Unsupervised 3D Shape Coverage Estimation with Applications to Colonoscopy

Yochai Blau, Daniel Freedman, Valentin Dashinsky, Roman Goldenberg, Ehud Rivlin
Google Research
{yochaib, danielfreedman, valkad, rgoldenberg, ehud}@google.com

Abstract

Reconstructing shapes from partial and noisy 3D data is a well-studied problem, which in recent years has been dominated by data-driven techniques. Yet in a low data regime, these techniques struggle to provide fine and accurate reconstructions. Here we focus on the relaxed problem of estimating shape coverage, i.e. asking "how much of the shape was seen?" rather than "what was the original shape?" We propose a method for unsupervised shape coverage estimation, and validate that this task can be performed accurately in a low data regime. Shape coverage estimation can provide valuable insights which pave the way for innovative applications, as we demonstrate for the case of deficient coverage detection in colonoscopy screenings.

1. Introduction

The last decade has seen a surge of research on 3D reconstruction and shape completion techniques, which are crucial in many computer vision and robotic systems. In this ill-posed inverse problem, the goal is to restore an original 3D shape or scene when only observing a partial and noisy sample of it. However, fully reconstructing the original shape is unnecessarily difficult for certain applications. In such settings, instead of asking "what was the original shape?", it may suffice to look at a simplified question: "how much of the original shape have we seen?". In this paper, we focus on this *shape coverage estimation* task (see Fig. 1). We propose a method for performing shape coverage estimation, validate its effectiveness, and demonstrate its applicability in a low data regime.

Obviously, if we perform 3D reconstruction, then we can quite easily estimate the area covered by the partial views at hand. Yet, 3D reconstruction from partial data is a challenging ill-posed task. In recent years, advances in 3D reconstruction have been dominated by data-driven techniques, which rely on an abundance of data. In the supervised scenario, many works have learned an end-to-end mapping from partial data to the reconstructed shape by leveraging a dataset of such input-output pairs [9, 47, 21, 53]. Others have studied the weakly-supervised/unsupervised scenarios, where such paired data is not available, but a large set of shape samples are present [59, 55, 67].

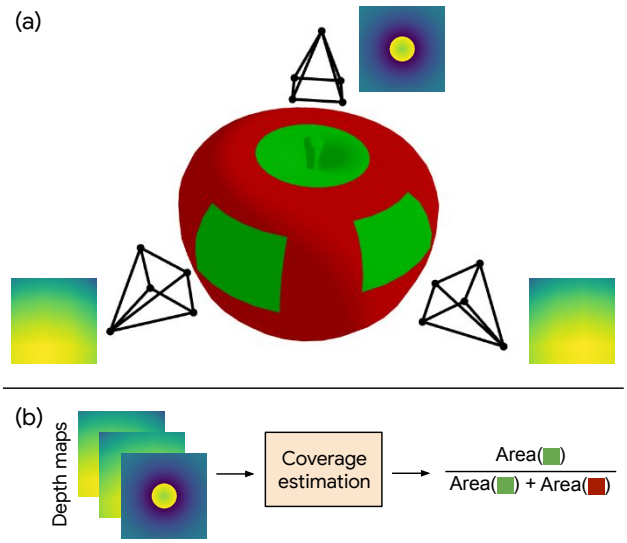

Figure 1: **Coverage estimation.** (a) Consider a depth sensor which acquires partial views of a 3D shape. The seen area is colored in green. (b) Given *only* the acquired depth maps (and camera poses) we estimate the shape coverage, *i.e.* the ratio of the seen shape surface to the total shape surface. This is performed in a low data regime, where (i) no depth maps/shape coverage pairs are available for supervised learning, and (ii) a scarce amount of full shape samples is available for devising a shape model.

By contrast, we are interested in the scenario which we refer to as the *Low Data Regime*. This regime implies the following:

1. Absence of Labelled Data: Our training data contains noisy depth views which represent partially covered surfaces. It does *not* contain the ground-truth coverage values; nor does it contain the complete surfaces from which the partial depth views are derived. This is the standard sense in which the data is unsupervised.

2. Paucity or Absence of Unlabelled Data: We have access to a surface model, *i.e.* a model which captures the family

of surfaces of interest; however, the key point is that this model is quite coarse. In particular, it is *not* learned from massive amounts of unlabelled *complete* surfaces. Rather, this model is either learned from a very small amount of unlabelled *complete* surfaces, or it is not learned at all - it is simply a crude manually specified model.

Clearly, 3D reconstruction models derived in the low data regime will likely be inferior to the data-driven models demonstrated in recent years. Such models are expected to provide coarse approximations of the original shape with relatively low accuracy. Nevertheless, insights on the relation between the partial inputs and the original shape can still be gained. Our goal in this paper is to estimate the portion of a shape surface that is covered by a set of depth views (see Fig. 1). We denote this task *shape coverage estimation*. Concretely, given a set of depth maps taken by a depth sensor with varying sensor pose, we estimate the ratio of *viewed* surface area to the entire surface area.

We formulate the notion of coverage mathematically, from which an algorithm may be derived. At the heart of this algorithm is a technique for finding a surface from a coarse shape model which best fits the observed partial views. This coarse shape model may be manually specified or constructed from a limited number of shape samples. The best-fitting surface is predicted by a DNN-based model, which is trained in an unsupervised fashion. Analyzing which parts of the surface are covered by the partial views then yields the surface coverage.

Shape coverage estimation can lead to innovative applications. Here, we show the potential for colorectal cancer (CRC) prevention by estimating the colon surface coverage during colonoscopy screenings. Colonoscopies are the gold-standard technique for removing precancerous polyps from the colon to prevent CRC [36]. Yet, it is estimated that over 20% of polyps are missed during the procedure [37], in part due to insufficient coverage of the surface. Ensuring full colon coverage during colonoscopy screenings can prevent CRC [11]. An *online* system alerting on low coverage could augment physicians' performance and reduce the missed polyp rate. Yet, it is practically difficult to obtain a sufficient dataset of 3D colon surface segments and ground truth coverage annotations for developing such a system in a conventional supervised-learning manner. Fortunately, as we show, accurate surface coverage estimation can be obtained in this low data regime scenario.

To summarize, our main contributions are:
1. Mathematical formulation of shape coverage estimation.
2. A method for estimating coverage in a low data regime.
3. Experimental validation of our method on datasets where ground-truth coverage can be deduced.
4. A demonstration that our method can efficiently estimate deficient coverage in colonoscopy screenings.

2. Related Work

Shape Completion Early works on 3D shape completion commonly relied on locally optimizing a surface to fit the points [29, 30], or leveraging symmetry or self-similarity in objects or scenes [49, 62, 73, 33]. Other works resorted to retrieval and alignment techniques to perform shape completion [48, 38, 19, 54]. Data-driven approaches also include learning a latent space of shapes and optimizing over this space to perform shape completion [70, 10, 12, 22]. With the rise in popularity of deep learning, many works have studied shape completion by end-to-end training of a DNN in the supervised scenario [9, 21, 67, 47, 14, 53, 6, 69, 16, 40]. Self-supervised and weakly-supervised approaches to shape completion have also been proposed [59, 8, 55]. These learning-based techniques rely on the abundance of training data, either as input-output pairs in the fully supervised setting, or as a set of valid shapes for model learning. In contrast, our setting assumes a severe lack of training data where end-to-end DNN training or complex model learning is not possible.

Shape Models Shape models are widely used in many applications such as 3D reconstruction, pose estimation, object detection and more [10, 32, 46, 2, 57, 26, 70, 12]. While early works commonly relied on hand-crafted models, recently deep generative models have been widely used to learn shape models from data [66, 67, 55, 17, 45, 56]. In contrast to these complex non-linear models, linear models learned from scans have been shown to accurately capture the variations of human body and face shapes [41, 1, 3].

Model-based 3D Pose and Shape Estimation Optimization based methods are a leading paradigm for performing 3D pose and shape estimation [4, 18, 35]. These methods usually produce accurate estimation, yet are relatively slow and are susceptible to local minima. Recently, deep learning methods have attempted to directly regress to model parameters in a one-shot fashion [25, 46, 50]. These are fast and do not rely on good initialization at test-time, yet may be less accurate and rely on large datasets for training. Other works have attempted to combine the best of both worlds [32, 64]. While most works infer pose and shape from images, closer to our setting are works which infer pose and shape from depth data [61, 51].

3D Reconstruction from a Single Image Many methods for single-image 3D reconstruction based on deep learning have been proposed. Fully supervised approaches include [13, 7, 43, 17, 23]. Other work focuses on self-supervised or weakly-supervised approaches which commonly enforce some form of consistency [60, 52, 63, 68, 65, 20, 28, 39]. While these works share similarities with our task, we directly process 3D data rather than images.

"Next-Best-View" This approach *e.g.* [34, 27, 42, 72] formulates the objective as maximizing the coverage of a 3D

surface. There are however key distinctions between these works and ours: (a) These works focus on maximally increasing the coverage with the *next* camera pose, but not on accurately estimating the *current* overall coverage (without additional scans). (b) Data-driven works [72, 42] assume full supervision where ground-truth shapes and coverage values are known for training and evaluation; our main contribution is a method for unsupervised coverage estimation.

3. Methods

3.1. Informal Problem Formulation

We begin with an informal account of the problem. We are given a partial view of a surface. In practice, the partial view will consist of a collection of depth images taken from different angles, and we will use this formulation explicitly in describing the network architecture in Section 3.4. For the purposes of much of the exposition, however, it is sufficient to treat the partial view as simply another surface which is a subset of the main surface, or a noisy approximation to such a subset. Given such a partial view of a surface, our goal is to estimate what fraction of the surface has been seen. This fraction is the so-called coverage.

Our main assumption is that we work in the Low Data Regime. This means two things: (1) absence of labelled data; (2) paucity or absence of unlabelled data (see Section 1). Our scheme for the computation of coverage is illustrated in Figure 2. We now describe the background to derive this scheme, beginning with a definition of coverage.

3.2. Defining Coverage

We denote a single surface as $X : \mathcal{U} \to \mathbb{R}^3$, where $u \in \mathcal{U} \subset \mathbb{R}^2$ is a parameterization of the surface, and $X(u)$ is continuous and sufficiently smooth (*e.g.* $\mathcal{C}^2$). For simplicity, we assume that the atlas consists of a single chart $\mathcal{U}$. The set of points corresponding to the surface will be written as $\mathcal{X} \equiv X(\mathcal{U}) = \{X(u) : u \in \mathcal{U}\}$. A family of surfaces is then denoted as $X : \mathcal{U} \times \mathcal{Q} \to \mathbb{R}^3$ where the parameters $q \in \mathcal{Q} \subset \mathbb{R}^d$ index which surface in the family we are referring to, and the family is d-dimensional. Again $X(u, q)$ is continuous and sufficiently smooth. The set of points corresponding to a particular surface in the family will be written as $\mathcal{X}_q \equiv X(\mathcal{U}, q) = \{X(u, q) : u \in \mathcal{U}\}$.

We now define coverage; we do so through an increasingly general set of assumptions. To begin with, suppose that we are given a single surface $\mathcal{X}$, and a partial view of that surface $\mathcal{S}$ which is an exact subset of $\mathcal{X}$. In this case, the coverage is defined straightforwardly as

$$C = \frac{A[\mathcal{S}]}{A[\mathcal{X}]} \tag{1}$$

where $A[\cdot]$ denotes the area of the relevant surface.

In the case that we are given an entire family of surfaces, then the coverage generalizes as

$$C = \frac{A[\mathcal{S}]}{A[\mathcal{X}_{q^*}]} \text{ where } q^* \text{ is such that } D(\mathcal{S}, \mathcal{X}_{q^*}) = 0 \tag{2}$$

where D is a distance between sets. That is, q^* is the member of the family of surfaces for which the partial view $\mathcal{S}$ is a subset. If there is more than one q^* with $D(\mathcal{S}, \mathcal{X}_{q^*}) = 0$, we take the one with highest coverage.

For illustrative purposes, we may think of D as the ordinary (asymmetric) Hausdorff distance, $D(\mathcal{S}, \mathcal{X}_{q^*}) = \sup_{s \in \mathcal{S}} \inf_{x \in \mathcal{X}_{q^*}} \|s - x\|$, though we will use a somewhat different distance measure in practice which we describe in Section 3.3. For our purposes, the crucial aspect of any distance measure D that we use is that it is asymmetric: all points in the partial view $\mathcal{S}$ should match a point in the full surface $\mathcal{X}_{q^*}$, but not the other way around.

What if the surface $\mathcal{S}$ is only approximately a subset of a member of the family of surfaces $\mathcal{X}_q$? This situation naturally arises in practice, where there is sensor noise, approximate algorithms for depth estimation, and so forth which generate the partial view $\mathcal{S}$. In this case, let us define the projection operator to be

$$\mathfrak{P}(s, \mathcal{X}) = \arg\min_{x \in \mathcal{X}} \|s - x\| \tag{3}$$

That is, $\mathfrak{P}(s, \mathcal{X})$ is the closest point on the surface $\mathcal{X}$ to the given point s. The projection operator is naturally extended to an entire surface as $\mathfrak{P}(\mathcal{S}, \mathcal{X}) = \{\mathfrak{P}(s, \mathcal{X}) : s \in S\}$. We extend this to a *distance-restricted* projection as follows:

$$\mathfrak{P}_\epsilon(\mathcal{S}, \mathcal{X}) = \{\mathfrak{P}(s, \mathcal{X}) : s \in S \text{ and } \|s - \mathfrak{P}(s, \mathcal{X})\| \leq \epsilon\} \tag{4}$$

which is the set of all points in $\mathcal{S}$ projected onto $\mathcal{X}$ which are within a distance of ϵ from $\mathcal{X}$. With these ideas in hand, we may naturally define coverage as

$$C = \frac{A[\mathfrak{P}_\epsilon(\mathcal{S}, \mathcal{X}_{q^*})]}{A[\mathcal{X}_{q^*}]} \text{ where } q^* = \arg\min_q D(\mathcal{S}, \mathcal{X}_q) \tag{5}$$

ϵ is chosen to take account of "reasonable noise": if points from $\mathcal{S}$ are too far away from the surface $\mathcal{X}$, then the points they project to on $\mathcal{X}$ ought not to be considered covered.

3.3. Algorithmic Approach

General Approach Our goal is to learn to compute coverage in an unsupervised fashion for a partial view $\mathcal{S}$. The key insight is that the definition of coverage in Equation (5) actually leads to an algorithm. In particular, rather than training a network to compute coverage directly, we instead train the network to compute the optimal q^*, *i.e.* the surface from the given family of surfaces which most closely matches $\mathcal{S}$. The coverage may then be computed directly from q^* using

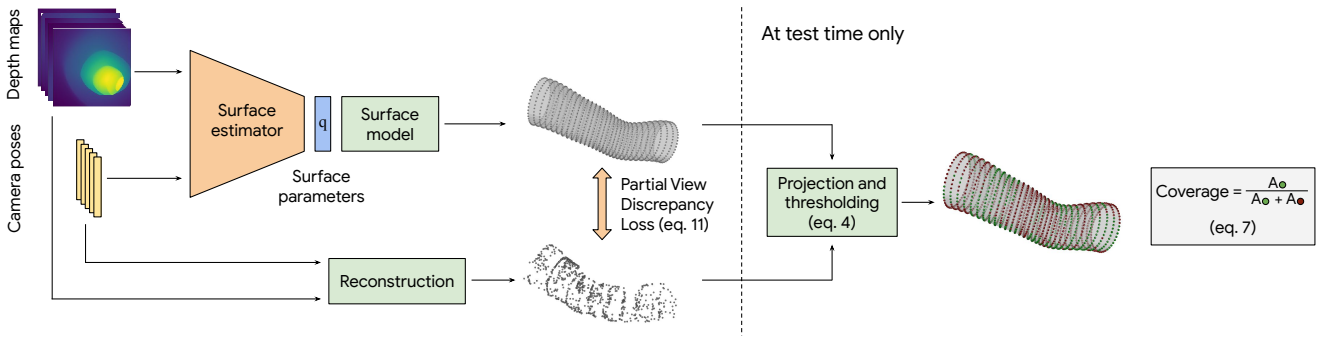

Figure 2: **Unsupervised coverage estimation.** We train a DNN to predict the surface of a 3D shape present in a partial sequence of depth maps and corresponding camera poses. Training is performed in an unsupervised manner, where discrepancies between the predicted surface and the 3D reconstruction of the depth maps are minimized (eq. (11)). Note that the surface model is fixed and serves as a prior of valid surfaces. *At test time:* surface points in proximity of observed 3D points in the reconstruction are considered "seen". The **coverage rate** is the ratio of the seen surface area to the total surface area.

the formula for C in Equation (5). We will see empirically that the resulting unsupervised algorithm will be well suited to the Low Data Regime. The scheme is shown in Figure 2.

To restate the above in a more formal mathematical fashion: let our network be of the form $F(\mathcal{S};\theta) \in \mathcal{Q}$, where the network's parameters are given by θ. Then our goal is to find θ such that

$$F(\mathcal{S};\theta) = \arg\min_{q} D(\mathcal{S}, \mathcal{X}_q) \tag{6}$$

Given the output of the network, it is straightforward to compute the coverage precisely as in Equation (5), *i.e.*

$$C(\mathcal{S};\theta) = \frac{A[\mathfrak{P}_\epsilon(\mathcal{S}, \mathcal{X}_{F(\mathcal{S};\theta)})]}{A[\mathcal{X}_{F(\mathcal{S};\theta)}]} \tag{7}$$

To train our network F, we minimize the following loss:

$$\min_{\theta} L(\theta) = \mathbb{E}_{\mathcal{S}}\left[D(\mathcal{S}, \mathcal{X}_{F(\mathcal{S};\theta)})\right] \tag{8}$$

where the expectation is over different noisy partial views $\mathcal{S}$ of elements of the family of surfaces given by $X(u,q)$.

Enabling Implementation The optimization problem in (8) leads to an algorithm for learning; however, as it stands the problem is not implementable in a straightforward fashion, for reasons relating to both the need for discretization and the differentiability of the loss. We therefore make the following four changes in order to alleviate these issues:

(1) Discretization: It is straightforward to transform the coverage definition in (5) and hence the loss in Equation (8), from working on continuous surfaces to discrete samples drawn from those surfaces. The form of the loss $L(\theta)$ remains the same, but the distance D is now over discrete sets. The other modification is to change the meaning of $A[\cdot]$ in the coverage computation (5) from area to a discrete measure (e.g. cardinality in the case of uniform sampling).

(2) Choice of Set-Distance Function: As described previously, the crucial aspect of the distance measure D is that it is asymmetric: in Equation (8), all points in the partial view $\mathcal{S}$ should match a point in the full surface $\mathcal{X}_{F(\mathcal{S};\theta)}$, but not the other way around. We therefore choose the following natural set distance:

$$D(\mathcal{S}, \mathcal{X}_{F(\mathcal{S};\theta)}) = \frac{1}{|\mathcal{S}|}\sum_{s\in S} \min_{x\in\mathcal{X}_{F(\mathcal{S};\theta)}} \|s-x\| \tag{9}$$

Unlike the Hausdorff distance, this D depends on *all* points in $\mathcal{S}$, which is desirable for robustness.

(3) Continuous Relaxation: The distance introduced in (9) is not smooth, due to the presence of the min operators. We therefore replace the min operation by a soft-min:

$$D(\mathcal{S}, \mathcal{X}_{F(\mathcal{S};\theta)}) = \frac{1}{|\mathcal{S}|}\sum_{s\in S}\sum_{x\in\mathcal{X}_{F(\mathcal{S};\theta)}} w\left(s,x;\mathcal{X}_{F(\mathcal{S};\theta)}\right)\|s-x\| \tag{10}$$

where $w(s,x;\mathcal{X}) = \frac{e^{-\|s-x\|/\sigma}}{\sum_{x'\in\mathcal{X}} e^{-\|s-x'\|/\sigma}}$. The hyperparameter σ controls how close the soft-min approximates the min: as σ gets smaller, the soft-min approaches the min.

(4) Depth Maps: In practice, we have access to the depth map d_i and corresponding pose p_i for each frame i in an n-frame video sequence. The partial view is then given by $\mathcal{S} = \cup_{i=1}^{n}\cup_{s\in d_i} T_{p_i}(s)$, where T_p is the transformation corresponding to pose p, and d_i is construed as a set of points.

Summary of Training Procedure: To summarize, our training procedure consists of minimizing the following loss with respect to the network parameters θ:

$$L(\theta) = \mathbb{E}_{\mathcal{S}}\left[\frac{1}{|\mathcal{S}|}\sum_{s\in S}\sum_{x\in\mathcal{X}_{F(\mathcal{S};\theta)}} w\left(s,x;\mathcal{X}_{F(\mathcal{S};\theta)}\right)\|s-x\|\right] \tag{11}$$

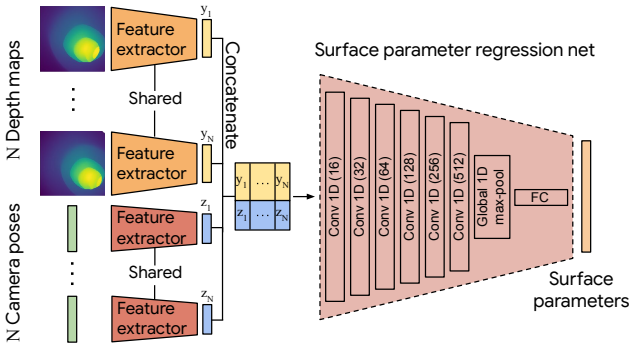

Figure 3: **Surface estimator architecture.** Features are extracted separately from each input depth map and camera pose. The depth feature extractor is a MobileNet [24], and the pose feature extractor is an 8-layer fully-connected net. The features are then concatenated, and fed through an 8-layer convolutional net which outputs the surface parameters. The number of filters are specified in parentheses. See the supplementary for full details.

where w is defined as in Equation (10). We refer to this loss as the *Partial View Discrepancy Loss*. Note that the mean in (11) is taken only over (a batch of) partial views $\mathcal{S}$, without any ground-truth shapes or coverage values appearing in the loss, such that training is truly unsupervised as discussed in Sections 1 and 3.1.

3.4. Surface Estimator Architecture

We now turn to a description of the network architecture, see Figure 3. The input to the network is a chronological sequence of N depth map and camera pose pairs $\{d_i, p_i\}_{i=1}^{N}$. First, features are extracted from each depth map and camera pose vector[1] separately. To support online inference capabilities, the lightweight low-latency MobileNet [24] is used as the depth map feature extractor. The top (classification) layer is discarded, resulting in a feature vector $y_i \in \mathbb{R}^{1024}$ for each depth map. This depth feature extractor is shared across all depth maps. The pose feature extractor is an 8-layer fully-connected net, with all layer widths set to 256 and followed by ReLU activations (except for the final layer), based on the architecture for processing pose vectors in [44] . The output is a feature vector $z_i \in \mathbb{R}^{256}$ for each pose. This pose feature extractor is shared across all the pose vectors.

Next, each depth feature and corresponding pose feature are concatenated, yielding feature vectors $m_i = (y_i, z_i) \in \mathbb{R}^{1280}$. Then, the N features $\{m_i\}$ are stacked, resulting in a feature map $\mathbf{M} \in \mathbb{R}^{1280 \times N}$. The feature map is then forwarded through the surface parameter regression net, which is a 1D convolutional network.[2] Specifically, the net consists of six 1D convolution layers, followed by global 1D max-pooling, and finally a fully-connected layer. The input $\mathbf{M}$ is considered a sequence of N inputs each with 1280 channels, thus the 1D convolutions and max-pooling are performed across the sequence (second feature map dimension). The number of filters in each layer is specified in Fig. 3. The convolution layers have a stride of 2, and are followed by ReLU activations (except for the last layer).

There are a total of 4.3 million parameters in the entire model, and it is trained in an end-to-end manner. Full architecture details can be found in the supplementary.

3.5. Implementation Details

In specifying the parameters $q \in \mathcal{Q}$ which define the family of possible shapes, we divide the parameters into two separate types: global geometric transformation parameters $\mathcal{Q}_g$ and "intrinsic" shape parameters $\mathcal{Q}_i$, with $\mathcal{Q} = \mathcal{Q}_g \times \mathcal{Q}_i$. The set of global geometric transformation parameters $\mathcal{Q}_g$ is a group; we take it to be the set of translations, rotations, and independent scalings in each dimension. The set of intrinsic shape parameters $\mathcal{Q}_i$ describes the more "interesting" aspects of the family of shapes, *e.g.* deformations.

$\mathcal{Q}_i$ may be any family of surfaces which is differentiable in its underlying parameters q_i. However, to emphasize the ability to use very coarse shape models, we will often use extremely simple affine models. More specifically, recall that $\mathcal{X}_q$ is a discrete set, in which case it may be written as a matrix $\mathcal{X}_q \in \mathbb{R}^{n \times 3}$ where n is the cardinality of the set. In this case, we can write our simple shape model as

$$\mathcal{X}_q = q_g R(Aq_i + b) \tag{12}$$

where $A \in \mathbb{R}^{3n \times k}$ and $b \in \mathbb{R}^{3n}$ specify the affine part of the model; R reshapes vectors of length $3n$ into matrices of size $n \times 3$; and q_g is the geometric transformation, which is applied to each row of the matrix separately. A and b can be learned via PCA from a small number of examples; and we will see that PCA models with very few modes k are sufficient for the computation of coverage in practice.

In order to improve the results of the network F, we can run test-time optimization using the same loss applied to the single instance of relevance. That is, we can initialize q as the output of the network $F(\mathcal{S}; \theta)$ and run gradient descent on $L_{\texttt{inf}}(q) = \frac{1}{|\mathcal{S}|} \sum_{s \in S} \sum_{x \in \mathcal{X}_q} w(s, x; \mathcal{X}_q) \|s - x\|$ for a small number of steps. This follows the practice of [32], which showed that networks for regressing pose and shape are less susceptible to local minima, and can provide good initializations for additional gradient descent steps. In practice, a small number of gradient descent steps is sufficient.

[1] We use 7-dimensional pose vectors, where 3 dimensions represent the camera translation and the remaining dimensions represent the camera rotation in quaternion format.

[2] The underlying assumption is that there is significance to the order in which the depth maps occur, as they are derived from the camera's traversal in $\mathbb{R}^3$. In the case where the partial view is the result of combining *unordered* depth maps, instead of using a 1D convolutional network, one may apply a network which ingests sets, such as in [71].

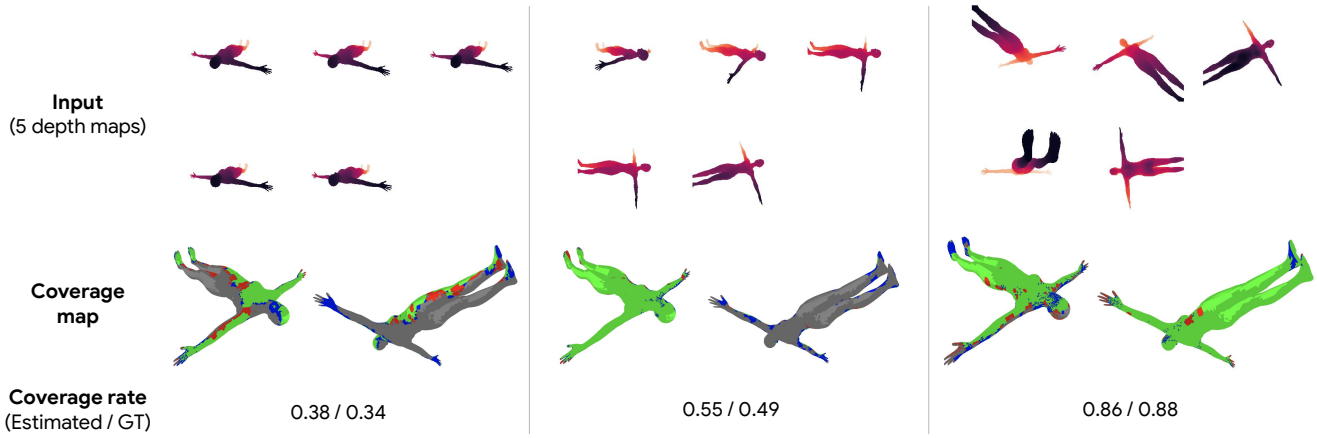

Figure 4: **Coverage estimation on body shapes.** *Top:* Input (partial view) depth maps; darker is closer. *Middle:* Local correspondence between our estimation and the ground-truth. Color coded texture indicates: green - correctly estimated covered area; gray - correctly estimated uncovered area; blue - estimated covered but actually uncovered area; red - estimated uncovered but actually covered area. *Bottom:* Estimated vs. ground-truth coverage rate.

4. Validation on Body Shapes

To validate our method in a clean, controlled environment, we first evaluate it on synthetic human body shapes generated by the Skinned Multi-Person Linear (SMPL) model [41]. In this experiment, we adopt a simplified setting where the parametric shape family is not a coarse approximation of the data model, but the exact data model. This will allow us to visualize our algorithm's behaviour. Note that we do not rely on such an assumption in general, and it will be dropped for the real world scenario in Sec. 5.

Shape Model The shape model is constructed by considerably reducing the SMPL model. While the SMPL model is parameterized by both articulated pose and body shape parameters, we freeze the pose parameters and only use the star-shaped (rest) body pose. The SMPL body shape family is represented by triangulated meshes of $n = 6890$ vertices spanned by 300 PCA vectors, where we use only the first $k = 10$ components which account for 97% of the data variance. The parametric shape model is thus given by eq. (12), where A is composed from the 10 SMPL shape components and b is given by the SMPL average body shape mesh.

Dataset Given this model, by randomly sampling the shape parameters q in (12), we generated 1960 training, 40 validation, and 300 test body meshes. To generate the input partial depth views, for each synthesized body shape we randomly sampled $N = 5$ virtual camera poses and rendered the corresponding depth maps using a z-buffer based approach [58]. By utilizing the underlying synthetic model used to generate the data, we can also compute the ground-truth coverage rate as the ratio between the surface area in the camera field-of-view and the total mesh surface areas. Note that only the partial depth maps / camera poses are used for unsupervised training and inference. The corresponding full body meshes and coverage rates are used solely for the purpose of visualization and evaluation.

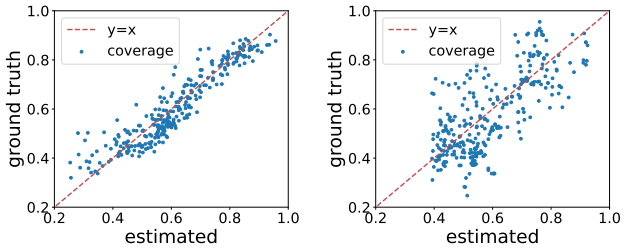

Figure 5: **Estimated vs. ground-truth coverage.** *Left:* For the body shapes experiment in Sec. 4. *Right:* For the synthetic colonoscopy experiment in Sec. 5.1.

Model Training The surface estimator was trained to minimize the loss in eq. (11) using the Adam optimizer [31] for 50 epochs with a batch size of 4 and learning rate of $2 \cdot 10^{-4}$. See the Supplementary for more training details.

Results After training concludes, we estimate the test samples' coverage with eq. (7). Figure 5 (left side) depicts the estimated vs. the ground-truth coverage for the test set, yielding a mean absolute coverage error of MAE $= 0.0440$. Figure 4 depicts our results on three samples. The top row shows the input depth maps, and the second row presents the local correspondence between our estimation and the ground-truth on the body mesh. The green/gray colors indicate correctly estimated covered/uncovered areas, and indeed the coverage maps are dominated by these two colors.

5. Coverage Estimation in Colonoscopies

Colorectal cancer (CRC) is the cause of an estimated 880K deaths globally per year [74]. For the most part, CRC is preventable by identifying and removing precancerous

polyps via colonoscopy screenings [36], yet successful prevention is dependent on the endoscopist's capabilities. It is approximated that over 20% of polyps are missed during these procedures [37], in part due to poor coverage of the colon surface. An *online* system which alerts on low coverage and allows for revisiting of the unseen areas within the ongoing procedure could augment the physicians' performance and reduce the missed polyp rate [11].

Polyps are missed for various reasons, some of which have been addressed by recent commercial polyp detection systems. The inherent limitation of current systems is the inability to detect polyps which *never appear in the camera field-of-view*. Furthermore, these recent AI-based systems may actually lead to a decrease in polyp detection rates, based on the clinicians "over-relying" on these systems and covering less colon surface [11]. As a result, leading clinicians have surfaced the need for complementary systems which ensure sufficient coverage [11].

We demonstrate the applicability of our coverage estimation technique for detecting deficient colon coverage. It is practically impossible to obtain ground-truth colon coverage on real colonoscopy videos, so one cannot derive a model in a supervised learning setting. A large dataset of 3D colon shapes is also unattainable, so deriving a fine colon shape model based on data-driven techniques is not an option. Nevertheless, our algorithm which is trained in an unsupervised manner and only requires a coarse shape model can predict colon coverage. A previous work [15] studied colon coverage, yet relied on ground-truth coverage obtained on synthetic data in a supervised setting.

5.1. Synthetic Colonoscopy Videos

We start by testing our method on synthetic colonoscopy videos, where we can obtain ground-truth coverage rates for the sake of evaluating performance. However, as opposed to Sec. 4, we do not assume that we have access to the complex 3D model used to create the data. The shape model here will be a truly coarse approximation of the data generation model, and will in fact be a "hand-crafted" model which is not derived from shape samples.

Dataset Our dataset consists of 32 synthetic colonoscopy videos, each including 10K frames consisting of RGB, depth map and camera pose data. These videos are based on the 3D colon model developed by 3D systems [75] and rendered using Blender [76]. Our dataset is split into 20/5/7 videos for training/validation/testing.

Preprocessing We train our model to estimate coverage on segments of 300 frames, corresponding to 10 seconds of video[3]. These segments are randomly extracted from the videos in an overlapping manner, and temporally downsampled by a factor of 15. The remaining $N = 20$ depth maps and the corresponding camera poses serve as the coverage estimator input. In total, 10K/100/400 segments are extracted for training/validation/testing from the train/validation/test set videos. As in Sec. 4, we compute the ground-truth coverage for each sample. Here as well, only the partial depth maps / camera poses are used for unsupervised training, and the ground-truth coverage rates exist solely for evaluation purposes.

[3]Our goal is to provide alerts on deficient colon coverage which allow the endoscopist to re-examine the segment within the ongoing procedure. Focusing on 10 second intervals allow such re-examinations without a significant increase in procedure time.

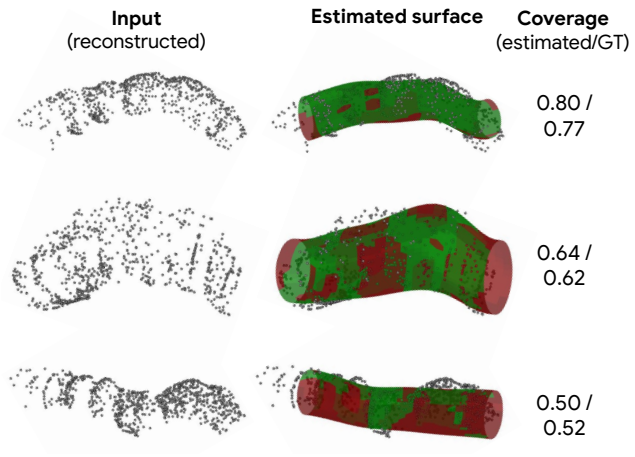

Figure 6: **Coverage estimation on synthetic colonoscopy segments.** *Left:* Point clouds reconstructed from the input depth maps and camera poses. *Middle:* Our algorithm's estimated shape surface, where green/red indicates seen/unseen areas. *Right:* The estimated/ground-truth shape coverage.

Shape Model With no dataset of colon shapes at hand, we manually design a crude model of colon segments. Colon segments are generally tube-shaped with perturbations of bends and curves. We start with a cylinder mesh of $n = 1500$ vertices. Then, we generate $m = 8000$ perturbed meshes, where for each the cylinder is bent by translating a set of vertices (see the Supplementary for details and examples). Each perturbed cylinder is represented by a vector $v \in \mathbb{R}^{3n}$ of the vertices' coordinates. All m vectors are stacked in a matrix $\mathbf{V} \in \mathbb{R}^{3n \times m}$, on which a PCA decomposition is performed and the first $k = 5$ components (which account for 98% of the data variance) are extracted. The shape model is given by eq. (12), where $A \in \mathbb{R}^{3n \times k}$ is composed from the top 5 PCA components, and $b \in \mathbb{R}^{3n}$ is the mean shape.

Model Training The model was trained to minimize the loss in eq. (11) using the Adam optimizer [31] for 10 epochs with a batch size of 8 and learning rate of $2 \cdot 10^{-4}$. See the Supplementary for more training details.

Results After training concludes, we estimate the test samples' coverage with eq. (7). Figure 5 (right side) plots the estimated vs. the ground-truth coverage for the test set, and

Method	Training	MAE
Physicians' annotation [15]		0.177
C2D2 [15]	Supervised	0.075
Ours	Unsupervised	0.092

Table 1: **Mean absolute error (MAE) on synthetic colonoscopy videos.** Our method does not require supervision (making it better-suited for *real* colonoscopy data). This comes at the cost of a 23% increase in MAE *on synthetic data*, which is still much better then physicians' ability to estimate colon coverage.

Table 1 compares our method's performance to prior work. Figure 6 visualizes our algorithm's process. Given a sequence of depth maps and camera poses (visualized here by the equivalent point cloud), the algorithm predicts the best fitting tube-shaped surface. Then, areas in proximity of points (colored in green) are considered seen, while the rest are considered unseen. Coverage is then estimated by taking the ratio of seen area to the total area.

5.2. Real Colonoscopy Videos

We now validate our method on real colonoscopy videos. It is impractical to obtain either ground-truth coverage or a dataset of colon shape samples in this setting, so here too we rely on a coarse "hand-crafted" model. Moreover, depth data and camera pose are not captured during colonoscopies. These must be estimated from raw RGB frames, introducing additional complexity into this experiment.

Dataset The dataset includes 470 deidentified videos taken during colonoscopy screening. These are split into 400/20/50 videos for training/validation/testing.

Preprocessing Similar to Sec. 5.1, we estimate coverage on segments of 200 frames, where 20K such training segments are randomly extracted in an overlapping manner from the videos. Yet unlike the synthetic data experiment, we only have RGB frames. Depth maps and camera poses are obtained by leveraging the technique of [5] for unsupervised monocular depth and camera egomotion estimation[4]. After obtaining the depth maps and poses, each sequence is temporally downsampled by a factor of 4, and the remaining $N = 50$ depth maps and poses serve as surface estimation model inputs. See the Supplementary for more details.

Shape Model and Training We use the same shape model as in Sec. 5.1. The network was trained to minimize the loss in eq. (11) using the Adam optimizer [31] for 10 epochs with a batch size of 2 and learning rate of $2 \cdot 10^{-5}$.

Processing time Inference requires 0.47 seconds (for a 6.66 second video segment) on a single GPU, which allows immediate re-examination within the ongoing procedure.

Results and Validation Figure 7 visualizes our algorithm's performance on real colonoscopy segments. Given

[4]We retrained this model on colonoscopy videos.

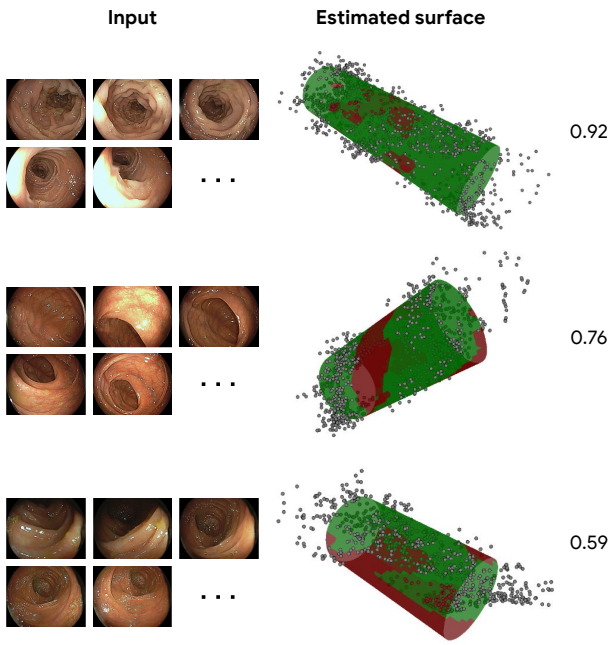

Figure 7: **Coverage estimation on real colonoscopy segments.** *Left:* Sample video frames of a colon segment. *Right:* Our algorithm's estimated shape surface, where green/red indicates seen/unseen areas, along with the estimated shape coverage.

only a sequence of RGB frames, we estimate the best fitting surface and predict the surface coverage via eq. (7). As ground-truth coverage is not attainable, expert Gastroenterologists (GIs) where asked to evaluate our algorithm's predictions. This study was planned carefully, bearing in mind that disagreement among GIs can be large when assessing colon coverage [15]. First, to remove ambiguous samples, we divide segments into two classes: (a) *mostly covered segment* - if the estimated coverage rate is over 0.8, and (b) *partially covered segment* - if the estimated coverage rate is under 0.65. Three GIs were shown 42 randomly chosen video segments (21 from each class) along with our predicted label, and asked: "does this description match the clip?" (yes/no answers). To obtain a reliable ground-truth set, we only take into account video segments on which all three GIs agreed on, which was the case for 20 segments. On this "ground-truth" set, the GI's found our algorithm's prediction to be correct on 85% of samples.

6. Conclusion

We have presented a new technique for estimating shape coverage, demonstrating its accuracy in the low data regime. Our technique enables novel applications, such as deficient coverage detection in colonoscopy screenings.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. In *ACM SIGGRAPH*, pages 408–416, 2005. 2

[2] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3769, 2014. 2

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Conference on Computer Graphics and Interactive Techniques*, pages 187–194, 1999. 2

[4] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578, 2016. 2

[5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2019. 8

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[7] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644, 2016. 2

[8] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2020. 2

[9] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3D-encoder-predictor cnns and shape synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017. 1, 2

[10] Amaury Dame, Victor A Prisacariu, Carl Y Ren, and Ian Reid. Dense reconstruction using 3D object shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1295, 2013. 2

[11] James E East and Jens Rittscher. Artificial intelligence for colonoscopic polyp detection: High performance versus human nature. *Journal of Gastroenterology and Hepatology*, 35(10), 2020. 2, 7

[12] Francis Engelmann, Jörg Stückler, and Bastian Leibe. Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In *German Conference on Pattern Recognition*, pages 219–230, 2016. 2

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–613, 2017. 2

[14] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. 2

[15] Daniel Freedman, Yochai Blau, Liran Katzir, Amit Aides, Ilan Shimshoni, Danny Veikherman, Tomer Golany, Ariel Gordon, Greg Corrado, Yossi Matias, and Ehud Rivlin. Detecting deficient coverage in colonoscopies. *IEEE Transactions on Medical Imaging*, 2020. 7, 8

[16] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 2

[17] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499, 2016. 2

[18] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *IEEE International Conference on Computer Vision*, pages 1381–1388, 2009. 2

[19] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4731–4740, 2015. 2

[20] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3D reconstruction with adversarial constraint. In *International Conference on 3D Vision*, pages 263–272, 2017. 2

[21] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *IEEE International Conference on Computer Vision*, pages 85–93, 2017. 1, 2

[22] Christian Hane, Nikolay Savinov, and Marc Pollefeys. Class specific 3D object shape priors using surface normals. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659, 2014. 2

[23] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3D object reconstruction. In *International Conference on 3D Vision*, pages 412–420, 2017. 2

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[25] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 2

[26] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1966–1974, 2015. 2

[27] Maciej Karaszewski, Marcin Adamczyk, and Robert Sitnik. Assessment of next-best-view algorithms performance with various 3D scanners and manipulator. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119:320–333, 2016. 2

[28] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 2

[29] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, volume 7, 2006. 2

[30] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013. 2

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 7, 8

[32] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *IEEE International Conference on Computer Vision*, pages 2252–2261, 2019. 2, 5

[33] Simon Korman, Eyal Ofek, and Shai Avidan. Peeking template matching for depth extension. In *IEEE International Conference on Computer Vision*, pages 2174–2182, 2015. 2

[34] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10(4):611–631, 2015. 2

[35] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6050–6059, 2017. 2

[36] Béatrice Lauby-Secretan, Nadia Vilahur, Franca Bianchini, Neela Guha, and Kurt Straif. The iarc perspective on colorectal cancer screening. *New England Journal of Medicine*, 378(18):1734–1740, 2018. 2, 7

[37] AM Leufkens, MGH Van Oijen, FP Vleggaar, and PD Siersema. Factors influencing the miss rate of polyps in a back-to-back colonoscopy study. *Endoscopy*, 44(05):470–475, 2012. 2, 7

[38] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3D reconstruction. In *Computer Graphics Forum*, volume 34, pages 435–446, 2015. 2

[39] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. *arXiv preprint arXiv:1706.07036*, 2017. 2

[40] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1886–1895, 2018. 2

[41] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, 2015. 2, 6

[42] Miguel Mendoza, J Irving Vasquez-Gomez, Hind Taud, L Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3D object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020. 2, 3

[43] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020. 5

[45] Charlie Nash and Christopher KI Williams. The shape variational autoencoder: A deep generative model of part-segmented 3D objects. In *Computer Graphics Forum*, volume 36, pages 1–12, 2017. 2

[46] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *International Conference on 3D Vision*, pages 484–494, 2018. 2

[47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 2

[48] Mark Pauly, Niloy J Mitra, Joachim Giesen, Markus H Gross, and Leonidas J Guibas. Example-based 3D scan completion. In *Symposium on Geometry Processing*, pages 23–32, 2005. 2

[49] Mark Pauly, Niloy J Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J Guibas. Discovering structural regularity in 3D geometry. In *ACM SIGGRAPH*, pages 1–11, 2008. 2

[50] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018. 2

[51] Gerard Pons-Moll, Jonathan Taylor, Jamie Shotton, Aaron Hertzmann, and Andrew Fitzgibbon. Metric regression forests for correspondence estimation. *International Journal of Computer Vision*, 113(3):163–175, 2015. 2

[52] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3D structure from images. In *Advances in Neural Information Processing Systems*, pages 4996–5004, 2016. 2

[53] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. OctNetFusion: Learning depth fusion from data. In *International Conference on 3D Vision*, pages 57–66, 2017. 1, 2

[54] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3D object shape from one depth image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015. 2

[55] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-DAE: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250, 2016. 1, 2

[56] Edward J Smith and David Meger. Improved adversarial systems for 3D object generation and reconstruction. In *Conference on Robot Learning*, pages 87–96, 2017. 2

[57] Shuran Song and Jianxiong Xiao. Sliding shapes for 3D object detection in depth images. In *European Conference on Computer Vision*, pages 634–651, 2014. 2

[58] W StraBer. *Schnelle Kurven-und Flachendarstellung auf graphischen Sichtgeraten*. PhD thesis, 1974. 6

[59] David Stutz and Andreas Geiger. Learning 3D shape completion from laser scan data with weak supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018. 1, 2

[60] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 2

[61] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–110, 2012. 2

[62] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *IEEE International Conference on Computer Vision*, volume 2, pages 1824–1831, 2005. 2

[63] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2634, 2017. 2

[64] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5236–5246, 2017. 2

[65] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3D interpreter network. In *European Conference on Computer Vision*, pages 365–382, 2016. 2

[66] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 2

[67] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 1, 2

[68] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 2

[69] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3D object reconstruction from a single depth view with adversarial learning. In *IEEE International Conference on Computer Vision Workshops*, pages 679–688, 2017. 2

[70] Sid Yingze Bao, Manmohan Chandraker, Yuanqing Lin, and Silvio Savarese. Dense object reconstruction with semantic priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1264–1271, 2013. 2

[71] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401, 2017. 5

[72] Rui Zeng, Wang Zhao, and Yong-Jin Liu. PC-NBV: a point cloud based deep network for efficient next best view planning. In *International Conference on Intelligent Robots and Systems*, pages 7050–7057, 2020. 2, 3

[73] Qian Zheng, Andrei Sharf, Guowei Wan, Yangyan Li, Niloy J Mitra, Daniel Cohen-Or, and Baoquan Chen. Non-local scan consolidation for 3D urban scenes. *ACM Transactions on Graphics*, 29(4):94–1, 2010. 2

[74] IARC Colorectal Cancer Fact Sheet 2018. https://gco.iarc.fr/today/data/factsheets/cancers/10_8_9-Colorectum-fact-sheet.pdf. 6

[75] 3D Systems GI Mentor Platform. https://simbionix.com/simulators/gi-mentor/gi-mentor/. 7

[76] Blender. https://www.blender.org/. 7