Self-Supervised Deep Depth Denoising

Vladimiros Sterzentsenko [*] Leonidas Saroglou [*] Anargyros Chatzitofis [*]
Spyridon Thermos [*] Nikolaos Zioulis [*] Alexandros Doumanoglou
Dimitrios Zarpalas Petros Daras
Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), Greece

Abstract

Depth perception is considered an invaluable source of information for various vision tasks. However, depth maps acquired using consumer-level sensors still suffer from non-negligible noise. This fact has recently motivated researchers to exploit traditional filters, as well as the deep learning paradigm, in order to suppress the aforementioned non-uniform noise, while preserving geometric details. Despite the effort, deep depth denoising is still an open challenge mainly due to the lack of clean data that could be used as ground truth. In this paper, we propose a fully convolutional deep autoencoder that learns to denoise depth maps, surpassing the lack of ground truth data. Specifically, the proposed autoencoder exploits multiple views of the same scene from different points of view in order to learn to suppress noise in a self-supervised end-to-end manner using depth and color information during training, yet only depth during inference. To enforce self-supervision, we leverage a differentiable rendering technique to exploit photometric supervision, which is further regularized using geometric and surface priors. As the proposed approach relies on raw data acquisition, a large RGB-D corpus is collected using Intel RealSense sensors. Complementary to a quantitative evaluation, we demonstrate the effectiveness of the proposed self-supervised denoising approach on established 3D reconstruction applications. Code is avalable at https://github.com/VCL3D/DeepDepthDenoising

1. Introduction

Depth sensing serves as an important information cue for all vision related tasks. Upon the advent of consumer grade depth sensors, the research community has exploited the availability of depth information to make performance leaps in a variety of domains. These include SLAM technology for robotics navigation, static scene capture or tracking for augmented reality applications [42], dynamic human performance capture [2], autonomous driving [7].

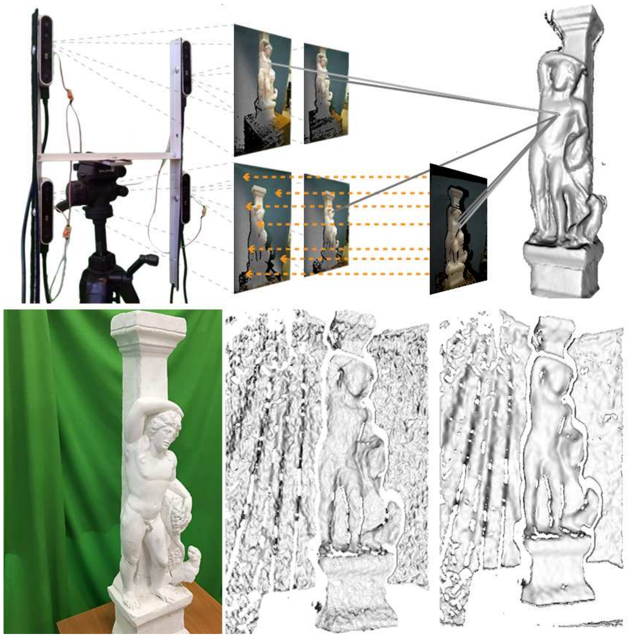

Figure 1. An abstract representation of the proposed method. Our model exploits depth-image-based rendering in a multi-view setting to achieve self-supervision using photometric consistency and geometrical and surface priors. A denoising example is visualized in the lower part (right-most), compared to a traditional filtering result (middle).

Depth sensors can be categorized based on either their interaction with the observed scene in *passive* (pure observation) and *active* (observation after actuation), or their technological basis in *stereo*, *structured light* (SL) and *time-of-flight* (ToF) respectively. While the latter two are active by definition, stereo-based sensors can operate in both passive and active mode as they estimate depth via binocular observation and triangulation. Given that they are driven by correspondence establishment, the active projection of textured patterns into the scene improves performance in low textured areas. However, the aforementioned sensor types suffer from high levels of noise and structural artifacts.

[*] Equal contribution

Most works that aim to address noisy depth estimations rely on using traditional filtering methods [30, 48], explicit noise modeling [37, 16, 3], and the exploitation of the Deep Learning (DL) paradigm in terms of deep denoising autoencoders. However, the former two require extensive parameter tuning to properly adapt to different levels of noise, struggle to preserve details, and lead to local (sensor-specific) solutions. On the other hand, recent studies utilizing deep autoencoders [18, 46] are able to capture context and lead to more global solutions. The main challenge with the data-driven approaches is that finding ground truth for supervision is a hard, time-consuming, usually expensive process and sometimes impossible. Although more recent unsupervised data-driven approaches [31] try to address the ground truth drawback, they rely on assumptions for the noise nature and properties, which do not apply to consumer level depth sensors.

In this work, the DL paradigm is adopted to address both the lack of ground truth data, as well as the necessity to investigate denoising without a priori assumptions. A fully-convolutional deep autoencoder is designed and trained following a self-supervised approach. In particular, self-supervision relies on simultaneously capturing the observed scene from different viewpoints, using multiple RGB-D sensors placed in a way that their fields of view (FoV) overlap. The color information acquired by the sensors is used for synthesizing target view using the predicted depth maps given known sensor poses. This process enables direct photometric supervision without the need for ground truth depth data. Depth and normal smoothness priors are used for regularization during training, while our inference only requires a single depth map as input. The model is trained and evaluated on a corpus collected with the newest Intel RealSense sensors [20] and consists of sparse data with high depth variation. However, note that on inference, the model can be applied to any consumer-level depth sensor. An overview of our method, along with a denoising example are depicted in Fig. 1.

Extensive quantitative evaluation demonstrates the effectiveness of the proposed self-supervised denoising method compared to state-of-the-art methods. Additionally, the performance of the deep autoencoder is further evaluated qualitatively by using the denoised depth maps in well-established 3D reconstruction applications, showcasing promising results given the noise levels of the utilized sensors. Note that the model structure enables efficient inference on recent graphics cards.

2. Related Work

Each depth sensing technology is affected with distinct systematic noise, a fact that renders the development of universal depth denoising methods a challenging task. In the following overview, related work is divided in three major categories, presenting state-of-the-art depth denoising approaches available in the literature.

Noise modeling. As depth sensors operate on different principles, they are also affected by different systematic noise that is unique to their underlying operation. As a result, one approach of addressing the levels of noise in depth maps is to model the underlying sensor noise. The initial work of [16] modeled Kinect's systematic noise into a scale and a distortion component, and was solved as a combined problem of noise modeling, extrinsic and intrinsic calibration, using planar surfaces and a checkerboard. In addition to denoising, [37] also performed depth map completion on data produced by a SL depth sensor. A probabilistic framework for foreground-background segmentation was employed, followed by a neighbourhood model for denoising which prevented depth blurring along discontinuities. A similar approach was recently proposed by [3], with the key difference being a polynomial undistortion function which was estimated in a finer granularity at the pixel level rather than a closed form equation. However, the heterogeneity of sensor noise models is difficult to generalize and apply in a variety of sensors. A prominent example is a bulk of recent work that deals with the noise inducing multiple path interference (MPI) issue of ToF sensors [12], [29] and [1]. They employ DL methods to correct and denoise the generated depth data, but these approaches are not applicable to other sensor types.

Classical and Guided Filtering. Traditional filtering approaches are more applicable to a variety of sensor types, with the most typical approach for depth denoising in various applications (*e.g.* [32]) being the bilateral filter [43], a well established computer vision filter. From a more practical standpoint, as depth sensors are typically accompanied by at least one light intensity sensor (color, infrared), many works have resorted to using this extra modality as a cleaner guidance signal to drive the depth denoising task. While indeed a promising approach, the use of intensity information relies on the aligned edge assumption between the two modalities, and as a result, both the joint bilateral [30] and rolling guidance [48] filters suffer from texture transfer artifacts. Thus, follow up works have focused on addressing this lack of structural correlation between the guide and the target images [38, 13, 28]. Finally, similar in concept approaches [33, 14, 45, 47] utilize shading information, extracted from the intensity images, in order to refine the acquired depth maps. Despite the increased robustness gained from surface information utilization, all aforementioned methods cannot alleviate from artifacts produced due to modalities misalignment. Additionally, the most significant drawback of typical filtering is its inability to understand the global context, thus operating on local level.

Learning methods. Data driven methods on the other hand, can better capture the global context of each scene, an important source of information that can drive the denoising task. The guided filtering concept has been implemented with convolutional neural networks (CNNs) in [10] and [26]. The former proposes a weighted analysis representation model in order to model the dependency between intensity and depth images, with a local weight function learned over labeled task-specific data. The latter currently represents the state-of-the-art in joint filtering. It uses 3 CNNs to learn to transfer structural information from the guiding image to the noisy one. While effective, it is learned in a fully supervised manner, meaning that it requires ground truth data, which are hard to obtain and would require collection for each different type of sensor. More recent works have resorted to near ground truth dataset generation in order to circumvent the lack of and difficulty in acquiring ground truth depth data. The ScanNet [9] dataset is used in [18] to produce raw-clean depth pairs by exploiting the implicitly denoised 3D reconstructed models and the known sensor poses during scanning, to synthesize them via rendering. A very deep multi-scale Laplacian pyramid based auto-encoder model is used and directly supervised with an additional gradient-based structure preserving loss. Despite the satisfactory results, inference is quite slow because of the depth of their network, making it unfeasible to use their model in real-world applications. Similarly, Kwon *et al.* [23] produce their raw-near ground truth pairs using [32], in order to train their multi-scale dictionary-based method. Additonally, Wu *et al.* [46] use a dynamic 3D reconstruction method [11] to non-rigidly fuse depth data and construct raw-clean depth map pairs. This work employs an auto-encoder with skip connections, coupled with a refinement network at the end that fuses the denoised data with intensity information to produce refined depth maps.

Unavailability and difficulty to generate ground-truth data in various contexts, is the major motivator for unsupervised methods. Noise2Noise [31] and its extensions Noise2Self [4] and Noise2Void [22] demonstrated how denoising can be achieved in an unsupervised manner without clean data. However the aforementioned approaches rely on certain distributional assumptions (*i.e.* zero-mean Gaussian i.i.d. noise), which do not apply on data acquired by consumer-level depth sensors. Evidently, methods for training without direct supervision are required. Our work addresses this issue by proposing the use of multiple sensors in a multi-view setting.

3. Depth Denoising

Our approach aims to circumvent the lack of ground truth depth data. An end-to-end framework, trained in the absence of clean depth measurements, learns to denoise the input depth maps. Using unstructured multi-view sensors that capture unlabelled color and depth data, our approach relies on view synthesis as a supervisory signal and, although utilizing color information during training, it requires only a single depth map as input during inference.

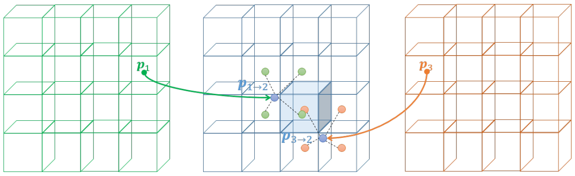

Figure 2. Our multi-view forward splatting scheme is illustrated. The source views - 1 and 3 (*green* and *orange* respectively) - splat their contributions to the target view - 2 (*blue*). Each source pixel ($\mathbf{p}_1$ and $\mathbf{p}_3$) reprojects to the target view ($\mathbf{p}_{1\to2}$ and $\mathbf{p}_{3\to2}$ respectively). The color information that they carry from their source views is spread over the neighborhood of their reprojections in a bilinear manner. In addition, these contributions are also weighted by each source pixel's confidence. As shown in the highlighted pixel of target view, multiple views combine their color information in the target splatted image.

3.1. Multi-view Self-Supevision

Each sensor jointly acquires a color image $\mathbf{I}(\mathbf{p}) \in \mathbb{R}^3$ and a depth map $D(\mathbf{p}) \in \mathbb{R}$, with $\mathbf{p} := (x, y) \in \Omega$ being the pixel coordinates in the image domain Ω defined in a $W \times H$ grid, with W and H being its width and height, respectively. Considering $\mathcal{V}$ spatially aligned sensors $v \in \{1, ..., \mathcal{V}\}$, whose viewpoint positions are known in a common coordinate system and expressed by their poses $\mathbf{T}_v := \left[\begin{smallmatrix}\mathbf{R}_v & \mathbf{t}_v \\ \mathbf{0} & 1\end{smallmatrix}\right]$, where $\mathbf{R}_v$ and $\mathbf{t}_v$ denote rotation and translation respectively, we can associate image domain coordinates from one viewpoint to another using:

$$\mathcal{T}_{s\to t}(\mathbf{p}_s) = \pi(\mathbf{T}_{s\to t}\pi^{-1}(D_s(\mathbf{p}_s), \mathbf{K}_s), \mathbf{K}_t), \qquad (1)$$

with $\mathbf{T}_{s\to t}$ being the relative pose between sensors s (source) and t (target), with the arrow showing the direction of the transformation. π and π^{-1} are the projection and deprojection functions that transform 3D coordinates to pixel coordinates and vice versa, using each sensor's intrinsics matrix $\mathbf{K}$. Note that we omit the depth map D_s, pose $\mathbf{T}_{s\to t}$ and the intrinsics $\mathbf{K}_s$ and $\mathbf{K}_t$ arguments from function $\mathcal{T}$ for notational brevity.

Under a multi-view context and given that each v sensor color image $\mathbf{I}_v$ and depth map D_v are aligned and defined on the same image domain Ω, color information can be transferred from each view to the other ones using Eq. 1 on a per pixel basis. Note that contrary to noisy depth measurements, the acquired color information can be considered as clean (or otherwise a much more consistent and higher quality signal). Traversing from one view to another via noisy depth will produce distorted color images due to depth errors manifesting into incorrect reprojections. Consequently, we can self-supervise depth noise through inter-view color reconstruction under the photoconsistency assumption.

Even though view synthesis supervision requires at least 2 sensors, more of them can be employed, as long as their

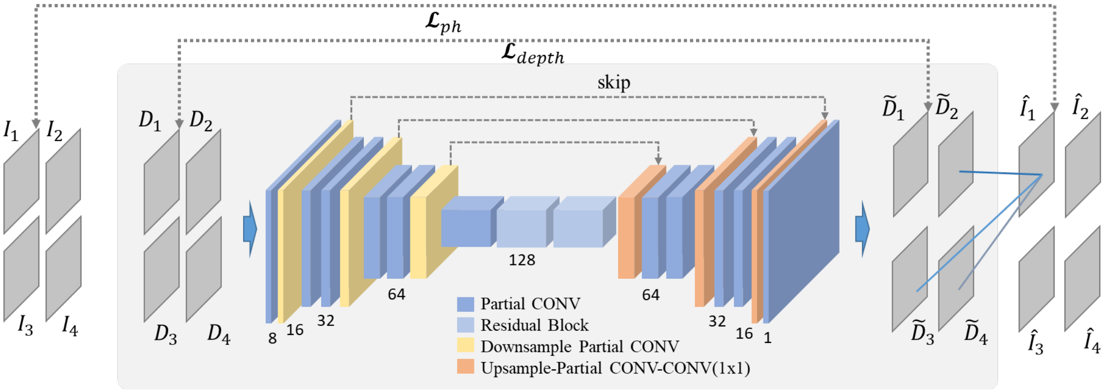

Figure 3. Detailed network architecture of the proposed depth denoising method. The network receives raw depth information from all available sensors ($D_1 - D_4$) and predicts denoised depthmaps ($\tilde{D}_1 - \tilde{D}_4$). Using differentiable rendering (see Section 3.1), a new target color image $\hat{\mathbf{I}}_1$ is synthesized from the non-target depth map predictions $D_2 - D_4$. Subsequently, $\hat{\mathbf{I}}_1$ is used to compute the L_{ph} loss (see Section 3.3), considering $\mathbf{I}_1$ as ground truth. Note that every input depth map is iteratively considered as target frame, while the total loss derives from the summation of each sensor loss.

poses in a common coordinate system are known, via the geometric correspondence function $\mathcal{T}$. This allows us to address apparent issues like occlusions and the limitations of a consistent baseline (restricted accuracy). Additionally, as the noise is inconsistent, multiple depth maps observing the same scene will simultaneously offer varying noise structure and levels, while increasing the diversity of the data. Intuitively, and similar to wide-baseline stereo, adding more sensors will offer higher reconstruction accuracy due to the variety of baselines. Note that since this approach is purely geometric, any number of unstructured sensor placements is supported.

Most works using view synthesis as a supervision signal utilize inverse warping [17] for image reconstruction. Under this reconstruction scheme, each target pixel samples from the source image, thus many target pixels may sample from the same source pixel. However, relying on erroneous depth values is problematic as occlusions and visibility need to be handled in an explicit manner via depth testing, itself relying on the noisy depth maps.

To overcome this, we employ differentiable rendering [44] and use forward splatting to accumulate color information to the target view. In forward splatting each source pixel accumulates its contribution to the target image, thus, as depicted in Fig. 2, many source pixels (from the same or even different views) may contribute to a single target pixel. This necessitates a weighted average accumulation scheme for reconstructing the rendered image. We define a splatting function $\mathcal{S}_{s\to t}(A_t, B_s, D_s, \mathbf{p}_s)$:

$$A_t(\mathcal{T}_{s\to t}(\mathbf{p}_s)) = w_c(D_s, \mathbf{p}_s)w_b(\mathcal{T}_{s\to t}(\mathbf{p}_s), \ddot{\mathbf{p}}_t)B_s(\mathbf{p}_s) \tag{2}$$

with A, B images defined in Ω, w_c weighting the source pixel's contribution, and w_b being a bilinear interpolation weight as the re-projections of Eq. 1 produce results at sub-pixel accuracy. Therefore, we "*splat*" each source pixel to contribute to the re-projected target pixel's immediate neighborhood, expressed by $\ddot{\mathbf{p}}_t \in \{{}_x\lfloor\mathbf{p}_t\rfloor_y, {}_x\lfloor\mathbf{p}_t\rceil_y, {}_x\lceil\mathbf{p}_t\rceil_y, {}_x\lceil\mathbf{p}_t\rfloor_y\}$, where $\lceil . \rceil$ and $\lfloor . \rfloor$ denote ceiling and floor operations respectively in the subscripted image domain directions x, y. Effectively, every source pixel will splat its contribution to four target pixels, enforcing local differentiability.

We weight the contribution of each pixel by taking its uncertainty into account which is expressed as the combination of the measurement noise along the ray, as well as the radial distortion error: $w_c(D, \mathbf{p}) = w_d(D, \mathbf{p})w_r(\mathbf{p})$. To increase applicability, both the depth uncertainty and the radial distortion confidence weights are modelled in a generic way. For depth uncertainty, we consider measurements closer to the sensor's origin as more confident than farther ones; $w_d(D, \mathbf{p}) = \exp(\frac{-D(\mathbf{p})}{\sigma_D})$, controlled by σ_D. Similarly, for the radial distortion confidence a generic FoV model is used [41]:

$$\mathcal{R}(\mathbf{p}) = \frac{tan(r(\mathbf{p})tan(\omega))}{tan(\omega)}, \tag{3}$$

where $r(\mathbf{p}) = \sqrt{(x^2 + y^2)}$ is the pixel's radius from the distortion center (*i.e.* the principal point) and ω is half the sensor's FoV. In this way, measurements in high distortion areas are considered as less confident and weighted by $w_r(\mathbf{p}) = \exp(\frac{r(\mathbf{p})}{\mathcal{R}(\mathbf{p})})$.

We splat and accumulate the weighted color contributions from a source image s to a target image t, as well as the weights themselves via the splatting function $\mathcal{S}$:

$$\mathcal{S}_{s\to t}(\hat{\mathbf{I}}_t, \mathbf{I}_s, D_s, \mathbf{p}_s), \qquad \mathcal{S}_{s\to t}(W_t, \mathbf{1}, D_s, \mathbf{p}_s), \tag{4}$$

where W and $\mathbf{1}$ are scalar maps of splatted weights and ones respectively, defined in the image domain Ω. In order to compute the reconstructed image, a weighted average normalization is performed in the target view; $\hat{\mathbf{I}}_t =$

$\hat{\mathbf{I}}_t \oslash (W_t \oplus \epsilon)$, with ϵ being a small constant to ensure numeric stability, while circles denote element wise operators.

Note that through forward splatting, the blended values of the target image enable gradient flow to all contributing measurements. In traditional rendering (discrete rasterization), gradient flow to depth values close to surface would be cut-off, and given the bidirectional nature of noise, this would encumber the learning process. On the contrary, using forward splatting, background depths only minimally contribute to the blended pixels due to the exponential weight factor, receiving minimal gradients, thus implicitly handling occlusions and visibility tests.

In a multi-view setting with S sensors, we can splat the contributions in a many-to-one scheme, in order to fully exploit multi-view predictions. For each view t, a splatted image is rendered by accumulating the color and weight splats from all other views to the zero-initialized $\hat{\mathbf{I}}_t, W_t$:

$$\forall \{s, t | t \neq s\} \in S : \mathcal{S}_{s \rightarrow t}(\mathbf{I}_s, \hat{\mathbf{I}}_t, D_s, \mathbf{p}_s), \mathcal{S}_{s \rightarrow t}(W_s, \mathbf{1}, D_s, \mathbf{p}_s) \tag{5}$$

and then subsequently $\hat{\mathbf{I}}_t$ is normalized. The presented depth-image-based differentiable rendering allows us to exploit photometric supervision in a many-to-many scheme, thus relying only on aligned color information to supervise depth denoising.

3.2. Network Architecture

The proposed data-driven approach is realized as a deep autoencoder depicted in Fig. 3. Its structure is inspired by the U-Net [35] architecture that consists of an encoder, a latent, and a decoder part, respectively. Note that the network is fully convolutional as there is no linear layer.

The encoder follows the typical structure of a CNN and consists of 9 convolutional (CONV) layers each followed by an Exponential Linear Unit (ELU) [8] activation function. The input is downsampled 3 times prior to the latent space using convolution with 3×3 kernels and stride 2, while the number of channels is doubled after every downsampling layer.

The latent part consists of 2 consecutive residual blocks each following the ELU-CONV-ELU-CONV structure adopting the pre-activation technique and the identity mapping introduced in [15] for performance improvement.

The decoder shares similar structure with the encoder, consisting of 9 CONV layers each followed by an ELU nonlinearity. The features are upsampled 3 times prior to the final prediction, using nearest neighbor upsampling followed by a CONV layer. Note that each downsampling layer is connected with the corresponding upsampling one (features with the same dimensions) with a skip connection. Subsequently, the activations of the upsampling layer are concatenated with the ones from the corresponding skip connection. After concatenation, a CONV layer with 1×1 kernel size follows, forcing intra-channel correlations learning.

In order to ensure that denoising is not affected either from invalid values due to data sparsity or depth difference in edge-cases, the recently presented partial convolutions [27] are used in every CONV layer. The required validity (binary) mask M is formed by parsing the input depth map D and setting $M(\mathbf{p}) = 1$ for $D(\mathbf{p}) > 0$ and $M(\mathbf{p}) = 0$ for zero depth. This mask is given as input to the network and is updated after each partial convolution as in [27].

During training, the network infers a denoised depth map for each sensor. Considering input from 4 sensors, as in Fig. 3, all depth maps are iteratively set as target frames. Thus, following the forward splatting technique presented in Section 3.1, target $\hat{\mathbf{I}}$ is synthesized using information from the non-target predicted depth maps. The target $\mathbf{I}$ and $\hat{\mathbf{I}}$ are used to compute the photometric loss, which is discussed in the next section. Note that the gradients are accumulated for all different target depth maps and the weights update of the network is performed once. This way we perform denser back-propagation in each iteration, even though our inputs are sparse, leading to faster and smoother convergence.

3.3. Losses

The proposed network is trained using a geometrically-derived photometric consistency loss function. Additionally, depth and normal priors are exploited as further regularization, which force spatial consistency and surface smoothness. The total loss that is used to compute the network gradients is defined as:

$$\mathcal{L}_{total} = \underbrace{\lambda_1 \mathcal{L}_{ph}}_{\text{data}} + \underbrace{\lambda_2 \mathcal{L}_{depth} + \lambda_3 \mathcal{L}_{surface}}_{\text{priors}}, \tag{6}$$

where $\lambda_1, \lambda_2, \lambda_3 \in (0, 1)$ are hyperparameters that add up to 1. The photometric loss, as well as the regularization functions are discussed in detail below.

Photometric consistency: $\mathcal{L}_{ph}$ forces the network to minimize the pixel-wise error between input $\mathbf{I}$ and $\hat{\mathbf{I}}$. Note that in order to perform correct pixel-wise supervision, we compute the binary mask of $\hat{\mathbf{I}}$, denoted as M_{splat}, where $M_{splat}(\mathbf{p}) = 1$ for $\hat{\mathbf{I}}(\mathbf{p}) > 0$ and $M_{splat}(\mathbf{p}) = 0$ for zero $\hat{\mathbf{I}}(\mathbf{p})$ values. Subsequently, the masked input image $\ddot{\mathbf{I}}$ is used as ground truth and is computed as $\ddot{\mathbf{I}} = M_{splat} \odot \mathbf{I}$, where $\odot$ denotes element-wise multiplication. $\mathcal{L}_{ph}$ is composed of two terms, namely the "color-based" $\mathcal{L}_{col}$ and the "structural" $\mathcal{L}_{str}$ loss, respectively. The color-based loss is defined as:

$$\mathcal{L}_{col} = \sum_{\mathbf{p}} \rho(M(\mathbf{p}) || \ddot{\mathbf{I}}(\mathbf{p}) - \hat{\mathbf{I}}(\mathbf{p}) ||_1), \tag{7}$$

where M is the validity mask (see Section 3.2) and $\rho(x) = \sqrt{x^2 + \gamma^2}$ is the Charbonnier penalty [6, 40] (γ is a near-zero constant) used for robustness against outliers. $\mathcal{L}_{col}$

aims to penalize deviations in the color intensity between $\ddot{\mathbf{I}}$ and $\hat{\mathbf{I}}$. On the other hand, we use structured similarity metric (SSIM) as the structural loss between $\ddot{\mathbf{I}}$ and $\hat{\mathbf{I}}$ which is defined as:

$$\mathcal{L}_{str} = 0.5 \sum_{\mathbf{p}} \phi(M(\mathbf{p})(1 - \text{SSIM}(\ddot{\mathbf{I}}(\mathbf{p}), \hat{\mathbf{I}}(\mathbf{p})))), \tag{8}$$

where M is the same validity mask as in Eq. 7 and $\phi(x)$ is the Tukey's penalty, used as in [5] given its property to reduce the magnitude of the outliers' gradients close to zero. Intuitively, $\mathcal{L}_{str}$ forces prediction invariance to local illumination changes and structural information preservation. Note that the aforementioned penalty functions are used to address the lack of constrains (*i.e.* Lambertian surfaces, no occlusions) that need to be met for photometric consistency supervision, albeit not applicable on real-world multi-view scenarios. Finally, the total photometric loss function is defined as the linear combination of the aforementioned color-based and structural losses, and is given by:

$$\mathcal{L}_{ph} = (1 - \alpha)\mathcal{L}_{col} + \alpha \mathcal{L}_{str}, \tag{9}$$

where $\alpha \in (0, 1)$ is a hyperparameter.

Depth regularization. We choose to further regularize the aforementioned photometric consistency loss by exploiting depth information priors. In particular, considering the residual $r = M \odot (D - \tilde{D})$, where $\tilde{D}$ is the denoised prediction of the network, we use the inverse Huber (BerHu) penalty [25]:

$$\mathcal{L}_{depth} = \begin{cases} |r|, & |r| \leq c \\ \frac{r^2 + c^2}{2c}, & |r| > c \end{cases}, \tag{10}$$

where c is a border value defined as the 20% of the maximum per batch residual $c = 0.2 \max(r)$. The choice of BerHu instead of $L2$ is based on [24], where it was found that it is more appropriate as a depth estimator, as it behaves as $L1$ for residuals lower than the border value.

Surface regularization. Besides depth regularization, a surface regularization prior is used to enforce smoothness in the predicted depth maps. In particular, the surface loss is given by:

$$\mathcal{L}_{surface} = 1 - \sum_{\mathbf{p}} \sum_{\mathbf{p}' \in \Theta_{\mathbf{p}}} |\langle \mathbf{n}(\mathbf{p}), \mathbf{n}(\mathbf{p}') \rangle| \frac{M(\mathbf{p})}{G(\Theta_{\mathbf{p}})}, \tag{11}$$

where $\mathbf{n}(\mathbf{p})$ is the normal vector of the 3D local surface computed by the deprojected points $\mathbf{v}(\mathbf{p})$, $\Theta_{\mathbf{p}}$ is the set of all 2D neighboring pixels around $\mathbf{p}$, $G(\Theta_{\mathbf{p}})$ is an operator that counts the number of valid depth pixels (non-zero) in the neighborhood $\Theta_{\mathbf{p}}$, and at last, $\langle \cdot, \cdot \rangle$ is the inner product between 2 vectors. Note that $\mathbf{n}(\mathbf{p})$ is normalized so that $|\langle \mathbf{n}, \mathbf{n}' \rangle| \in [0, 1]$ and $| \cdot |$ is the absolute value operator.

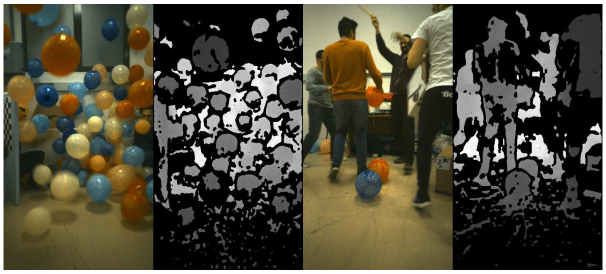

Figure 4. Collected training set samples showing the captured content (balloons and the multi-person activities).

4. Experimental Results

In this section we quantitatively and qualitatively demonstrate the effectiveness of our self-supervised approach against recent state-of-the-art supervised methods, as well as traditional filtering approaches. The recently released Intel RealSense D415, an active stereo RGB-D sensor, is used for data collection and evaluation.

Training RGB-D Dataset. For training our model, a new RGB-D corpus has been collected, using multiple D415 devices containing more than 10K quadruples of RGB-D frames. We employ $V = 4$ vertically orientated sensors in a semi-structured deployment as depicted in Fig. 1, using a custom-made H-structure. The H-structure offers approximate symmetric placement and different vertical and horizontal baselines. For the sake of spatio-temporal alignment between the color and the depth streams of the sensor, the infrared RGB stream was used instead of the extra RGB only camera. This ensures the alignment of the color and depth image domains, and circumvents a technical limitation of the sensors that does not offer precise HW synchronization between the stereo pair and the RGB camera. The sensor is configured to its "high accuracy" profile, offering only high confidence depth estimates, but at the same time, producing highly sparse depth data, *i.e.* $\approx 60\%$ of zero-depth values in typical human capturing scenarios. Data were captured using [39], spatial alignment between the 4 sensors was achieved by the multi-sensor calibration of [34], while, precise temporal alignment was achieved through the inter-sensor HW synchronization offered by the D415 sensors.

As our approach relies on view synthesis for supervision, we can easily collect raw data for training. This is advantageous compared to using 3D reconstruction methods to generate near ground truth datasets [46, 18]. With respect to the dataset content, aiming to create a dataset of sufficient depth variability, we captured human actions simultaneously performed by multiple people as well as a special set of moving textured balloons of different colors. In detail, multiple subjects (1-7) performed free (*i.e.* not predefined) actions, while a variety of balloons were blown in the air using a blowing machine, creating depth maps of high variability. Note that the random movement patterns fully covered the

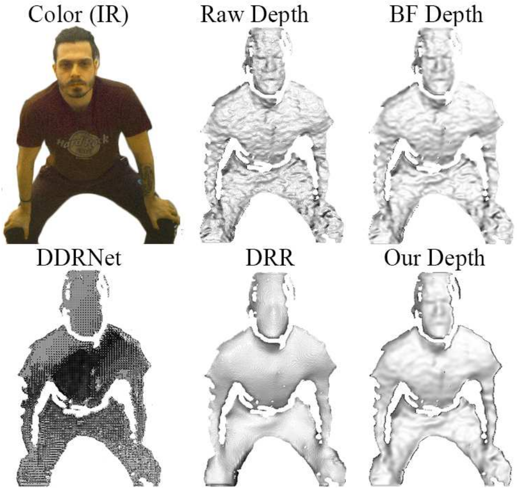

Figure 5. Qualitative results using D415 data.

sensors' FoV and prevented spatial bias in the training set. Indicative samples are depicted in Fig. 4.

Implementation Details. The training methodology along with the network hyper-parameters are presented in Section 1.1 of the supplementary material.

Evaluation Methodology. The proposed model is evaluated against traditional filtering methods, such as Bilateral Filter (BF [43]), Joint Bilateral Filter (JBF [21]), Rolling Guidance (RGF [48]), as well as data-driven approaches such as DRR [18] and DDRNet [46]. Note that for the DDRNet case, the refinement part of the network is omitted in order to have a fair comparison in denoising. Due to the lack of ground truth, depth maps from Kinect v2 (K2) [36] are used as "close to ground truth" data for the quantitative evaluation. That is, a 70-pair RGB-D set of instant samples with varying content are captured using a rigid-structure that combines K2 and D415, and is used as test set for evaluation purposes. In particular, to achieve the closest possible positioning between the modalities, the two sensors are placed in a way that the overlap of their FoV is high, while the structure is calibrated using the Matlab Stereo Camera Calibrator App [49]. The evaluation consists of 3 experiments varying from direct depth map comparison to application-specific measurements. In detail, for the first experiment the depth maps captured by the D415 sensor are denoised by the proposed network and the state-of-the-art methods and the result is evaluated using the K2 ground truth data. Subsequently, using the aforementioned rigid-structure, we capture 15 scanning sequences with both sensors (D415, K2) simultaneously, which are then utilized as inputs to KinectFusion. For our last experiment, we utilize a multi-view setup to capture 5 full-body samples. Note that besides quantitative evaluation, for each experiment qualitative results are also presented.

Results. In the first experiment we use projective data association to compare the performance of denoising methods on D415 data against the close to ground truth K2 depth maps. The results are presented in Table 1 (columns 2-7) and showcase the effectiveness of the proposed method against supervised methods and traditional filters. Despite the low mean absolute error differences between the quantified methods, the RMSE results prove the effectiveness of our approach to denoise depth maps by achieving the lowest error deviation. Regarding surface errors, our method ranks third following DRR [18] and RGF [48] with slight differences. However, DRR filtering results in depth map oversmoothing and spatial offsets (bendings), degenerating high frequent details, thus causing large distance errors. On the other hand, DDRNet [46] under-performs in D415 depth data denoising. This can be attributed either to the context specific learning of the network on high density depth maps of humans without background, which showcases the disadvantage of using specific 3D reconstruction methods to generate near ground truth data for supervision, and the inability to generalize well. Another reason may be that the noise level of D415 is higher than the sensors [46] was trained with. In addition, the fact that D415 produces sparse results hampers the applicability of CNN-based methods that did not account for that, due the nature of convolutional regression. Finally, classical and guided filters present comparatively larger errors than the proposed method. Qualitative results[1] of the depth map denoising are illustrated in Fig. 5. It is apparent that local filtering cannot sufficiently denoise due to its local nature, while the learning-based alternatives either oversmooth (DRR) or fail to generalize to other sensors (DDRNet). Instead, our approach smooths out the noise while preserving structural details.

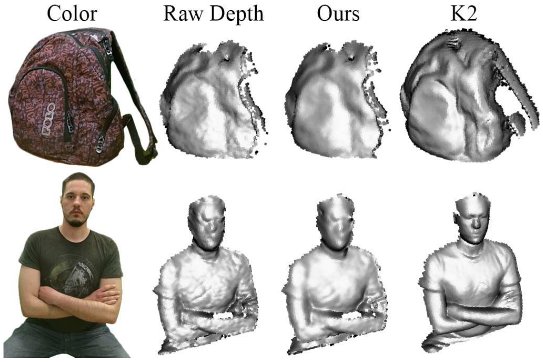

Figure 6. Qualitative results using KinectFusion.

The second experiment demonstrates results in an application setting using KinectFusion to 3D reconstruct static scenes. The rationale behind this experiment is the comparison of the scanning results using the denoised depth maps of D415 and comparing the result with that of a K2 scan. Quantitative results are presented in Table 1 (last column), while qualitative results are illustrated in Fig. 6.

[1] Additional qualitative results related to our experiments are included in the supplementary material document.

Table 1. Quantitative evaluation of the denoising algorithms: Depth-map and surface errors as well as errors in a 3D reconstruction task.

	Euclidean Distance		**Normal Angle Difference**				**Kinect Fusion**
	MAE (mm)	**RMSE (mm)**	**Mean (°) ↓**	**10.0 (%) ↑**	**20.0 (%) ↑**	**30.0 (%) ↑**	**RMSE (mm)**
DDRNet [46]	114.57	239.06	52.85	1.78	7.30	16.59	50.79
DRR [18]	75.40	201.49	**30.23**	**10.95**	**34.69**	**57.76**	37.31
JBF [21]	27.10	84.84	38.57	6.14	21.08	39.61	27.68
RGF [48]	26.60	81.35	31.84	9.46	31.00	53.58	32.58
BF [43]	26.11	73.25	35.04	7.42	25.38	46.11	29.85
Ours	**25.11**	**58.95**	32.09	9.61	31.34	53.65	**24.74**

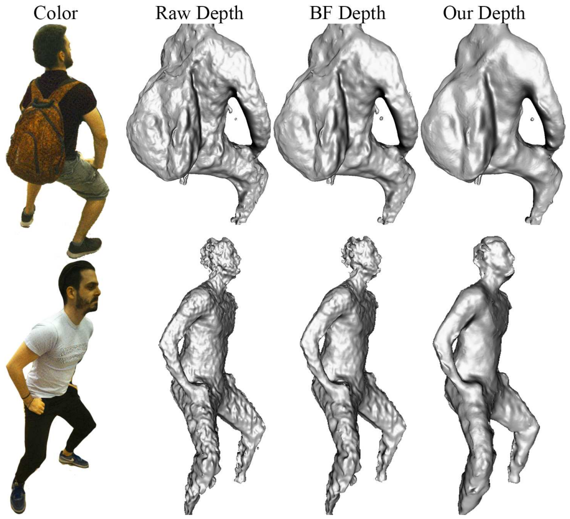

Figure 7. Qualitative results using Poisson reconstruction.

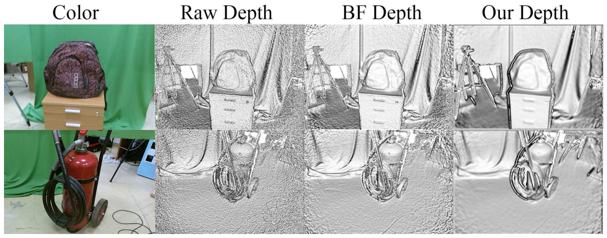

Figure 8. Qualitative comparison using K2 data.

In this experiment we opt to use an aggregated metric that handles surface and geometry information jointly, point-to-plane. Instead of relying on the nearest neighbor for distance computation, we calculate the Least Square Planes for each point in the close-to-ground truth point cloud using all vertices in a 5mm radius (a 2mm voxel grid size was used when 3D scanning). The distance of each calculated plane of the ground-truth point cloud against the closest point from the denoised point clouds contribute a term to the final RMSE.

While KinectFusion reconstructs surfaces by aggregating and fusing depth measurements it also implicitly denoises the result through the TSDF fusion process. In order to accentuate surface errors we conduct another experiment, this time using Poisson reconstruction [19], which requires better surface estimates in order to appropriately perform 3D reconstruction. This allows us to qualitatively assess the denoised output smoothness, while also showcasing the preservation of structure. We spatially align 4 D415 sensors in a 360° placement and capture depth frame quadruples of static humans. We use deprojected raw and denoised depth maps to point clouds and calculate per point normals using the 10 closest neighbors. These oriented point clouds are reconstructed using [19] with the results illustrated in Fig. 7. It is apparent that BF, one of the performing filters of the first experiment, performs smoothing without removing all noise as it operates on local level. On the contrary, the 3D reconstructed model using the denoised depth maps of our model achieves higher quality results, mainly attributed to its ability to capture global context more effectively.

Finally, while other denoising CNNs trained using other sensors fail to produce good results on D415, we also present qualitative results on K2 data[2], albeit trained using D415 noisy depths. Fig. 8 shows that our model gracefully handles noise from other sensors, contrary to fully supervised methods that are trained on datasets of a specific context (sensor, content).

5. Conclusion

In this paper, an end-to-end model was presented for the depth denoising task. To tackle the lack of ground truth depth data, the model was trained using multiple RGB-D views of the same scene using photometric, geometrical, and surface constraints in a self-supervised manner. The model outperformed both traditional filtering and data-driven methods, through direct depth map denoising evaluation and two well-established 3D reconstruction applications. Further, it was experimentally shown that our model, unlike other data-driven methods, maintains its performance when denoising depth maps captured from other sensors. The limitations of the method lie in the need of color information for supervision, and sensors' hardware synchronization.

Acknowledgements. We thank Antonis Karakottas for his help with the hardware setup, and the subjects recorded in our dataset for their participation. We also acknowledge financial support by the H2020 EC project VRTogether under contract 762111, as well as NVidia for GPU donation.

[2] We collect our own data as the K2 dataset of [46] is not yet publicly available.

References

[1] Gianluca Agresti and Pietro Zanuttigh. Deep learning for multi-path error removal in ToF sensors. In *ECCVW*, pages 410–426, 2018.

[2] Dimitrios S. Alexiadis, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras. Fast deformable model-based human performance capture and FVV using consumer-grade RGB-D sensors. *Pattern Recognition*, 79:260–278, 2018.

[3] Filippo Basso, Emanuele Menegatti, and Alberto Pretto. Robust intrinsic and extrinsic calibration of RGB-D cameras. *IEEE Transactions on Robotics*, (99):01–18, 2018.

[4] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *ICML*, 2019.

[5] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *ICCV*, pages 2830–2838, 2015.

[6] Qifeng Chen and Vladlen Koltun. Fast MRF optimization with application to depth reconstruction. In *CVPR*, pages 3914–3921, 2014.

[7] Yiping Chen, Jingkang Wang, Jonathan Li, Cewu Lu, Zhipeng Luo, Han Xue, and Cheng Wang. LiDAR-Video driving dataset: Learning driving policies effectively. pages 5870–5878, 2018.

[8] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by Exponential Linear Units (ELUs). In *ICLR*, 2016.

[9] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion. *ACM Transactions on Graphics*, 36(4):1, 2017.

[10] Shuhang Gu, Wangmeng Zuo, Shi Guo, Yunjin Chen, Chongyu Chen, and Lei Zhang. Learning dynamic guidance for depth image enhancement. In *CVPR*, pages 712–721, 2017.

[11] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera. *ACM Transactions on Graphics*, 36(3):32, 2017.

[12] Qi Guo, Iuri Frosio, Orazio Gallo, Todd Zickler, and Jan Kautz. Tackling 3D ToF artifacts through learning and the FLAT dataset. In *ECCV*, pages 368–383, 2018.

[13] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust image filtering using joint static and dynamic guidance. In *CVPR*, pages 4823–4831, 2015.

[14] Yudeog Han, Joon-Young Lee, and In So Kweon. High quality shape from a single RGB-D image under uncalibrated natural illumination. In *ICCV*, pages 1617–1624, 2013.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016.

[16] Daniel Herrera, Juho Kannala, and Janne Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064, 2012.

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025. 2015.

[18] Junho Jeon and Seungyong Lee. Reconstruction-based pairwise depth dataset for depth image enhancement using CNN. In *ECCV*, pages 438–454, 2018.

[19] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):29, 2013.

[20] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel (R) realsense (TM) stereoscopic depth cameras. In *CVPRW*, pages 1267–1276, 2017.

[21] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics*, 26(3), 2007.

[22] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2Void - Learning denoising from single noisy images. In *CVPR*, pages 2129–2137, 2019.

[23] HyeokHyen Kwon, Yu-Wing Tai, and Stephen Lin. Data-driven depth map refinement via multi-scale sparse representation. In *CVPR*, pages 159–167, 2015.

[24] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248, 2016.

[25] Sophie Lambert-Lacroix and Laurent Zwald. The adaptive BerHu penalty in robust regression. *Journal of Nonparametric Statistics*, 28(3):487–514, 2016.

[26] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Joint image filtering with deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1, 2019.

[27] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[28] Si Lu, Xiaofeng Ren, and Feng Liu. Depth enhancement via low-rank matrix completion. In *CVPR*, pages 3390–3397, 2014.

[29] Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min H. Kim, Xin Tong, and Diego Gutierrez. DeepToF: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Transactions on Graphics*, 36(6):1–12, 2017.

[30] Kazuki Matsumoto, Francois De Sorbier, and Hideo Saito. Plane fitting and depth variance based upsampling for noisy depth map from 3D-ToF cameras in real-time. In *ICPRAM*. Science and and Technology Publications, 2015.

[31] Valeriya Naumova and Karin Schnass. Dictionary learning from incomplete data for efficient image restoration. In *EUSIPCO*, pages 1425–1429, 2017.

[32] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011.

[33] Roy Or - El, Guy Rosman, Aaron Wetzler, Ron Kimmel, and Alfred M. Bruckstein. RGBD-fusion: Real-time high precision depth recovery. In *CVPR*, pages 5407–5416, 2015.

[34] Alexandros Papachristou, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras. Markerless structure-based multi-sensor calibration for free viewpoint video capture. In *Proc. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 88–97, 2018.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[36] John Sell and Patrick O'Connor. The xbox one system on a chip and Kinect sensor. *IEEE Micro*, 34(2):44–53, 2014.

[37] Ju Shen and Sen-ching S. Cheung. Layer depth denoising and completion for structured-light RGB-D cameras. In *CVPR*, pages 1187–1194, 2013.

[38] Xiaoyong Shen, Chao Zhou, Li Xu, and Jiaya Jia. Mutual-structure for joint filtering. In *ICCV*, 2015.

[39] Vladimiros Sterzentsenko, Antonis Karakottas, Alexandros Papachristou, Nikolaos Zioulis, Alexandros Doumanoglou, Dimitrios Zarpalas, and Petros Daras. A low-cost, flexible and portable volumetric capturing system. In *SITIS*, pages 200–207, 2018.

[40] Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, pages 2432–2439, 2010.

[41] Zhongwei Tang, Rafael Grompone von Gioi, Pascal Monasse, and Jean-Michel Morel. A precision analysis of camera distortion models. *IEEE Transactions on Image Processing*, 26(6):2694–2704, 2017.

[42] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. In *CVPR*, pages 6565–6574, 2017.

[43] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[44] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3D scene inference via view synthesis. In *ECCV*, pages 302–317, 2018.

[45] Chenglei Wu, Michael Zollhöfer, Matthias Nießner, Marc Stamminger, Shahram Izadi, and Christian Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics*, 33(6):1–10, 2014.

[46] Shi Yan, Chenglei Wu, Lizhen Wang, Feng Xu, Liang An, Kaiwen Guo, and Yebin Liu. DDRNet: Depth map denoising and refinement for consumer depth cameras using cascaded CNNs. In *ECCV*, pages 155–171, 2018.

[47] Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, and Stephen Lin. Shading-based shape refinement of RGB-D images. In *CVPR*, pages 1415–1422, 2013.

[48] Qi Zhang, Xiaoyong Shen, Li Xu, and Jiaya Jia. Rolling guidance filter. In *ECCV*, pages 815–830, 2014.

[49] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.