

Connecting the Dots: Learning Representations for Active Monocular Depth Estimation

Gernot Riegler^{1,*} Yiyi Liao^{2,*} Simon Donne² Vladlen Koltun¹ Andreas Geiger²

¹Intel Intelligent Systems Lab

²Autonomous Vision Group, MPI-IS / University of Tübingen

{firstname.lastname}@intel.com

{firstname.lastname}@tue.mpg.de

Abstract

We propose a technique for depth estimation with a monocular structured-light camera, i.e., a calibrated stereo set-up with one camera and one laser projector. Instead of formulating the depth estimation via a correspondence search problem, we show that a simple convolutional architecture is sufficient for high-quality disparity estimates in this setting. As accurate ground-truth is hard to obtain, we train our model in a self-supervised fashion with a combination of photometric and geometric losses. Further, we demonstrate that the projected pattern of the structured light sensor can be reliably separated from the ambient information. This can then be used to improve depth boundaries in a weakly supervised fashion by modeling the joint statistics of image and depth edges. The model trained in this fashion compares favorably to the state-of-the-art on challenging synthetic and real-world datasets. In addition, we contribute a novel simulator, which allows to benchmark active depth prediction algorithms in controlled conditions.

1. Introduction

With the introduction of the Microsoft Kinect, active consumer depth cameras have greatly impacted the field of computer vision, leading to algorithmic innovations [13, 28] and novel 3D datasets [6, 7, 35, 37], especially in the context of indoor environments. Likewise, the increasing availability of affordable and comparably robust depth sensing technologies has accelerated research in robotics.

While this progress is remarkable, current research based on consumer depth cameras is limited by the depth sensing technology used onboard these devices, which is typically kept simple due to computational and memory constraints. For instance, the original Kinect v1 uses a simple correlation-based block matching technique [33], while Intel RealSense cameras exploit semi-global matching [16]. However, neither of these approaches is state-of-the-art in current stereo benchmarks [25, 32, 34], most of which are

dominated by learning-based approaches.

In this paper, we exploit the potential of deep learning for this task. In particular, we consider the setting of active monocular depth estimation. Our setup comprises a camera and a laser projector which illuminates the scene with a known random dot pattern. Depending on the depth of the scene this pattern varies from the viewpoint of the camera. This scenario is appealing as it requires only a single camera compared to active stereo systems. Furthermore, the neural network is not tasked to find correspondences between images. Instead, our network directly estimates disparity from the point pattern in a local neighborhood of a pixel.

Training deep neural networks for active depth estimation is difficult as obtaining sufficiently large amounts of accurately aligned ground-truth is very challenging. We therefore propose to train active depth estimation networks without access to ground-truth depth in a fully self-supervised, or weakly supervised fashion. Towards this goal, we combine a photometric loss with a disparity loss which considers the edge information available in the ambient image. We further propose a geometric loss which enforces multi-view consistency of the predicted geometry. To the best of our knowledge this is the first deep learning approach to active monocular depth estimation.

In summary, we make the following contributions: We find that a convolutional network is surprisingly effective at estimating disparity, despite the fact that information about the absolute location is not explicitly encoded in the input features. Based on these findings, we propose a deep network for active monocular depth prediction. Our method does not require pseudo ground-truth from classical stereo algorithms as in [10]. Instead, it gains robustness by photometric and geometric losses. We show that the ambient edge information can be disentangled reliably from a single input image, yielding highly accurate depth boundaries despite the sparsity of the projected IR pattern. Research on active depth prediction is hampered by the lack of large datasets with accurate ground-truth depth. We thus contribute a simulator and dataset which allow to benchmark active depth prediction algorithms in realistic, but controlled conditions.

* Joint first authors with equal contribution.

2. Related Work

Active Depth Sensing: Structured light estimation techniques use a projector to illuminate the scene with known light pattern which allows to reconstruct also textureless scenes with high accuracy. Techniques that fall into this category can be classified as either *temporal* or *spatial*. Temporal techniques illuminate the scene with a temporally varying pattern which can be uniquely decoded at every camera pixel. This requires multiple images of the same scene and thus, cannot be employed in dynamic scenes.

We therefore focus our attention on the spatial structured light setting where depth information is encoded in a locally unique 2D pattern. Most related approaches obtain depth from the input image by searching for local correspondences between the camera image and a reference pattern. A prime example is the algorithm in the Kinect V1 sensor [21] which first extracts dots from the input image and then correlates a local window around each dot with corresponding patches in the reference image. This is similar to classical block matching algorithms in the stereo literature [33]. Despite facing an easier task compared to the passive stereo setting, correlation based algorithms suffer in accuracy due to their simplifying assumptions about reflectance (photoconsistency) and geometry (constant disparity inside an entire patch).

Fanello et al. [10] show a different formulation: depth estimation as a supervised learning problem. More specifically, exploiting epipolar geometry they train one random forest per row predicting for every pixel the absolute x -coordinate in the reference image. This point-of-view allows them to obtain a very fast parallel implementation, running at 375 Hz at Megapixel resolution. To train their random forests, they leverage PatchMatch Stereo [1] as pseudo ground-truth. In contrast, we capitalize on the strengths of deep learning and propose a deep network that can be trained in a self-supervised fashion. In addition to the projected point pattern, our loss functions exploit multi-view consistency, as well as ambient information.

Active stereo setups like the Intel RealSense D435 exploit structured light to improve binocular stereo reconstruction by augmenting textureless areas with a pattern on which traditional methods can be applied [16, 33]. Fanello et al. [11] propose an algorithm to learn discriminative features that are efficient to match. Zhang et al. [41] exploit ideas from self-supervised learning to train an active stereo network without needing ground-truth depth. This setup is similar to the passive stereo setup with a stereo image pair as input and the task is to learn a correlation function. In contrast, we consider the active monocular setup and use self-supervised learning to train a network that predicts disparity from a single annotated image.

Stereo Matching: Binocular stereo matching is one of the

oldest problems in computer vision and current approaches [19, 22, 36] achieve impressive performance on established benchmarks like KITTI [25] or Middlebury [32]. However, passive techniques still suffer in textureless regions where the data term is ambiguous and the model needs to interpolate large gaps. This is particularly problematic for indoor environments where textureless regions dominate.

In this paper, we mitigate this problem by leveraging a pattern projector offset by a baseline with respect to the camera. However, we exploit ideas from the stereo community to self-supervise our approach, i.e., we train our model such that the reference pattern warped by the estimated disparity coincides with the observed pattern.

Single Image Depth Prediction: Reconstructing geometry from a single image has been a long standing goal in computer vision [30, 31], but only recently first promising results have been demonstrated [9, 14, 40]. The reason for this is the ill-posed nature of the task with many possible explanations for a single observation.

Like single image depth prediction techniques, we also utilize only a single camera. However, in contrast to purely appearance based methods, we also exploit the structure of a point pattern from an extrinsic calibrated projector in addition to the ambient information in the image.

3. Active Monocular Depth Estimation

In this section, we first review the spatial structured light imaging principle and propose a forward model for generating images in this setting. Then, we describe the network architecture and the loss functions of our approach.

3.1. Spatial Structured Light

The operation principle of a monocular spatial structured light sensor [21, 26, 39] is illustrated in Fig. 1. Light emitted by a laser diode gets focused by a lens and dispersed into multiple random rays via a diffractive optical element (DOE), yielding a simple random dot pattern projector. The pattern projected onto the object is perceived by a camera. The projector can be regarded as a second camera with its virtual image plane showing the reference pattern determined by the DOE. As random patterns are locally unique, correspondences can be established between the perceived image and the virtual image of the projector using classical window-based matching techniques. Given a match and assuming rectified images, the disparity d can be calculated as the difference between x -coordinates of the corresponding pixels in the perceived image and the reference pattern. In this paper we follow an alternative approach and pose disparity estimation as a regression problem conditioned on the input image. Given the disparity d , the scene depth z can be obtained as $z = bf/d$, where b denotes the baseline and f is the focal length of the camera.

Forward Model: We now introduce our mathematical image formation model for a spatial structured light system. Let $\mathbf{I} \in \mathbb{R}^{W \times H}$ denote the image perceived by the camera, $W \times H$ being the image dimensions. We assume that the noisy image \mathbf{I} is obtained from a noise-free image $\mathbf{J} \in \mathbb{R}^{W \times H}$ by adding Gaussian noise with affine, signal dependent variance [12]. The noise-free image \mathbf{J} itself comprises two components: the reflected laser pattern $\mathbf{R} \in \mathbb{R}^{W \times H}$ and an ambient image $\mathbf{A} \in \mathbb{R}^{W \times H}$, which captures reflected light from other sources. Assuming Lambertian reflection, the intensity of the reflected pattern \mathbf{R} depends on the projection pattern $\mathbf{P} \in \mathbb{R}^{W \times H}$, the distance to the object $\mathbf{Z} \in \mathbb{R}^{W \times H}$, the reflectivity of the material $\mathbf{M} \in \mathbb{R}^{W \times H}$ and the orientation of the surface with respect to the light source Θ [29]. Overall, we obtain:

$$\begin{aligned}
 I(\mathbf{x}) &\sim \mathcal{N}(J(x, y), \sigma_1^2 J(x, y) + \sigma_2^2) \\
 J(x, y) &= A(x, y) + R(x, y) \\
 R(x, y) &= \frac{P(x, y)M(x, y) \cos(\Theta(x, y))}{Z(x, y)^2}.
 \end{aligned} \tag{1}$$

Here, we assume quadratic attenuation with respect to the distance of the object from the light source. Strictly speaking, quadratic attenuation is only true for point light sources. However, similar attenuation can be assumed for a laser projector due to divergence of the laser beams.

We leverage this model in Section 4.1 for simulating the image generation process when synthesizing scenes based on 3D CAD models. We also make use of it to inform our decisions for disentangling \mathbf{I} into an ambient and a point pattern component. Disentangling the two components has advantages: The ambient image comprises *dense* information about depth continuities which often align with the boundaries of the ambient image. The point pattern, on the other side, carries *sparse* information about the absolute depth at the projected points. Our model is thus able to improve depth boundaries compared to the traditional approach which considers only the sparse point pattern.

3.2. Network Architecture

We pose disparity estimation as a regression problem which we model using a fully convolutional network architecture. Supervised training is impractical for active depth prediction models as obtaining ground-truth depth with an accuracy significantly higher than the accuracy of the model itself is challenging. Therefore, we train our model using photometric, disparity and geometric constraints.

Our *photometric loss* enforces consistency between the input image and the warped reference pattern via the estimated disparity map. Our *disparity loss* models first-order (e.g., gradient) statistics of the disparity map conditioned on the edges of the latent ambient image. Our *geometric loss* enforces consistency of the 3D geometry re-

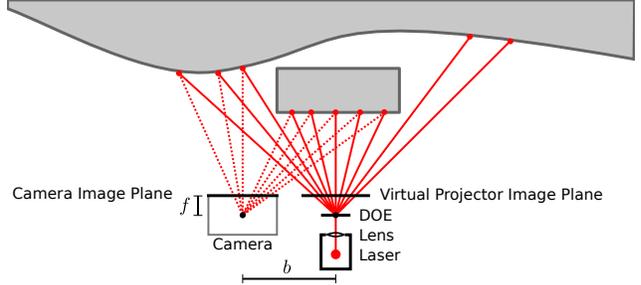


Figure 1: **Spatial Structured Light.** Coherent light is emitted by a laser diode. A diffractive optical element (DOE) splits the ray (solid red lines) and projects a random dot pattern into the scene. The dot pattern is perceived by a camera (dashed red lines) at baseline b . Given the uniqueness of random dot patterns in a local region, correspondences can be established and depth is inferred via triangulation.

constructed from two different views. Note that in contrast to self-supervised single image depth estimation techniques [14, 38, 43] which use photometric losses across viewpoints, we use a geometric loss across viewpoints as the scene changes with the location of the projector. Instead, we exploit photometric constraints to correlate the observation with the reference pattern. Our experiments (Section 4) demonstrate that all three losses are complementary and yield the best results when applied in combination.

Our overall model is illustrated in Fig. 2. As the geometric loss (green box) requires access to depth estimates from two different vantage points, we show two instances of the same network (red box and blue box), processing input image \mathbf{I}_i and input image \mathbf{I}_j , respectively. The parameters of the model are depicted in yellow. The disparity decoder and edge decoder parameters are shared across all training instances. The relative camera motion between any two views (i, j) is unique to a specific image pair and thus not shared across training instances. We now describe all components of our model in detail.

Image Preprocessing: As shown in Eq. 1, the camera image \mathbf{I} depends on various factors such as the ambient illumination \mathbf{A} , as well as the reflected pattern \mathbf{R} which in turn depends on the materials \mathbf{M} of the objects in the scene, the depth image \mathbf{Z} and the projected dot pattern \mathbf{P} . To mitigate the dependency of the reflected pattern \mathbf{R} from material \mathbf{M} and scene depth \mathbf{Z} , we exploit local contrast normalization [18, 41]:

$$\hat{\mathbf{P}} = \text{LCN}(\mathbf{I}, x, y) = \frac{I(x, y) - \mu_{\mathbf{I}}(x, y)}{\sigma_{\mathbf{I}}(x, y) + \epsilon}. \tag{2}$$

Here, $\mu_{\mathbf{I}}(x, y)$ and $\sigma_{\mathbf{I}}(x, y)$ denote mean and standard deviation in a small region (11×11 in all experiments) around (x, y) , and ϵ is a constant to eliminate low-level sensor noise

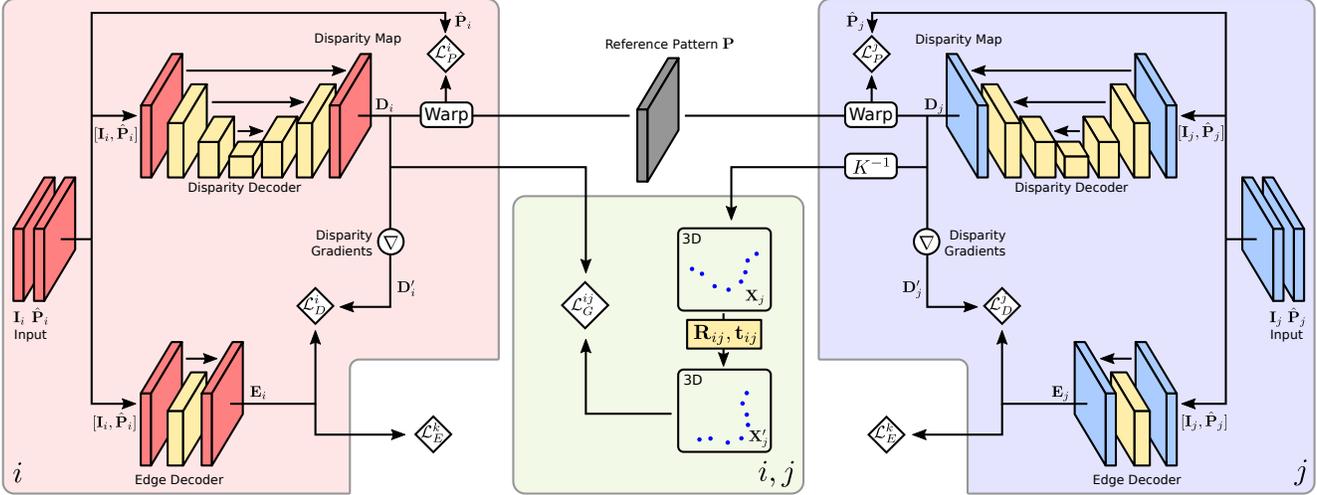


Figure 2: **Model Overview.** Input images I_i and I_j taken from two different viewpoints are processed to yield disparity maps D_i and D_j , respectively. The photometric loss \mathcal{L}_P and the disparity loss \mathcal{L}_D are applied *separately* per training image (here i and j). The geometric loss \mathcal{L}_G is applied to *pairs* of images (i, j) and measures the geometric agreement after projecting the 3D points from view j to view i given the relative motion $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ between the two views. The yellow colored boxes depict trainable parameters of our model. The disparity decoder and edge decoder parameters are shared across all training images. In contrast, one set of rigid motion parameters $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ is instantiated per training image pair (i, j). The operators are abbreviated as follows. “Warp”: bilinear warping of reference pattern via estimated disparity map, Δ : gradient magnitude, K^{-1} : projection of disparity map into 3D space based on known camera intrinsics.

and avoid numerical instabilities. While parts of the ambient illumination \mathbf{A} remain present in $\hat{\mathbf{P}}$, the strength of the ambient illumination is typically weaker than the intensity of the laser pattern and thus can be safely ignored when estimating depth from $\hat{\mathbf{P}}$.

Disparity Decoder: We concatenate the original image with the contrast normalized image and pass it to a disparity decoder which predicts a disparity map from the input. We use disparity instead of depth as output representation since disparity directly relates to image-based measurements, in contrast to depth. Surprisingly, we found that predicting disparity is easier compared to predicting absolute location [10] in the self-supervised setting. We provide an empirical analysis and further insights on this topic in our experimental evaluation.

The architecture of our decoder is similar to the U-net architecture proposed in [14, 22], interleaving convolutions with strided convolutions for the contracting part, and up-convolutions with convolutions for the expanding part. We use ReLUs [27] between convolution layers and skip-connections to retain details. The final layer is followed by a scaled sigmoid non-linearity which constrains the output disparity map to range between 0 and d_{max} . More details about our architecture can be found in the supplementary.

Edge Decoder: As the point pattern to supervise the disparity decoder is relatively sparse (see Fig. 4), the photo-

metric loss alone is not sufficient for learning to predict accurate and sharp object boundaries. However, information about the object boundaries is present in the ambient component of the input image. In particular, it is reasonable to assume that disparity gradients coincide with gradients in the ambient image (but not vice-versa) as the material, geometry and lighting typically varies across objects.

We exploit this assumption using an edge decoder which predicts ambient image edges \mathbf{E}_i directly from the input image I_i . Motivated by the fact that ambient edges can be well separated from the point pattern and other nuisance factors using local information, we exploit a shallow U-Net architecture for this task which enables generalization from few training examples. The final layer of this U-Net is followed by a sigmoid non-linearity which predicts the probability of an ambient edge at each pixel. Details about the network architecture are provided in the supplementary.

3.3. Loss Function

We now describe our loss function which is composed of four individual losses (illustrated with \diamond in Fig. 2): a photometric loss \mathcal{L}_P , a disparity loss \mathcal{L}_D , an edge loss \mathcal{L}_E and a geometric consistency loss \mathcal{L}_G . While \mathcal{L}_P , \mathcal{L}_D and \mathcal{L}_E operate on a single view i , the geometric loss \mathcal{L}_G requires pairs of images (i, j) as it encourages agreement of the predicted 3D geometry from multiple different views.

Let \mathcal{D} denote a training set of short video clips recorded

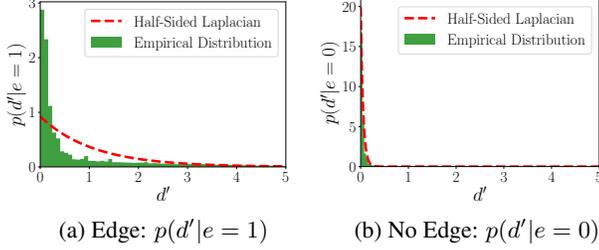


Figure 3: **Disparity Gradient Magnitude.** Empirical distribution (green) vs. fitted parametric distribution (red).

with a spatial structured light sensor and let $\mathcal{T} \in \mathcal{D}$ with $\mathcal{T} = \{\mathbf{I}_i\}_{i=0}^M$ be an element of \mathcal{D} . We call \mathcal{T} a track of length M and \mathbf{I}_i the i 'th frame of track \mathcal{T} . Further, let \mathcal{E} denote a set of pairs (\mathbf{I}, \mathbf{A}) , with \mathbf{A} denoting the ambient image. The overall loss with respect to \mathcal{D} and \mathcal{E} is given by

$$\sum_{\mathcal{T} \in \mathcal{D}} \left[\frac{1}{M} \sum_{i \in \mathcal{T}} (\mathcal{L}_P^i + \mathcal{L}_D^i) + \frac{1}{M^2} \sum_{i,j \in \mathcal{T}} \mathcal{L}_G^{i,j} \right] + \sum_{k \in \mathcal{E}} \mathcal{L}_E^k, \quad (3)$$

neglecting the relative weighting factors between the different loss functions for clarity. Note that frames from the same track show the same static scene from different viewpoints while the geometry changes across tracks. We require this distinction as our geometric loss ($\mathcal{L}_G^{i,j}$) requires ≥ 2 images of the same scene. We now describe each of the loss functions involved in Eq. 3 in detail.

Photometric Loss: Let $\hat{\mathbf{P}}_i$ denote the contrast normalized input image \mathbf{I}_i , \mathbf{D}_i the predicted disparity map and \mathbf{P} the contrast normalized reference pattern as it would appear at the virtual image plane of the pattern projector (see Fig. 1). The photometric loss is defined as

$$\mathcal{L}_P^i(\hat{\mathbf{P}}_i, \mathbf{P}, \mathbf{D}_i) = \sum_{x,y} \|\hat{\mathbf{p}}_i(x,y) - \mathbf{p}(x - D_i(x,y), y)\|_C, \quad (4)$$

where $\hat{\mathbf{p}}_i(x,y)$ denotes a small patch of $\hat{\mathbf{P}}_i$ centered at (x,y) and $\|\cdot\|_C$ represents the smooth Census transform [15]. Our photometric loss is similar to the ones used in existing work on self-supervised optical flow [17, 23] or depth [14, 41] prediction, except that we warp the reference pattern instead of an image captured by a second camera.

Disparity Loss: The sparse random dot pattern \mathbf{P} does not sufficiently constrain the disparity estimation problem. This can be observed particularly at disparity boundaries where the training signal is insufficient to obtain sharp and accurate object boundaries as evidenced by our experiments. While the dot pattern does not provide cues about edge boundaries, the ambient image does provide such complementary information as disparity boundaries often coincide with edges in the ambient image. We therefore model the correlations between the predicted edge map \mathbf{E} and the gradient magnitude $\mathbf{D}' = |\nabla \mathbf{D}|$ in our disparity loss.

Let $d' = |\nabla D(x,y)|$ denote the magnitude of the disparity gradient at pixel (x,y) . Let further $e \in \{0,1\}$ be a binary random variable representing the presence ($e=1$) or absence ($e=0$) of an edge in the ambient image. We assume a Bernoulli distribution over e , $p(e) = \lambda^e(1-\lambda)^{1-e}$, parameterized by the edge decoder $\lambda = E(x,y) \in [0,1]$. Furthermore, we assume that $p(d'|e)$ is modeled by a half-sided Laplacian distribution with location parameter $\mu=0$:

$$p(d'|e) = [d' \geq 0] \frac{\exp(-|d'|/b_e)}{b_e}. \quad (5)$$

The joint distribution over d' and e factorizes as follows:

$$p(d', e) = \frac{\exp(-|d'|/b_0)}{b_0} p_{e=0} + \frac{\exp(-|d'|/b_1)}{b_1} p_{e=1}. \quad (6)$$

To take this distribution into account, we formulate our disparity loss as the negative log probability density:

$$\mathcal{L}_D^i(\mathbf{D}'_i, \mathbf{E}_i) = \sum_{x,y} -\log(p(D'_i(x,y), E_i(x,y))). \quad (7)$$

Empirically, we observed that the heavy-tailed half-sided Laplacian distribution with location parameter $\mu=0$ models the conditional disparity gradient distribution sufficiently well. However, extending our model to other distributions is straightforward. In practice, we estimate b_0 and b_1 from a small set of images. Fig. 3 shows the empirical disparity gradient magnitudes (green) and the half-sided Laplacian fit (red) to this distribution for the edge and non-edge case. As expected, the probability for larger disparity gradients is higher in the edge case than in the non-edge case, demonstrating the dependency between d' and e which we exploit in our disparity loss \mathcal{L}_D .

Edge Loss: Training the edge decoder based on \mathcal{L}_D alone would lead to the trivial solution $E(x,y) = 1$ for all pixels. We thus introduce a cross-entropy edge loss \mathcal{L}_E^k that regularizes the decoded edges \mathbf{E}_k against the boundaries of the ambient image \mathbf{A}_k corresponding to input image \mathbf{I}_k :

$$\mathcal{L}_E^k(\mathbf{E}_k, \mathbf{A}_k) = - \sum_{x,y} A'_k(x,y) \log E_k(x,y) + w(1 - A'_k(x,y)) \log(1 - E_k(x,y)). \quad (8)$$

Here, $\mathbf{A}'_k = \text{LCN}_\epsilon(|\nabla \mathbf{A}_k|)$ denotes the local contrast normalized gradient magnitude of the ambient image \mathbf{A}_k and w is a weight factor to account for the distribution imbalance (edges occur less frequent than non-edges). Gradients in the ambient image are typically weaker than gradients in the disparity image. We therefore use the contrast normalized gradient magnitude $\text{LCN}(|\nabla \mathbf{A}_k|)$ instead of $|\nabla \mathbf{A}_k|$ to emphasize weak edges in the ambient image.

While \mathcal{L}_E^k requires supervision in terms of the ambient image \mathbf{A} , it is important to note that ground-truth for the ambient image \mathbf{A} is much easier to obtain than supervision

for the disparity map \mathbf{D} . While for the latter, very accurate depth sensors (e.g., laser scanners) and precise pose estimates are required, the former can be obtained by collecting images of static scenes using a tripod, capturing pairs of images with the projector turned on and off. Furthermore, predicting ambient edges from the input image is a relatively simple task compared to disparity estimation. We thus use a shallow network with a small number of parameters for the edge decoder which is less prone to overfit, even when provided with only a moderate number of training images. Details are provided in the supplementary.

Geometric Loss: Additional supervision can be incorporated by considering consistency of the predicted geometry across multiple views. Towards this goal, we convert the disparity map of a second view \mathbf{D}_j into a 3D point cloud $\mathbf{X}_j = (\mathbf{x}_{j,1}, \dots, \mathbf{x}_{j,HW})$ using the differentiable inverse projection equation $\mathbf{x} = \frac{bf}{d} \mathbf{K}^{-1}(x, y, 1)^\top$ where (x, y) denotes pixel location, d disparity, b baseline, f focal length and \mathbf{K} camera intrinsics. Next, we transform the point cloud \mathbf{X}_j into the camera coordinate system of the first view i using the rigid motion parameters $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$. Let us denote the transformed point cloud as $\mathbf{X}'_j = \mathbf{R}_{ij}\mathbf{X}_j + \mathbf{t}_{ij}$. Our geometric loss compares the depth of each point in \mathbf{X}'_j to its corresponding depth value in the first view:

$$\mathcal{L}_G^{ij}(\mathbf{D}_i, \mathbf{X}'_j) = \sum_{\mathbf{x} \in \mathbf{X}'_j} \min(|\mathbf{x}_z - bfD_i^{-1}(\mathbf{K}\mathbf{x})|, \tau). \quad (9)$$

Here, \mathbf{x}_z denotes the z -component (= depth) of 3D point \mathbf{x} , and τ is a truncation threshold which accounts for differences in the point sets due to occlusions. Furthermore, $D_i^{-1}(\mathbf{K}\mathbf{x})$ denotes the inverse disparity at pixel $\mathbf{K}\mathbf{x}$, i.e., the projection of 3D point \mathbf{x} into the i 'th camera view. Note that this loss is applied in both directions $i \leftrightarrow j$ in Eq. 3.

3.4. Training and Inference

We first extract the reference pattern \mathbf{P} by pointing the laser projector at a white wall, warping the resulting image into the virtual projector image plane and applying local contrast normalization. We then train our model in two stages. First, we pre-train the disparity and edge decoder without the geometric loss. In the second stage, we train the entire model using all losses specified in Eq. 3. We use ADAM [20] and a learning rate of 10^{-4} . At inference time, we only retain the disparity decoder to obtain disparity map \mathbf{D}_i from a previously unseen test image \mathbf{I}_i . We select the network parameters from all training epochs by minimizing the photometric error on the training set, as we found this to correlate well with our test metrics. See supplementary for details.

4. Experimental Evaluation

In this section we systematically evaluate our method. We first introduce our structured light renderer in Sec-

tion 4.1 that is used to generate synthetic, but plausible images with accurate ground-truth disparity maps. In Section 4.2 this synthetic dataset is utilized in ablation studies to quantify the influence of our design choices. We further compare our method on this dataset to simple baselines and state-of-the-art methods. Finally, to demonstrate the effectiveness of our method on real structured light data, we evaluate it in Section 4.3 on the dataset provided by [5].

4.1. Structured Light Renderer

To accurately evaluate various design choices and compare different methods against each other, we require data with precise ground-truth. While such data can be obtained with laser scanners [5], those are typically expensive, slow and do not scale to larger datasets. Additionally, they require large efforts for aligning the predictions with the scanned 3D model. An alternative approach is to use volumetric fusion [8, 28] to generate ground-truth data that is at least more complete than the individual depth scans. However, this bears the problem of limited accuracy and working range [24]. Hence, for the first part of our experimental evaluation, we resort to synthetically generated scenes using a custom structured light renderer.

Towards this goal, we follow the principles discussed in Section 3.1. For simplicity, let us assume that the virtual camera is centered at the origin and looking towards the positive z -direction. We first cast a ray $\mathbf{r}_C = \mathbf{K}_C^{-1}(x, y, 1)^T$ for each pixel (x, y) of the image. Along this ray direction, we probe if any triangle has been hit. If not, we return a value indicating an invalid ground-truth depth and set the image intensity to black. Otherwise, we return the ground-truth depth as distance from camera to the ray-triangle intersection \mathbf{x} . Next, we compute the ray $\mathbf{r}_P = \frac{\mathbf{x} - \mathbf{c}_P}{\|\mathbf{x} - \mathbf{c}_P\|_2}$ that emits from the projector center \mathbf{c}_P to the 3D point \mathbf{x} and test if (a) it is occluded by any triangle in front of \mathbf{x} and (b) the ray is still inside the virtual image plane of the projector. Given that both criteria are met, we obtain the pattern intensity from the reference pattern using bilinear interpolation. We further apply a simple Blinn-Phong model [2] for shading the ambient image \mathbf{A} . To combine the ambient image \mathbf{A} with the interpolated pattern we implemented Eq. 1 with the simplification that we assume constant material reflection \mathbf{R} and light source orientation Θ for all scenes.

Using this structured light renderer we generate 8,448 short sequences, 8,192 for training, and 256 for testing. Each short sequence consists of four renderings with the camera center randomly translated within a $20 \times 20 \times 20\text{cm}$ box and the camera looking towards the center of the scene. We use the camera matrix of the Kinect V1 sensor and the original reference pattern. The baseline between camera and projector is set to 7.5cm and the image size is 640×480 pixels. To populate the scene with objects, we take a subset of chair meshes from the ShapeNet Core dataset [4],

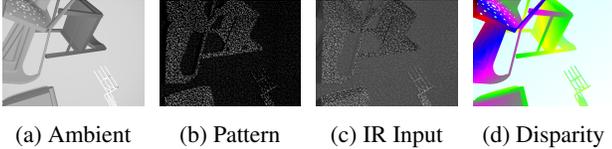


Figure 4: **Structured Light Rendering Example.** (a) Ambient image with Blinn-Phong shading. (b) Warped pattern with intensities dependent on scene depth. (c) Blending of the ambient image with the pattern produces the final image I . (d) Ground-truth disparity map for this scene.

randomly scaled and rotated, placed at a distance between $2 - 3m$. In addition, we add a randomly slanted background plane to the scene at a distance between $2 - 7m$. Fig. 4 shows an example output of our rendering pipeline.

4.2. Evaluation on Rendered Data

The synthetic dataset introduced in the previous section is used in our first set of experiments, where we demonstrate the influence of our design choices.

Metrics: To quantitatively compare our results, we use the percentage of outliers $o(t)$ as metric. We compute the difference between the estimated and ground-truth disparity map and evaluate the percentage of pixels where the disparity difference is greater than a certain threshold t .

Ablation Study: In our first evaluation, we demonstrate the effectiveness of our fully convolutional architecture for disparity estimation. To isolate the effect of the representation from the choice of (self-supervised) loss function, we conduct this experiment in the fully supervised setting on our synthetic dataset. Table 1 shows our results when directly estimating the disparity ($I \rightarrow D$), classifying the absolute location of the local patch in the reference pattern along the x -axis such as in [10] ($I \rightarrow P$), and using independent regression layers per row ($I \rightarrow D_{row}$ and $I \rightarrow P_{row}$). In addition, we also evaluate a passive monocular depth estimation model on our synthetic dataset. We realize this by predicting disparity directly from the ambient image ($A \rightarrow D$). We find that directly estimating the disparity based on the input I works best, followed by estimating the position. Note, that the position is estimated using regression, as a classification formulation would lead to an impractical large output space, i.e., 640 classes per pixel. We further observe that the network is not able to learn accurate disparity from the ambient image alone which validates the utility of the active depth estimation setup.

To demonstrate the contribution of the individual loss terms introduced in Section 3.3, we start training our network with the photometric loss \mathcal{L}_P only, and incrementally add the disparity loss \mathcal{L}_D and the geometric loss \mathcal{L}_G . Note that the edge decoder is trained on a small subset of the dataset (see supplementary for details). Our results are summarized in Table 2. We observe that the photometric

	$o(0.5)$	$o(1)$	$o(2)$	$o(5)$
$I \rightarrow D$	6.22	3.00	1.63	0.85
$I \rightarrow D_{row}$	8.19	4.35	2.40	1.07
$I \rightarrow P$	11.83	5.08	2.46	1.11
$I \rightarrow P_{row}$	58.12	28.81	8.02	2.11
$A \rightarrow D$	90.71	81.41	63.53	32.00

Table 1: **Architectural Choices.**

	$o(0.5)$	$o(1)$	$o(2)$	$o(5)$
Supervised	6.22	3.00	1.63	0.85
\mathcal{L}_P	10.92	6.00	4.10	2.72
$\mathcal{L}_P + \mathcal{L}_D$	8.67	4.23	2.56	1.52
$\mathcal{L}_P + \mathcal{L}_D + \mathcal{L}_G$	6.77	3.88	2.57	1.63

Table 2: **Influence of Loss Function.**

	$o(0.5)$	$o(1)$	$o(2)$	$o(5)$	$o_u(1)$	$o_u(5)$
Block Matching	7.84	7.20	7.06	6.83	4.44	4.23
FastMRF [5]	12.07	8.36	6.71	5.14	5.25	3.57
HyperDepth [10]	15.01	12.63	11.83	11.49	7.39	6.73
Ours	6.77	3.88	2.57	1.63	1.75	0.70

Table 3: **Quantitative Results on Synthetic Data.**

loss alone, unsurprisingly, performs worse than the supervised trained network. However, if we add the disparity loss, we significantly improve the results. If we also add the geometric loss (with truncation value set to $\tau = 0.01m$) we observe a further increase in performance, especially for small details captured in the metrics $o(0.5)$ and $o(1)$. Fig. 5 shows a qualitative comparison when incrementally adding the loss terms. The network is able to learn the overall shape with only supervision from the photometric loss \mathcal{L}_P , but the prediction is noisy and leads to bleeding at the edges. With the edge information encoded in the disparity loss \mathcal{L}_D , the network is able to extrapolate disparities correctly into occluded region. The geometric loss \mathcal{L}_G further reduces errors, in particular in large homogeneous areas.

Baseline Comparison: After verifying our design choices, we compare our full model to several baselines: OpenCV block matching [3], FastMRF [5] and HyperDepth [10]. For FastMRF we use the implementation provided by the authors. For HyperDepth we use our own re-implementation using the same hyper-parameters as proposed in [10]. However, we used deeper trees as we found that this to yield better performance. Note that we use ground truth disparity maps for training HyperDepth.

The results on our synthetic dataset are summarized in Table 3 and qualitative results are provided in Fig. 5. We observe that our method yields overall the best results. HyperDepth yields qualitatively good results in smooth regions, but fails at discontinuities, leading to worse overall results. In the last two columns of Table 3, we further evaluate our method on novel object categories in ShapeNet Core (cars, airplanes and watercrafts) unseen during training, demonstrating the generalization of our method on unseen objects.

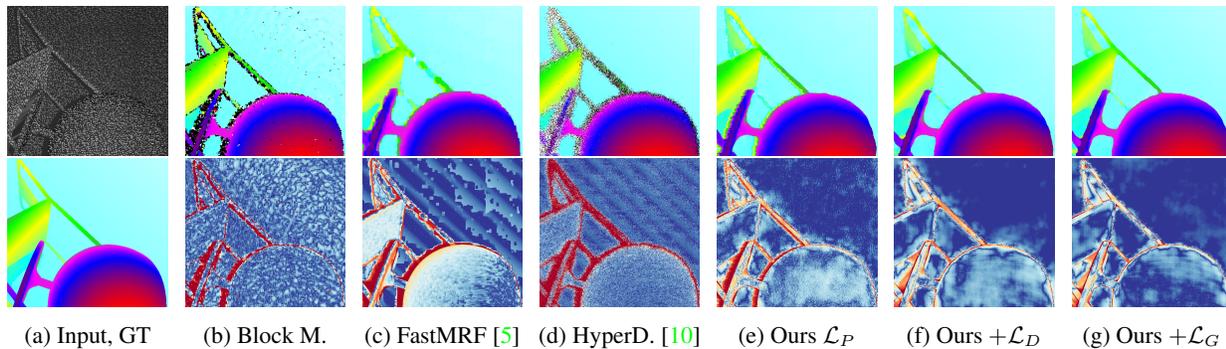


Figure 5: **Qualitative Results on Synthetic Data.** See text for details.

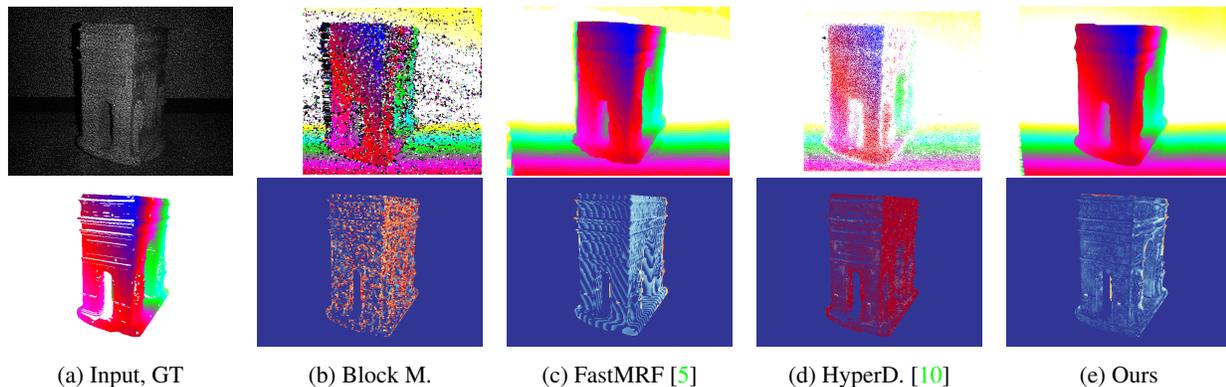


Figure 6: **Qualitative Results on Real Data.** See text for details.

	acc	comp	h. mean
Block Matching	551.082	3.883	7.712
FastMRF [5]	12.690	6.971	8.999
HyperDepth [10]	8.759	5.263	6.575
Ours	11.042	3.147	4.898

Table 4: **Quantitative Results on Real Data.** h. mean denotes the harmonic mean of accuracy and completeness.

4.3. Evaluation on Real Data

For quantitative evaluation on real data, we use the dataset of Chen et al. [5] which consists of 5 accurately scanned models (see supplementary for qualitative evaluations in more complex real world scenarios), along with sequences recorded using a PrimeSense Carmine range sensor by placing the objects at a distance of about $1m$ and rotating them around the up-axis in 30° steps. To align the 3D models with the estimated depth maps, we follow the protocol described in [5]. First, we obtain a rough alignment by manually matching the model mesh with the point-cloud from the estimated depth. This is then refined with a Point-Plane-ICP as implemented in Open3D [42]. For the quantitative evaluation we remove points from the model that are not visible in the given view and project the estimated depth map to 3D. Given the estimated and the ground truth point cloud, we compute accuracy as the mean distance from the estimated 3D points to the closest ground-truth points and completeness vice-versa. In addition to accuracy and com-

pleteness, we report the harmonic mean of those two metrics. We compare to the same methods as in our previous experiment on the synthetic data, with HyperDepth trained on the block matching results as pseudo ground-truth. As the dataset contains only 60 images in total, HyperDepth is trained on all images it is also evaluated on. Table 4 summarizes our quantitative results (numbers in mm). Qualitative results are provided in Fig. 6. HyperDepth yields slightly better accuracy as it aggressively masks bad pixels in post-processing, but has worse completeness results, whereas our method yields a good trade-off between accuracy and completeness and achieves the overall lowest harmonic mean.

5. Conclusion

We have presented a novel method for estimating depth using an active monocular camera. By combining photometric and geometric information, we were able to train a neural network in a self-supervised fashion on this problem. In addition, we demonstrated significant improvements when conditioning disparity gradients on edge information extracted from the ambient component of the input image using weak supervision. Our results indicate that weakly supervised training yields results similar to the same network trained on ground-truth data. Moreover, our method compares favorably to the state-of-the-art.

Acknowledgement: This work was supported by the Intel Network on Intelligent Systems.

References

- [1] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proc. of the British Machine Vision Conf. (BMVC)*, 2011. 2
- [2] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *ACM Trans. on Graphics*, 1977. 6
- [3] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008. 7
- [4] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *arXiv.org*, 1512.03012, 2015. 6
- [5] Q. Chen and V. Koltun. Fast MRF optimization with application to depth reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6, 7, 8
- [6] S. Choi, Q. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv.org*, 1602.02481, 2016. 1
- [7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1
- [8] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. 2017. 6
- [9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 2
- [10] S. R. Fanello, C. Rhemann, V. Tankovich, A. Kowdle, S. Orts-Escolano, D. Kim, and S. Izadi. Hyperdepth: Learning depth from structured light without matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 4, 7, 8
- [11] S. R. Fanello, J. P. C. Valentin, C. Rhemann, A. Kowdle, V. Tankovich, P. L. Davidson, and S. Izadi. Ultrastereo: Efficient learning-based matching for active stereo systems. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [12] A. Foi, M. Trimeche, V. Katkovnik, and K. O. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. on Image Processing (TIP)*, 17(10):1737–1754, 2008. 3
- [13] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. W. Fitzgibbon. Efficient Regression of General-Activity Human Poses from Depth Images. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2011. 1
- [14] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 4, 5
- [15] D. Hafner, O. Demetz, and J. Weickert. Why is the census transform good for robust optic flow computation? In *Proc. of the International Conf. on Scale Space and Variational Methods in Computer Vision (SSVM)*, 2013. 5
- [16] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341, 2008. 1, 2
- [17] J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 5
- [18] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2009. 3
- [19] A. Kendall, H. Martirosyan, S. Dasgupta, and P. Henry. End-to-end learning of geometry and context for deep stereo regression. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2
- [20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015. 6
- [21] M. Martinez and R. Stiefelwagen. Kinect unleashed: Getting control over high resolution depth maps. In *Machine Vision and Applications (MVA)*, 2013. 2
- [22] N. Mayer, E. Ilg, P. Haeusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4
- [23] S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2018. 5
- [24] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, and D. Kondermann. When Can We Use KinectFusion for Ground Truth Acquisition. In *Proc. Workshop on Color-Depth Camera Fusion in Robotics*, 2012. 6
- [25] M. Menze, C. Heipke, and A. Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 140:60–76, 2018. 1, 2
- [26] J. L. Moigne and A. M. Waxman. Structured light patterns for robot mobility. *IEEE Journal of Robotics and Automation (JRA)*, 4(5):541–548, 1988. 2
- [27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. of the International Conf. on Machine Learning (ICML)*, 2010. 4
- [28] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 1, 6
- [29] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 3
- [30] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems (NIPS)*, 2006. 2
- [31] A. Saxena, M. Sun, and A. Y. Ng. Make3D: learning 3D scene structure from a single still image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31:824–840, 2009. 2

- [32] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nestic, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proc. of the German Conference on Pattern Recognition (GCPR)*, 2014. 1, 2
- [33] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47:7–42, 2002. 1, 2
- [34] T. Schöps, J. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1
- [35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGB-D images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2012. 1
- [36] X. Song, X. Zhao, H. Hu, and L. Fang. Edgestereo: A context integrated residual pyramid network for stereo matching. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2018. 2
- [37] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart. Towards a benchmark for rgb-d slam evaluation. In *Proc. Robotics: Science and Systems (RSS)*, 2011. 1
- [38] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv.org*, 1704.07804, 2017. 3
- [39] P. Vuylsteke and A. Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 12(2):148–164, 1990. 2
- [40] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [41] Y. Zhang, S. Khamis, C. Rhemann, J. P. C. Valentin, A. Kowdle, V. Tankovich, M. Schoenberg, S. Izadi, T. A. Funkhouser, and S. R. Fanello. Activestereonet: End-to-end self-supervised learning for active stereo systems. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 2, 3, 5
- [42] Q.-Y. Zhou, J. Park, and V. Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 8
- [43] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3