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. Depending 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 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
The training scheme of our DIS-SF model for a sample pair of frames $i$ and $j$, and a reference pattern $P$. The dot images $I_i$ and $I_j$ are fed to the DispNet [29] separately to predict disparities $D_i$ and $D_j$. On another path, LiteFlowNet [18] generates optical flow of these two frames $F_{i\rightarrow j}$ exploiting ambient images $A_i$ and $A_j$ jointly. The photometric loss $L_{ph}$ and the smoothness loss $L_s$ are applied to images separately, whereas the multi-view loss $L_{mv}$, 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 $I$ and estimates a disparity map $D$ as output.

Furthermore, we introduce 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 $L_{ph}$ enforces consistency between the input image and the warped reference pattern via the estimated disparity map. For smoothness loss $L_s$, 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 $L_{mv}$, 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 $I$ and the reference pattern $P$. Although we use ambient images $A$ 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 $D$ only based on dot images $I$. Instead, the pairs of ambient images are exploited as the input of LiteFlowNet [18] to predict the optical flow map $F$. 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 $\phi \in \mathbb{R}^{C \times H \times W}$ denote a feature map of size $H \times W$ with $C$ channels, and $X \in \mathbb{R}^{3 \times H \times W}$ denote the corresponding 3D points obtained using the imperfect disparities and camera projection matrix $K \in \mathbb{R}^{3 \times 4}$. Let us assume we have a pair of images with feature maps of $(\phi_i, \phi_j)$ and 3D points of $(X_i, X_j)$. Frame $i$ is assumed as the target frame, and we want to fuse the information of $\phi_j$ into $\phi_i$. Our model’s first step is warping both feature map $\phi$ and 3D points $X$ on the 2D grid via optical flow predictions $F_{i \rightarrow j}$ and $F_{j \rightarrow i}$. 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 $\phi_{j \rightarrow i} = w^{j \rightarrow i}(\phi_j)$ and $X_{j \rightarrow i} = w^{j \rightarrow i}(X_j)$ denote warped features and warped points, where $w^{j \rightarrow i}(\cdot)$ stands for bilinear 2D warping via the optical flow $F_{i \rightarrow j}$. We also define a binary mask map $M_{j \rightarrow i}$ to ensure all returned indices correspond to valid pixels due to 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.

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 $\phi_j$ into the target feature map $\phi_i$. 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 $[\phi_{j \rightarrow i}, \phi_i]$ and point map $[X_{j \rightarrow i}, X_i]$ and slide a $3 \times 3$ 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 $\phi_{i \rightarrow i} = \phi_i$, $X_{i \rightarrow i} = X_i$, and $M_{i \rightarrow i} = \tilde{I}$. Also, let $\phi(h, w)$ and $X(h, w)$ represent the features and the coordinate of the position $(h, w)$ on the grid map where $0 \leq h < H$ and $0 \leq w < W$. We first search for the nearest points to the center point of the sliding window on the target frame $i$:

$$l^*(h, w), m^*(h, w), n^*(h, w)$$

$$= k \cdot \text{arg min}_{i \leq m \leq i + 1, -1 \leq n \leq +1} \frac{|X_{l \rightarrow i}(h + m, w + n) - X_i(h, w)|}{M_{l \rightarrow i} + \epsilon}$$

where $k \cdot \text{arg min} g(\cdot)$ returns the $k$ indices that minimize the function $g(\cdot)$, and $\epsilon$ is a small constant. $M_{l \rightarrow i}$ 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 $3 \times 3$. 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:

$$\phi'_{j \rightarrow i}(h, w) = \Psi \times \sum_{l^*, m^*, n^*} \left(\phi_{l^* \rightarrow i}(h + m^*, w + n^*) \odot \text{MLP}(X_{l^* \rightarrow i}(h + m^*, w + n^*) - X_i(h, w))\right)$$

where MLP is a multi-layer perceptron mapping 3D vectors to $C$-dimensional weights, $\odot$ denotes element-wise product, and $\Psi$ 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 $1 \times 1$ 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 $\phi'_{j \rightarrow i}$ 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 as
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 $\phi_j \rightarrow i$ are being fused into the target frame’s features $\phi_i$. Binary mask map $M_j \rightarrow i$, 3D points of the target frame $X_i$ and warped frame $X_j \rightarrow i$, and the warped result of the first 3D convolution of the auxiliary frame $\phi'_j \rightarrow i$ are also inputs of this block. $\phi''_i$ stands for the output of this block, and $\phi'_i$ represents the output of the first 3D convolution required for fusing into other frames’ fusion blocks. $\text{Conv}(k \times k, s)$ and $\text{3DConv}(k \times k, s)$ denote 2D and continuous 3D convolution respectively, with kernel size of $k$ and stride $s$, and the block $\text{Rescaling}$ denotes the operations described in Equation (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
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 \( \Gamma = \{I_i, A_i\}_{i=1}^{N} \) denote the image samples from the same scene. The overall loss function consists of a photometric loss \( L_{ph} \), a smoothness loss \( L_{s} \), a multi-view loss \( L_{mv} \), and a pseudo-ground truth loss \( L_{pgt} \):

\[
L = \frac{1}{N} \sum_{i \in \Gamma} (L_{ph}^i + \lambda_1 L_{s}^i + \lambda_2 L_{pgt}^i) + \frac{1}{N(N-1)} \sum_{i,j \in \Gamma} \lambda_3 L_{mv}^{ij}
\]

(5)

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

Let \( D \) denote the disparity map, \( I \) denote the local contrast normalized input image, and \( P \) 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:

\[
L_{ph}^i = \sum_{h,w} \| \tilde{I}_i(h,w) - P(h,w - D_i(h,w)) \|_{C}
\]

(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:

\[
L_{s}^i = |\nabla_h D_i|^\beta |\nabla_w A_i| + |\nabla_w D_i|^\beta |\nabla_h A_i|
\]

(7)

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

\[
L_{mv}^{ij} = \left\| X_i - w^{j-i} \left( T_{j \rightarrow i} \times [X_j, \bar{1}] \right) \right\| z \times M_{j \rightarrow i}^{j}
\]

(8)

where \( T_{j \rightarrow i} \in \mathbb{R}^{3 \times 4} \) is the transformation matrix consisting of ego motion parameters, \( \bar{1} \) is an all one matrix, and \( (\cdot)_{z} \) operator returns the depth \( z \) of its input 3D vector. \( M_{j \rightarrow i}^{j} \) is a binary mask map validating warped points similarly to \( M_{j \rightarrow i} \) in Section 3.2, but it strictly excludes low confidence points from supervising the training. For more details regarding \( M_{j \rightarrow i}^{j} \), refer to the supplementary.

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

\[
L_{pgt}^i = |D_i - D'_i|
\]

(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

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

\(^a\) 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).
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


[13] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raven-


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


