992. 7

[9] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10023–10032, 2019. 2, 4

[10] Sean Ryan Fanello, Julien Valentin, Christoph Rhemann, Adarsh Kowdle, Vladimir Tankovich, Philip Davidson, and Shahram Izadi. Ultrastereo: Efficient learning-based matching for active stereo systems. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6535–6544. IEEE, 2017. 1, 2

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 2, 3, 6

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019. 2, 3, 6

[13] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 2

[14] David Hafner, Oliver Demetz, and Joachim Weickert. Why is the census transform good for robust optic flow computation? In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 210–221. Springer, 2013. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[16] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 1, 7

[17] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[18] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018. 2, 3

[19] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[20] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. In *International Conference on Learning Representations*, 2018. 2

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 5

[22] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2017. 1

[23] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980, 2017. 5

[24] Ioannis Kleitsiotis, Nikolaos Dimitriou, Konstantinos Votis, and Dimitrios Tzovaras. Color-guided adaptive support weights for active stereo systems. In *International Conference on Computer Vision Systems*, pages 501–510. Springer, 2019. 1, 2

[25] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017. 2

[26] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed

points. In *Advances in neural information processing systems*, pages 820–830, 2018. 2, 4

[27] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Trans. Graph.*, 39(4), July 2020. 2

[28] Manuel Martinez and Rainer Stiefelhagen. Kinect unleashed: Getting control over high resolution depth maps. In *MVA*, pages 247–250, 2013. 2

[29] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 3, 6

[30] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, New Orleans, Louisiana, Feb. 2018. 4

[31] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9250–9256. IEEE, 2019. 2, 3

[32] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12240–12249, 2019. 2

[33] Gernot Riegler, Yiyi Liao, Simon Donne, Vladlen Koltun, and Andreas Geiger. Connecting the dots: Learning representations for active monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7624–7633, 2019. 1, 2, 3, 7, 8

[34] Sean Ryan Fanello, Christoph Rhemann, Vladimir Tankovich, Adarsh Kowdle, Sergio Orts Escolano, David Kim, and Shahram Izadi. Hyperdepth: Learning depth from structured light without matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5441–5450, 2016. 1, 2, 7, 8

[35] Sean Ryan Fanello, Julien Valentin, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, Carlo Ciliberto, Philip Davidson, and Shahram Izadi. Low compute and fully parallel computer vision with hashmatch. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3874–3883, 2017. 1

[36] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 1920. 1

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 5

[38] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *International Conference on Learning Representations*, 2019. 2

[39] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 2

[40] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018. 2, 4

[41] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8071–8081, 2019. 2

[42] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *European conference on computer vision*, pages 230–247. Springer, 2020. 2

[43] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 2, 4

[44] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5

[45] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016. 2

[46] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018. 2, 4

[47] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2

[48] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 2

[49] Yinda Zhang, Sameh Khamis, Christoph Rhemann, Julien Valentin, Adarsh Kowdle, Vladimir Tankovich, Michael Schoenberg, Shahram Izadi, Thomas Funkhouser, and Sean Fanello. Activestereonet: End-to-end self-supervised learning for active stereo systems. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–801, 2018. 1, 2, 3, 5

[50] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 2

[51] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 36–53, 2018. 2, 4