High-quality Textured 3D Shape Reconstruction with Cascaded Fully Convolutional Networks

Zheng-Ning Liu[†], Yan-Pei Cao[†], Zheng-Fei Kuang, Leif Kobbelt, Shi-Min Hu[*]

Abstract—We present a learning-based approach to reconstructing high-resolution three-dimensional (3D) shapes with detailed geometry and high-fidelity textures. Albeit extensively studied, algorithms for 3D reconstruction from multi-view depth-and-color (RGB-D) scans are still prone to measurement noise and occlusions; limited scanning or capturing angles also often lead to incomplete reconstructions. Propelled by recent advances in 3D deep learning techniques, in this paper, we introduce a novel computation and memory efficient cascaded 3D convolutional network architecture, which learns to reconstruct implicit surface representations as well as the corresponding color information from noisy and imperfect RGB-D maps. The proposed 3D neural network performs reconstruction in a progressive and coarse-to-fine manner, achieving unprecedented output resolution and fidelity. Meanwhile, an algorithm for end-to-end training of the proposed cascaded structure is developed. We further introduce *Human10*, a newly created dataset containing both detailed and textured full body reconstructions as well as corresponding raw RGB-D scans of 10 subjects. Qualitative and quantitative experimental results on both synthetic and real-world datasets demonstrate that the presented approach outperforms existing state-of-the-art work regarding visual quality and accuracy of reconstructed models.

Index Terms—high-fidelity reconstruction, 3D vision, cascaded architecture.

1 Introduction

High-quality reconstruction of 3D objects and scenes is key to 3D environment understanding, mixed reality applications, as well as the next generation of robotics, and has been one of the major frontiers of computer vision and computer graphics research for years [1], [2], [3], [4], [5]. Meanwhile, the availability of consumer-grade RGB-D sensors, such as the *Microsoft Kinect* and the *Intel RealSense*, involves more novice users to the process of scanning surrounding 3D environments, opening up the need for robust reconstruction algorithms which are resilient to errors in the input data (e.g., noise, distortion, and missing areas).

In spite of recent advances in 3D environment reconstruction, acquiring high-fidelity 3D shapes with imperfect data from casual scanning procedures and consumer-level RGB-D sensors is still a particularly challenging problem. Since the pioneering *KinectFusion* work [4], many 3D reconstruction systems, both real-time [1], [6], [7], [8], [9] and offline [5], have been proposed, which often use a volumetric representation of the scene geometry, i.e., the truncated signed distance function (TSDF) [10]. However, depth measurement acquired by consumer depth cameras contains a significant amount of noise, plus limited scanning angles lead to missing areas, making vanilla depth fusion suffer from blurring surface details and incomplete geometry. Another line of research [2], [11], [12] focuses on reconstructing complete geometry from noisy and sparsely-sampled point clouds, but cannot process point clouds with a large percentage of missing data and may produce bulging artifacts.

The wider availability of large-scale 3D model repositories [13], [14] stimulates the development of data-driven approaches for shape reconstruction and completion. Assembly-based methods, such as [15], [16], require carefully *segmented* 3D databases as input, operate on a few specific classes of objects, and can only generate shapes with limited variety. On the other hand, recent deep learning-based approaches [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] mostly focus on inferring 3D geometry from single-view images [18], [19], [21], [22], [24], [25], [26] or high-level information [20], [27] and often get stuck at low resolutions (typically 32^3 voxel resolution) due to high memory consumption, which is far too low for recovering geometric details.

In this work, we present a coarse-to-fine approach to high-fidelity volumetric reconstruction of 3D shapes from noisy and incomplete inputs using a 3D cascaded fully convolutional network (3D-CFCN) architecture, which outperforms state-of-the-art alternatives regarding the resolution and accuracy of reconstructed models. Our approach chooses recently introduced octree-based efficient 3D deep learning data structures [28], [29], [30] as the basic building block, however, instead of employing a standard single-stage convolutional neural network (CNN), we propose to use multi-stage network cascades for detailed shape information reconstruction, where the object geometry is predicted and refined progressively via a sequence of sub-networks. The rationale for choosing the cascaded struc-

- *Manuscript received XX XX, 20XX; revised XX XX, 20XX.*
- [*] *Corresponding author.*
- [†] *Zheng-Ning Liu and Yan-Pei Cao contributed equally to this work (joint first author).*

- *Zhe-Ning Liu is with the Department of Computer Science and Technology, Tsinghua University. Email: lzhengning@gmail.com.*
- *Yan-Pei Cao is with the Department of Computer Science and Technology, Tsinghua University and Owlii Inc. Email: caoyanpei@gmail.com*
- *Zheng-Fei Kuang is with the Department of Computer Science and Technology, Tsinghua University. Email: kzf15@mails.tsinghua.edu.cn.*
- *Leif Kobbelt is with Visual Computing Institute of RWTH Aachen University Email: kobbelt@cs.rwth-aachen.de.*
- *Shi-Min Hu is with the Department of Computer Science and Technology, Tsinghua University. Email: shimin@tsinghua.edu.cn.*

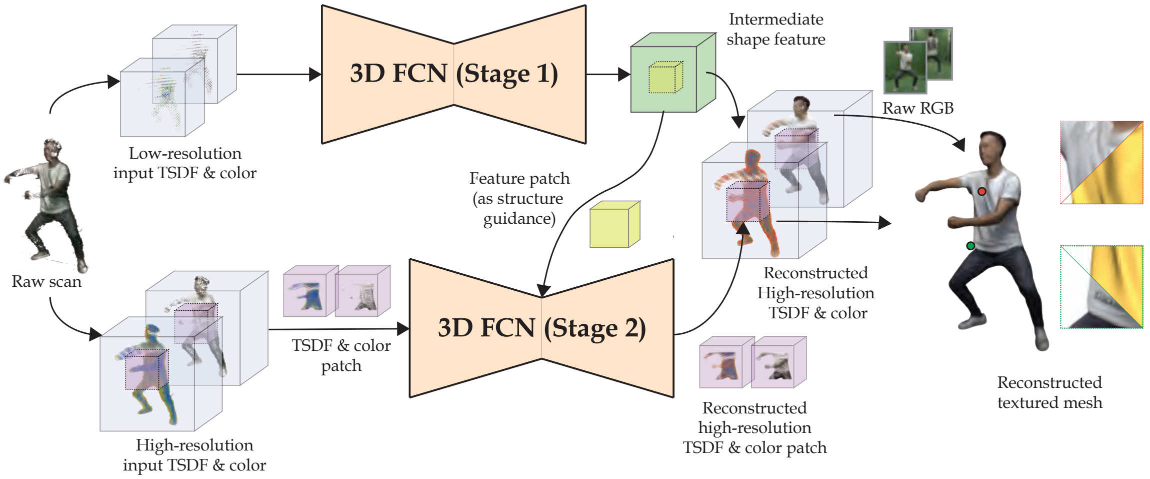
3D FCN (Stage 1)
Intermediate shape feature
Raw RGB
Low-resolution input TSDF & color
Feature patch (as structure guidance)
Raw scan
TSDF & color patch
3D FCN (Stage 2)
Reconstructed High-resolution TSDF & color
Reconstructed textured mesh
Reconstructed high-resolution TSDF & color patch
High-resolution input TSDF & color

Fig. 1: Illustration of a two-stage 3D-CFCN architecture. Given partial and noisy raw RGB-D scans as input, a fused low-resolution TSDF-color volume is fed to the stage-1 3D fully convolutional network (3D-FCN), producing an intermediate representation. Exploiting this intermediate feature, the network then 1) regresses a low-resolution but complete TSDF-color volume and 2) predicts which volumetric patches should be further refined. For each patch that needs further refinements, the corresponding block is cropped from a fused high-resolution TSDF-color input, and the stage-2 3D-FCN uses it to infer a detailed high-resolution local TSDF-color volume, which substitutes the corresponding region in the aforementioned regressed TSDF-color volume and thus improves the output's resolution. Note a patch of the global intermediate representation also flows into stage 2 to provide structure guidance. And lastly, predicted color is blended with raw RGB images to produce the final texture map. The rightmost column shows the high-quality reconstruction. Close-ups show accurately reconstructed geometry and appearance details, e.g., wrinkles and texts on clothes. Note the input scan is fused from 2 viewpoints.

ture is two-fold. First, to predict high-resolution (e.g., 512^3, 1024^3, or even higher) geometry information, one may have to deploy a deeper 3D neural network, which could significantly increase memory requirements even using memory-efficient data representations. Second, by splitting the geometry inference into multiple stages, we also simplify the learning tasks, since each sub-network now only needs to learn to reconstruct 3D shapes at a certain resolution.

Training a cascaded architecture is a nontrivial task, particularly when octree-based data representations are employed, where both the *structure* and the *value* of the output octree need to be predicted. We thus design the sub-networks to learn where to refine the 3D space partitioning of the input volume, and the same information is used to guide the data propagation between consecutive stages as well, which makes end-to-end training feasible by avoiding exhaustively propagating every volume block.

While geometry information provided by high-resolution shape reconstructions enables applications such as shape analysis and physical simulation, obtaining correct geometric shapes is just the beginning step of 3D reconstruction and modeling; recovering accurate color information and appearances of 3D shapes is also essential for human and machine perception. Therefore, based on our cascaded network architecture, we further introduce an integrated method for texture reconstruction. Geometry and color information of target objects are jointly learned for consistency and efficiency. However, generating high-frequency texture details remains hard for neural networks even with high output spatial resolutions, and thus we propose to blend the predicted color and projective texture maps, thereby leading to complete texture maps with rich fine details.

The primary contribution of our work is a novel learning-based, progressive approach for high-accuracy 3D shape reconstruction from imperfect data, which also comes with a hybrid method for recovering high-fidelity shape textures. To train and quantitatively evaluate our model on real-world 3D shapes, we also contribute a dataset containing both detailed full body reconstructions and raw RGB-D scans of 10 subjects. We then conduct careful experiments on both simulated and real-world datasets, comparing the proposed framework to a variety of state-of-the-art alternatives. These experiments show that, when dealing with noisy and incomplete inputs, our approach produces 3D shapes with significantly higher accuracy and quality than other existing methods.

Our initial work has been published in the European Conference on Computer Vision 2018 [31]. In this paper, we further address the issue of color information reconstruction, aiming to recover complete and detailed texture maps. In addition, we provide more quantitative evaluations, including more comparisons with state-of-the-art methods and computational consumption analyses of the proposed cascaded architecture.

2 Related Work

2.1 3D Shape Reconstruction

There has been a large body of work focused on 3D reconstruction over the past a few decades. We refer the reader to [32] and [33] for detailed surveys of methods for reconstructing 3D objects from point clouds and RGB-D streams, respectively. Here we only summarize the most relevant previous approaches and categorize them as geometric, assembly-based, and learning-based approaches.

Geometric Approaches. In the presence of sample noise and missing data, many choose to exploit the smoothness assumption, which constrains the reconstructed geometry to satisfy a certain level of smoothness. Gradient-domain methods [2], [34], [35] require that the input point clouds be equipped with (oriented) normals and utilize them to estimate an implicit soft indicator function which discriminates the interior region from the exterior of a 3D shape. Similarly, [36], [37] use globally supported radial basis functions (RBFs) to interpolate the surface. On the other hand, a series of moving least squares (MLS) -based methods [38], [39] attack 3D reconstruction by fitting the input point clouds to a spatially varying low-degree polynomial. By assuming local or global surface smoothness, these approaches, to a certain extent, are robust to noise, outliers, and missing data.

Sensor visibility is another widely used prior in scan integration for object and scene reconstruction [10], [40], which acts as an effective regularizer for structured noise [41] and can be used to infer empty spaces. For large-scale indoor scene reconstruction, since the prominent Kinect-Fusion, plenty of systems [1], [5], [9] have been proposed. However, they are mostly focused on improving the accuracy and robustness of camera tracking in order to obtain a globally consistent model.

Compared to these methods, we propose to learn natural 3D shape priors from massive training samples for shape completion and reconstruction, which better explores the 3D shape space and avoids undesired reconstructed geometries resulted from hand-crafted priors.

Assembly-based Approaches. Another line of work assumes that a target object can be described as a composition of primitive shapes (e.g., planes, cuboids, spheres, etc.) or known object parts. [42], [43] detect primitives in input point clouds of CAD models and optimize their placement as well as the spatial relationship between them via graph cuts. The method introduced in [44] first interactively segments the input point cloud and then retrieves a complete and similar 3D model to replace each segment, while [16] extends this idea by exploiting the contextual knowledge learned from a scene database to automate the segmentation as well as improve the accuracy of shape retrieval. To increase the granularity of the reconstruction to the object component level, [15] proposes to reassemble parts from different models, aiming to find the combination of candidates which conforms the input RGB-D scan best. Although these approaches can deal with partial input data and bring in semantic information, 3D models obtained by them still suffer from the lack of geometric diversity.

Learning-based Approaches. 3D deep neural networks have achieved impressive results on various tasks [13], [45], [46], such as 3D shape classification, retrieval, and segmentation. As for generative tasks, previous research mostly focuses on inferring 3D shapes from (single-view) 2D images, either with only RGB channels [17], [18], [19], [20], [21], [22], [23], or with depth information [24], [25], [26]. While showing promising advances, these techniques are only capable of generating rough 3D shapes at low resolutions. Similarly, in [27], [47], shape completion is also performed on low-resolution voxel grids due to the high demand of computational resources.

Another series of research focuses on recovering shape and pose of 3D human body from single [48], [49] or multiple [50], [51], [52] images. Kanazawa et al. [48] propose an end-to-end deep learning framework for recovering 3D meshes of human bodies from monocular RGB images by inferring parameters of the SMPL model [53]; however, SMPL does not have sufficient degrees of freedom for expressing geometric details such as clothes and hand poses, thus the results of [48] may suffer from limited fidelity. Alldieck et al. [50] extend SMPL by introducing pervertex offsets to the template mesh, take as input monocular videos and reconstruct 3D human body models which allow personalized geometry variations while still restrict the topology to be the same as SMPL. Varol et al. [49] predict 3D human body shapes from natural images under a multi-task learning framework, using volumetric occupancy maps as 3D representation for the neural network. Trained on a large synthetic dataset, [49] achieves state-of-the-art accuracy for body shape estimation, nevertheless its final outputs are still of limited spatial resolution (up to 128^3). Huang et al. [51] propose to reconstruct 3D probability fields from a sparse set of calibrated multi-view images and can produce promising reconstruction results; for each voxel in the volume, the method associates it with a feature vector which is fused from the convolutional features of its corresponding pixels on different 2D images, then a probability of being on the surface is inferred using a classification network. However, since each voxel is processed separately and independently in [51], the estimated probability field (and hence the final surface) does not always guarantee to be clean and smooth. In contrast, our approach incorporates both local and global context during reconstruction, thus can recover consistent geometric structures and details at each scale. Note that different from aforementioned human body reconstruction methods, our approach does not assume any predefined parametric models and thus can be applied to arbitrary objects.

Aiming to complete and reconstruct 3D shapes at higher resolutions, [54] proposes a 3D Encoder-Predictor Network (3D-EPN) to firstly predict a coarse but complete shape volume and then refine it via an iterative volumetric patch synthesis process, which copy-pastes voxels from k-nearest-neighbors to improve the resolution of each predicted patch. [55] extends 3D-EPN by introducing a local 3D CNN to perform patch-level surface refinement. However, these methods both need separate and time-consuming steps before local inference, either nearest neighbor queries [54], or 3D boundary detection [55]. By contrast, our approach only requires a single forward pass for 3D shape reconstruction and produces higher-resolution results (e.g., 512^3 vs. 128^3 or 256^3). Inspired by recent image denoising and super-resolution algorithms, [52] employs a symmetric autoencoder for refining probabilistic visual hulls derived from a sparse set of viewpoints. Nevertheless, to deal with high-resolution volume data, it chooses to perform the reconstruction in a sliding window fashion, which is time-consuming and abandons the global context, making it hard to complete large holes in the input data. On the other hand, [29], [56] propose efficient 3D convolutional architectures by using octree representations, which are designed to decode high-resolution geometry information from dense intermediate features; nevertheless, no volumetric convolutional encoders and corresponding shape reconstruction architectures are provided. While [3] presents an OctNet-based [28] end-to-end deep learning framework for depth fusion, it refines the intermediate volumetric output globally, which makes it infeasible for producing reconstruction results at higher resolutions even with memory-efficient data structures. Instead, our 3D-CFCN learns to refine output volumes at the level of local patches, and thus significantly reduces the memory and computational cost.

2.2 Texture Reconstruction

High-quality texture acquisition plays an equally important role in 3D reconstruction and is a challenging task when considering limited number of views, uncontrolled illumination conditions, and imprecise geometry reconstructions. Most previous work adopts the projective texture mapping approach that maps RGB images onto reconstructed geometry. For example, Collet et al. [57] and Orts et al. [58] apply direct image projection with normal-weighted blending in their well-controlled lighting environments. In general cases, Gal et al. [59] minimize visible texture seams between texture patches via global optimization where compatible textures are assigned to adjacent triangles. They perform a multi-label graph-cut optimization and refine the label under a coarse-to-fine scheme. *TextureMontage*, proposed by Zhou et al. [60], partitions the mesh and the images using feature correspondences, and takes the surface texture inpainting technique as an additional post-processing step. Zhou et al. [61] globally optimize the camera poses for all input images together with non-rigid correction functions, resolving texture patch misalignments caused by inaccurate geometry and camera poses. However, the global optimization steps involved in above approaches also implies very long computation time in order of minutes per image. Very recently, Du et al. [62] introduces *Montage4D*, a real-time solution for mutli-view texture blending with per-vertex, geodesics-guided, and view-dependent weights, reducing blurring effect and visible texture seams. Nonetheless, it does not address the issue of insufficient viewing angles.

Image-based Rendering. The image-based rendering (IBR) technique is becoming an effective way to achieve view-dependent visual effects such as highlights [63], [64], [65], [66]. Previous researchers use per-view input information, such as per-view geometry and super-pixel over-segmentation to preserve depth boundaries even with imprecise 3D reconstructions [67], [68]. View-dependent texture mapping, which is initially proposed by Debevec et al. [65], [66], is widely used in IBR systems. They blend potential source images using angles between the novel view and source views as blending weights. Eisemann et al. [67] propose a view-dependent texturing algorithm that consists of a symbiosis between classical linear interpolation and optical flow-based warping refinement to correct for local texture misalignments and warping the textures accordingly in the rendered image domain. Chaurasia et al. [68] introduce a IBR method which is robust to missing or unreliable geometry by over-segmenting input images into super-pixels and improve poorly reconstructed areas based on a graph structure. In terms of indoor navigation, Hedman et al. [69] combine indoor-friendly depth sensors and multi-view stereo for improved reconstruction and propose a scalable rendering algorithm which uses mesh simplification and tiling to accelerate free-viewpoint IBR of indoor scenes. The fidelity of IBR depends critically on the quality and *quantity* of images used. In contrast, our approach can deliver plausible visual effects with only a few (e.g., 2 or 4) input images.

Learning-based Solutions. Fitzgibbon et al. [70] pose the problem of novel view synthesis as a learning problem, optimizing the novel-view image to match the statistics of example input patches. More recent work [71], [72], [73], [74], [75] apply CNNs to novel view prediction. Instead of synthesizing novel views from scratch, Zhou et al. [71] train a CNN to learn *appearance flows*, i.e., 2-D coordinate vectors specifying which pixels in the input view could be used, to reconstruct the target view. Via a generative completion network, Park et al. [72] further refine the invisible parts in the input image after predicting the flow. Hedman et al. [73] present a deep learning approach to blending for interactive IBR, where they use the held-out strategy on real image data to learn blending weights for combining input photo contributions. Martin-Brualla et al. [74] propose a re-rendering method that learns to enhance coarse renderings to high resolution and high quality stereo images for VR and AR applications. While they design a loss for stereo consistency, their method cannot extend to full texture atlas reconstruction. Although promising, these approaches either suffer from low-resolution results and visual artifacts or are too computational intensive for real-time applications. Instead, our method can recover textures in invisible areas while adding zero overhead during rendering time.

3 Method

This section introduces our 3D-CFCN model. We first give a condensed review of relevant concepts and techniques in Sec. 3.1. Then we present the proposed architecture and its corresponding training pipeline in Sec. 3.2 and Sec. 3.3. Sec. 3.4 summaries the procedure of collecting and generating the data which we used for training our model. And finally in Sec. 3.5, details of our texture reconstruction method are given.

3.1 Preliminaries

3.1.1 Volumetric Representation and Integration

The choice of underlying data representation for fusing depth measurements is key to high-quality 3D reconstruction. Approaches varies from point-based representations [76], [77], 2.5D fields [78], [79], to volumetric methods based on occupancy maps [80] or implicit surfaces [1], [10]. Among them, TSDF-based volumetric representations have become the preferred method due to their ability to model continuous surfaces, efficiency for incremental updates in parallel, and simplicity for extracting surface interfaces. In this work, we adopt the definition of TSDF from [4]:

$$\mathrm{V}(\mathbf{p}) = \Psi(\mathrm{S}(\mathbf{p})), \tag{1}$$

$$\mathrm{S}(\mathbf{p}) = \begin{cases} \|\mathbf{p} - \partial\Omega\|_2, & if\ \mathbf{p} \in \Omega \\ -\|\mathbf{p} - \partial\Omega\|_2, & if\ \mathbf{p} \in \Omega^c \end{cases}, \tag{2}$$

$$\Psi(\eta) = \begin{cases} \min(1, \frac{\eta}{\mu})\,\mathrm{sgn}(\eta), & if\ \eta \geq -\mu \\ invalid, & otherwise \end{cases}, \tag{3}$$

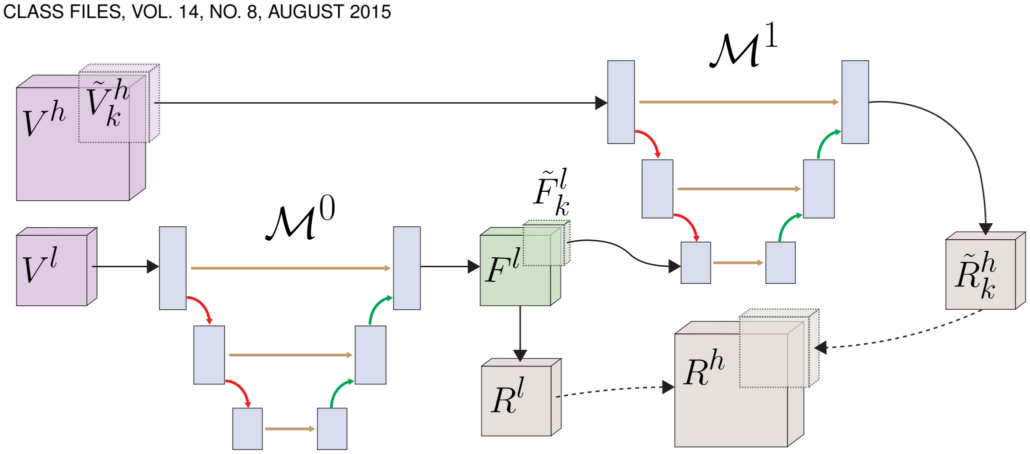

Fig. 2: Architecture of a two-stage 3D-CFCN. In this case, the network takes a pair of low- and high-resolution (i.e., 128^3 and 512^3) noisy and incomplete TSDF volume $\{V_l, V_h\}$ as input, and produces a refined TSDF volume at 512^3 voxel resolution. For conciseness, we only demonstrate the data flow for shape reconstruction in this figure.

where S is the standard signed distance function (SDF) with Ω being the object volume, and Ψ denotes the truncation function with μ being the corresponding truncation threshold. The truncation is performed to avoid surface interference, since in practice during scan fusion, the depth measurement is only locally reliable due to surface occlusions. In essence, a TSDF obliviously encodes free space, uncertain measurements, and unknown areas.

Given a set of depth scans at hand, we follow the approach in [10] to integrate them into a TSDF volume:

$$\mathrm{V}(\mathbf{p}) = \frac{\sum w_i(\mathbf{p})\,\mathrm{V}_i(\mathbf{p})}{\sum w_i(\mathbf{p})}, \tag{4}$$

where $\mathrm{V}_i(\mathbf{p})$ and $w_i(\mathbf{p})$ are the TSDFs and weight functions from the i-th depth scan, respectively.

3.1.2 OctNet

3D CNNs are a natural choice for operating TSDF volumes under the end-to-end learning framework. However, the cubic growth of computational and memory requirements becomes a fundamental obstacle for training and deploying 3D neural networks at high resolution. Recently, there emerges several work [28], [29], [30] that propose to exploit the sparsity in 3D data and employ octree-based data structures to reduce the memory consumption, among which we take OctNet [28] as our basic building block.

In OctNet, features and data are organized in the *grid-octree* data structure, which consists of a grid of shallow octrees with maximum depth 3. The structure of shallow octrees are encoded as bit strings so that the features and data of sparse octants can be packed into continuous arrays. Common operations in convolutional networks (e.g., convolution, pooling and unpooling) are defined on the grid-octree structure correspondingly. Therefore, the computational and memory cost are significantly reduced, while the OctNet itself, as a processing module, can be plugged into most existing 3D CNN architectures transparently. However, one major limitation of OctNet is that the structure of grid-octrees is determined by the input data and keeps fixed during training and inference, which is undesirable for *reconstruction* tasks where hole filling and detail refinement need to be performed. We thus propose a Structure Refinement Module which refines the octree structure on-the-fly to eliminate this drawback in Sec. 3.2.

3.2 Architecture

Our 3D-CFCN is a cascade of volumetric reconstruction modules, which are OctNet-based fully convolutional subnetworks aiming to infer missing surface areas and refine geometric details. Each module $\mathcal{M}^i$ operates at a given voxel resolution and spatial extent. We find *512^3 voxel resolution* and a corresponding *two-stage* architecture suffice to common daily 3D scanning tasks in our experiments, and thus will concentrate on this architecture in the rest of the paper; nevertheless, the proposed 3D-CFCN framework can be easily extended to support arbitrary resolutions and number of stages. We choose to employ a multi-stage cascaded scheme for two main reasons. Firstly, while one could deploy a single-stage *deep* 3D neural network to infer high-resolution shape and color information, the computational and memory cost would increase drastically even using memory-efficient data structures, as the network needs to process the volumetric feature globally at each resolution. Secondly, we make the learning tasks simpler by dividing shape and color prediction into multiple stages, as each subnetwork is only responsible for doing reconstruction at a certain resolution and scale.

In our implementation, for both sub-networks, we adopt the U-net architecture [81] while substituting convolution and pooling layers with the corresponding operations from OctNet. Skip connections are also employed between corresponding encoder and decoder layers to make sure the structures of input volumes are preserved in the inferred output predictions. To complete the partial input data and refine its grid-octree structure, we refrain from using OctNet's unpooling operation and propose a *structure refinement module*, which learns to predict whether an octant needs to be split for recovering finer geometric details. Note that even if the memory footprint of running a *single* module using dense volumetric representation is acceptable, employing OctNet is crucial to multi-stage joint training, batch training and inference time efficiency (see Sec 4.5, Table 3 and Table 4).

The first sub-network, $\mathcal{M}^0$, receives the encoded low-resolution (i.e., 128^3) TSDF volume V^l (see Sec. 3.4), which is fused from raw depth scans $\{\mathcal{D}_i\}$ of an 3D object $\mathcal{S}$, as input and produces a feature map F^l as well as a reconstructed TSDF volume R^l at the same resolution. Then for each 16^3

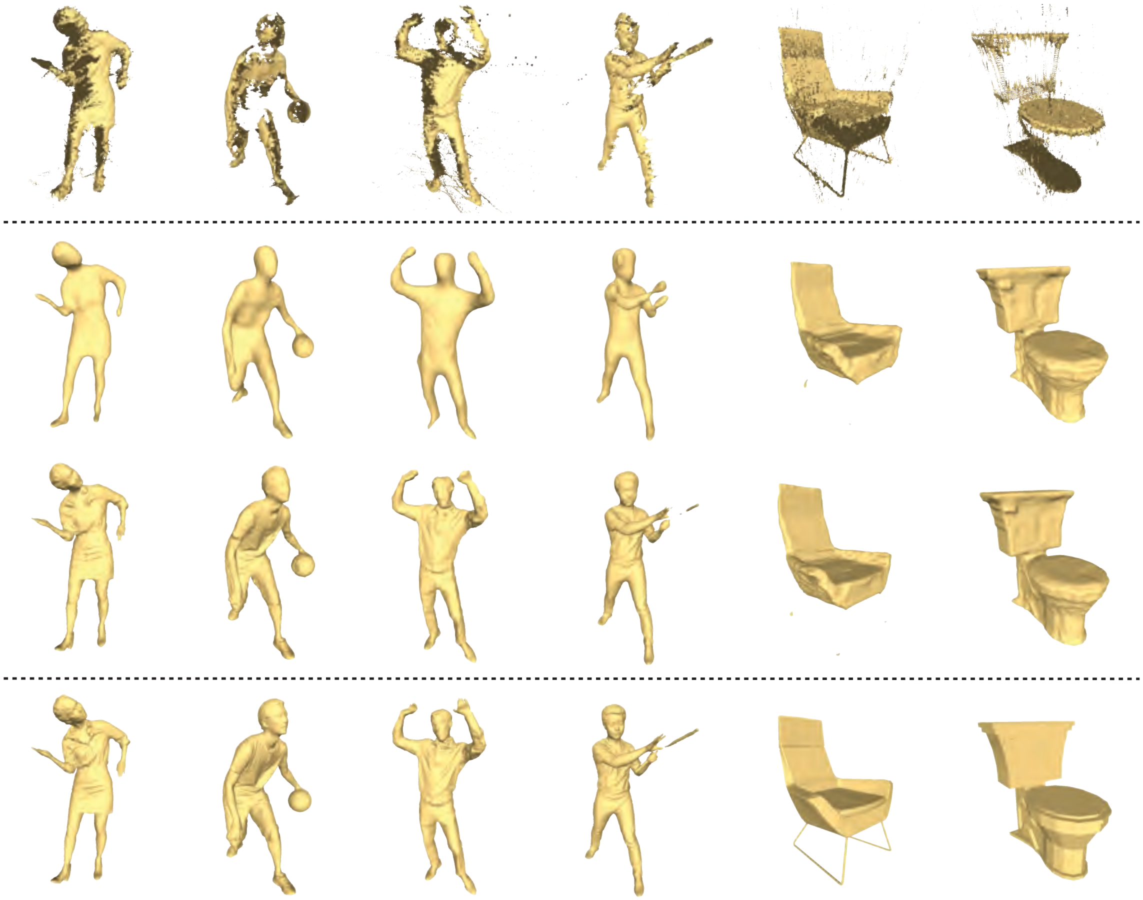

Fig. 3: Results gallery of high-accuracy shape reconstruction. Top row: Input scans fused from 2 randomly picked viewpoints. Second row: Reconstruction results of the first stage of our 3D-CFCN. Third row: Full-resolution reconstruction results of the two-stage 3D-CFCN architecture. Bottom row: Ground-truth references.

patch $\tilde{F}_k^l$ of F^l, we use a modified structure refinement module to predict if its corresponding block in R^l needs further improvement.

If a TSDF patch $\tilde{R}_k^l$ is predicted to be further refined, we then crop its corresponding 64^3 patch $\tilde{V}_k^h$ from V^h, which is an encoded TSDF volume fused from the same depth scans $\{\mathcal{D}_i\}$, but at a higher voxel resolution, i.e., 512^3. $\tilde{V}_k^h$ is next fed to the second stage $\mathcal{M}^1$ to produce a local feature map $\tilde{F}_k^h$ with increased *spatial* resolution and reconstruct a more detailed local 3D patch $\tilde{R}_k^h$ of $\mathcal{S}$. Meanwhile, since input local TSDF patches $\{\tilde{V}_k^h\}$ may suffer from a large portion of missing data, we also propagate $\{\tilde{F}_k^l\}$ to incorporate global guidance. More specifically, a propagated $\tilde{F}_k^l$ is concatenated with the high-level 3D feature map after the second pooling layer in $\mathcal{M}^1$ (see Fig. 2). Note this extra path, in return, also helps to refine F^l during back propagation. Finally, the regressed local TSDF patch $\{\tilde{R}_k^h\}$ is substituted back into the global TSDF, which can be further used to extract surfaces.

To avoid inconsistency across TSDF patch boundaries, we add interval overlaps when cropping feature maps and TSDF volumes. When cropping $\{\tilde{F}_k^l\}$, we expand two more voxels on each side of the 3D patch, making the actual resolution of $\{\tilde{F}_k^l\}$ grow to 20^3; similarly, for $\{\tilde{V}_k^h\}$ and $\{\tilde{F}_k^h\}$, we apply 8-voxel overlapping and increase their resolution to 80^3. However, when substituting back $\{\tilde{R}_k^h\}$, overlapping regions are discarded. So in its essence, this cropping approach acts as a smart padding scheme. Note that all local patches are still organized in grid-octrees.

Structure Refinement Module. Since the unpooling operation of OctNet restrains the possibility of refining the octree structure on-the-fly, inspired by [3], [29], we propose to replace unpooling layers with a structure refinement module. Instead of inferring new octree structures implicitly from reconstructions as in [3], we use 3^3 convolutional filters to directly predict from feature maps whether an octant should be further split. In contrast, OGN [29] predicts three-state masks using 1^3 filters followed by three-way softmax. To determine if a 3D local patch needs to be fed to $\mathcal{M}^1$, we take the average "split score" of all the octants in this patch and compare it with a confidence threshold ρ $(= 0.5)$. By employing this adaptive partitioning and propagation scheme, we achieve high-resolution volumetric reconstruction while keeping the computational and memory cost to a minimum level.

3.3 Training

The 3D-CFCN is trained in a supervised fashion on a TSDF dataset $\{\mathcal{F}_n = \{V^l, V^h, G^l, G^h\}\}$ in two phases, where V^l and V^h denote the incomplete input TSDFs at low and high voxel resolution, while G^l and G^h are low- and high-resolution ground-truth TSDFs, respectively.

In the first phase, $\mathcal{M}^0$ is trained alone with a hybrid of ℓ_1, binary cross entropy, and structure loss:

$$\mathcal{L}(\theta;\, V^l, G^l) = \mathcal{L}_{\ell_1} + \lambda_1 \mathcal{L}_{bce} + \lambda_2 \mathcal{L}_s. \tag{5}$$

The ℓ_1 term is designed for TSDF denoising and reconstruction. Let N represent the output resolution of current stage, $\mathbf{v}$ be a voxel in the volume, and P, G be the predicted and reference TSDFs, respectively. The ℓ_1 loss is then defined as

$$\mathcal{L}_{\ell_1} = \frac{1}{N^3} \sum_{\mathbf{v}} |P(\mathbf{v}) - G(\mathbf{v})|. \tag{6}$$

We additionally predict the signs of TSDF entries and employ an auxiliary binary cross entropy loss $\mathcal{L}_{bce}$ to provide the network with more guidance for learning shape completion:

$$\begin{aligned}\mathcal{L}_{bce} = \frac{1}{N^3} \sum_{\mathbf{v}} & sgn(G(\mathbf{v})) \cdot \log(S(\mathbf{v})) + \\ & (1 - sgn(G(\mathbf{v}))) \log(1 - S(\mathbf{v})),\end{aligned} \tag{7}$$

where S is the predicted probability of the TSDF value of a voxel being positive. In our experiments, we find $\mathcal{L}_{bce}$ also leads to faster convergence. Our structure refinement module is learned with $\mathcal{L}_s$, where

$$\mathcal{L}_s = \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} BCE\left(1 - f(o', T_{gt}), p(o)\right). \tag{8}$$

Here, $\mathcal{O}$ represents the set of octants in the current grid-octree, and BCE denotes the binary cross entropy. $p(o)$ is the prediction of whether the octant o should to be split, while o' is the corresponding octant of o in the ground-truth grid-octree structure T_{gt} (in this case, the structure of G_l). We define $f(o', T_{gt})$ as an indicator function that identifies whether o' exists in T_{gt}:

$$f(o', T_{gt}) = \left\{ \begin{array}{ll} 1, & \exists\, \tilde{o}',\ such\ that\ h(\tilde{o}') \leq h(o') \\ 0, & otherwise \end{array} \right., \tag{9}$$

where h denotes the height of an octant in the octree.

Furthermore, we employ multi-scale supervision [46], [82] to alleviate potential gradient vanishing. Specifically, after each pooling operation, the feature map is concatenated with a downsampled input TSDF volume at the corresponding resolution, and we evaluate the downscaled hybrid loss at each structure refinement layer.

In the second phase, $\mathcal{M}^1$ is trained; at the same time, $\mathcal{M}^0$ is being fine-tuned. To alleviate over-fitting and speed up the training process, among all the local patches that are predicted to be fed to $\mathcal{M}^1$, we keep only K of them randomly and discard the rest (we set $K = 2$ across our experiments). At this stage, the inferred global structure $\tilde{F}_k^l$ flows into $\mathcal{M}^1$ to guide the shape completion, while the refined local features also provide feedbacks and improves $\mathcal{M}^0$. The same strategy, i.e., hybrid loss (see Eq. 5) and multi-scale supervision, is adopted here when training $\mathcal{M}^1$ together with $\mathcal{M}^0$.

3.4 Training Data Generation

3.4.1 Synthetic Dataset

Our first dataset is built upon the synthetic 3D shape repository ModelNet40 [13]. We choose a subset of 10 categories, with 4051 shape instances in total (3245 for training, 806 for testing). Similar to existing approaches, we set up virtual cameras around the objects[1] and render depth maps, then simulate the volumetric fusion process [10] to generate ground-truth TSDFs. To produce noisy and partial training samples, previous methods [1], [3], [55] add random noise and holes to the depth maps to mimic sensor noise. However, synthetic noise reproduced by this approach usually does not conform real noise distributions. Thus, we instead implement a synthetic stereo depth camera [83]. Specifically, we virtually illuminate 3D shapes with a structured light pattern, which is extracted from *Asus XTion* sensors using [84], [85], and apply the PatchMatch Stereo algorithm [86] to estimate disparities (and hence depth maps) across stereo speckle images. In this way, the distribution of noise and missing area in synthesized depth images behaves much closer to real ones, thus makes the trained network generalize better on real-world data. In our experiments, we pick 2 or 4 virtual viewpoints randomly when generating training samples.

In essence, apart from shape completion, learning volumetric depth fusion is to seek a function $g(\{\mathcal{D}_1, \ldots, \mathcal{D}_n\})$ that maps raw depth scans to a noise free TSDF. Therefore, to retain information from all input depth scans, we adopt the histogram-based TSDF representation (TSDF-Hist) proposed in [3] as the encoding of our input training samples. A 10D smoothed-histogram, which uses 5 bins for negative and 5 bins for positive distances, with the first and the last bin reserved for truncated distances, is allocated for each voxel. The contribution of a depth observation is distributed linearly between the two closest bins. For outputs, we simply choose plain 1-dimensional TSDFs as the representation.

Since we employ a cascaded architecture and use multi-scale supervision during network training, we need to generate training and ground-truth sample pairs at multiple resolutions. Specifically, TSDFs at 32^3, 64^3, 128^3, 256^3, and 512^3 voxel resolutions are simultaneously generated in our experiments.

3.4.2 Real-world Dataset

We construct a high-quality dynamic 3D reconstruction (or, free-viewpoint video, FVV) system similar to [57] and collect 10 4D sequences of human actions, each capturing a different subject. Then a total of 9746 frames are randomly sampled from the sequences and split into training and test set by the ratio of $4:1$. We name this dataset as *Human10*[2]. For each frame, we fuse 2 or 4 randomly picked *raw scans* and obtain the TSDF-Hist encodings of the training sample; while the ground-truth TSDFs is produced by virtually scanning (see the previous section) the corresponding output triangle mesh of our FVV system. The sophisticated pipeline of our FVV system guarantees the quality and accuracy of the output mesh and texture, however, the design and details of the FFV system is beyond the scope of this paper.

3.5 Texture Reconstruction

After recovering the geometric models of target objects, we then reconstruct the corresponding appearance information to further improve visual realism. One straightforward solution is to directly project observed RGB images onto the predicted geometry. However, the texture will be incomplete

[1] We place virtual cameras at the vertices of a icosahedron.

[2] https://lzhengning.github.io/human10/

due to occlusion and limited number of view points. Also, imprecise geometry may lead to distorted textures. Similarly, since we aim to reconstruct the shape and appearance of the target objects from *a few* (typically, 2 to 4) views, it will be hard for IBR approaches to produce satisfying results with such limited quantity of input images. Furthermore, another limitation of IBR techniques is the use of *all* input images during rendering time, introducing additional memory loads for real-time applications. Another solution could be applying learning-based methods in the image (or, texture) space. Nonetheless, it would be difficult to control the parameterization of predicted and ground-truth meshes to be exactly same such that perceptual [87] or adversarial [88] losses can be used. Therefore, we propose to exploit our 3D-CFCN architecture with additional RGB images to learn the appearance information from data.

Similar to the TSDF-based volumetric representation, with a set of RGB-D images, the color volume [89] is defined as :

$$\mathrm{C}(\mathbf{p}) = \frac{\sum \omega_{\mathrm{i}}^{\mathrm{c}}(\mathbf{p})\mathrm{C}_{\mathrm{i}}(\mathbf{p})}{\sum \omega_{\mathrm{i}}^{\mathrm{c}}(\mathbf{p})}, \tag{10}$$

where $\mathrm{C}_{\mathrm{i}}(\mathbf{p})$, $\omega_i^c(\mathbf{p})$ are the color and weight of the corresponding pixel in the i-th input RGB image, respectively.

Instead of predicting color information after obtaining the predicted geometric shapes, we propose to reconstruct both geometry and texture information *jointly*. Considering that the spatial distribution of the color volume is highly correlated with the TSDF volume, the end-to-end joint learning task is easily tractable and also more efficient compared with the two-step learning scheme. Another motivation for joint learning is to maintain the consistency between the geometry and the appearance since they share the same low-level features and octree structures. To this end, we introduce an additional color loss term $\mathcal{L}_c$ to $\mathcal{L}$ for color prediction, where

$$\mathcal{L}(\theta; V^l, G^l) = \mathcal{L}_{\ell_1} + \lambda_1\mathcal{L}_{bce} + \lambda_2\mathcal{L}_s + \lambda_3\mathcal{L}_c \tag{11}$$

$$\mathcal{L}_c = \frac{1}{\sum \tau(\mathbf{p})} \sum \tau(\mathbf{p}) \|\mathrm{C}(\mathbf{p}) - \mathrm{C}_{\mathrm{gt}}(\mathbf{p})\| \tag{12}$$

$$\tau(\mathbf{p}) = \left\{ \begin{array}{ll} 1, & if -1 < \mathrm{V}(\mathbf{p}) < 1 \\ 0, & otherwise \end{array} \right. . \tag{13}$$

Here $\tau(\mathbf{p})$ represents whether a voxel $\mathbf{p}$ is close enough to the surface. In this way, only voxels close to the shell of the target shape are considered in the color loss function. This constraint reduces computation on redundant voxels and avoids the network generating extra occupancy volume to distinguish black color from empty voxels. We find this constraint improves both accuracy and speed during training in our experiments.

Thus, the input of the network is a 3D volume with $n+3$ channels (i.e., n-bin Tsdf-hist encoding plus 3 channels for RGB values), and the output is the predicted TSDF and RGB values. To produce a colored model, we then extract the color of each vertex on the reconstructed mesh from the corresponding color volume. After parameterizing the mesh via the LSCM algorithm [90], we bake the vertex color to a texture map.

Texture maps often contain much more fine details than their corresponding geometric shapes. Although the designed network is able to recover the low- to middle-frequency components of the target models' appearance, it remains hard to reconstruct high-frequency visual details, which may lead to blurring effects. Since input RGB images already provide some appearance information about the target objects under certain view points, we can blend input RGB images with predicted color maps for improved fidelity. One straightforward approach is to directly map input images onto the predicted 3D shapes with normal weighted blending, and override the predicted texture. However, it may lead to visible seams due to the color inconsistency between predicted and directly observed regions. We address this issue by applying Poisson Blending algorithm [91] to blend the predicted texture and direct mapping texture.

4 Experiments

We have evaluated our 3D-CFCN architecture on both ModelNet40 and Human10 and compared different aspects of our approach with other state-of-the-art alternatives.

4.1 High-quality Shape Reconstruction

In our experiments, we train the 3D-CFCN separately on each dataset for 20 epochs (12 for stage 1, 8 for two stages jointly), using the ADAM optimizer [92] with 0.0001 learning rate, which takes $\approx$ 80 hours to converge. Balancing weights in Eq. 5 are set to: $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$. During inference, it takes ≈ 3.5 s on average to perform a forward pass through both stages on a NVIDIA GeForce GTX 1080 Ti. The Marching Cubes algorithm [93] is used to extract surfaces from output TSDFs. Figs. 1, 3, and 4 illustrate the high-quality reconstruction results achieved with our 3D-CFCN architecture.

In Fig. 3 we show a variety of test cases from both Human10 and ModelNet40 dataset. All the input TSDF-Hists were fused using depth maps from 2 viewpoints, and the same TSDF truncation threshold were applied. Despite the presence of substantial noise and missing data, our approach was able to reduce the noise and infer the missing structures, producing clean and detailed reconstructions. Comparing the second and the third column, for Human10 models, stage 2 of our 3D-CFCN significantly improved the quality by bringing more geometric details to output meshes; on the other hand, 128^3 voxel resolution suffices to ModelNet40, thus stage 2 does not show significant improves in these cases.

4.1.1 *Auxiliary Visual Hull Information*

In practice, most RGB-D sensors can capture synchronized depth and color images, which opens up the possibility of getting auxiliary segmentation masks [94]. Given the segmentation masks from each view, a corresponding visual hull [95], which is essentially an occupancy volume, can be extracted. Visual hulls provide additional information about the distribution of both *occupied* and empty spaces, which is crucial to reliable shape completion. We thus evaluated the performance of our 3D-CFCN when visual hull information is available. Towards this goal, we added corresponding visual hull input branches to both two stages,

TABLE 1: Quantitative comparisons of shape reconstruction techniques. Relative Hausdorff RMS distance with respect to the diagonals of bounding boxes are measured against the ground-truth triangle meshes. All baseline methods use input data fused from 2 views.

Method	Human10	ModelNet40
PSR	0.0092	0.0620
Hull-constrained PSR	0.0086	0.0388
3D-EPN	0.0263	0.0178
OctNetFusion	0.0040	0.0035
3D-CFCN (2 views)	0.0035	0.0032
3D-CFCN (2 views w/ visual hull)	0.0031	0.0019
3D-CFCN (4 views)	0.0021	0.0010

which are concatenated with intermediate features after two 3^3 convolutional layers. Table 1 reports the average Hausdorff RMS distance between predicted and ground-truth 3D meshes, showing that using additional visual hull volumes as input brought a performance gain around 11%. Both TSDF-Hists and visual hull volumes in this experiment were generated using 2 viewpoints. Note that we scaled the models in Human10 to fit into a 3^3 bounding box. To further prove the effectiveness of the proposed visual hull branch, we also compared the reconstruction accuracy of our approach and hull-constrained PSR [57] (Table 1, second row), showing the significant advantage of the presented 3D-CFCN architecture.

4.1.2 Number of Viewpoints

Here we evaluated the impact of the completeness of input TSDF-Hists, i.e., the number of viewpoints used for fusing raw depth scans, on reconstruction quality. We trained and tested the 3D-CFCN architecture using TSDF-Hists fused from 2 and 4 viewpoints, listing the results in Table 1. As expected, using more depth scans led to increasing accuracy of output meshes, since input TSDF-Hists were less incomplete.

4.1.3 Robustness to Calibration and Tracking Error

Apart from sensor noise, calibration and tracking error is another major factor that can crack scanned models. To evaluate the robustness of the proposed approach to calibration and tracking error, we added random perturbations (from 2.5% to 10%) to ground-truth camera poses, generated corresponding test samples, and predicted the reconstruction results using 3D-CFCN. As shown in Fig. 5, although the network has not been trained on samples with calibration error, it can still infers geometric structures reasonably.

4.2 Texture Reconstruction

In this part, we only consider experiments on Human10 dataset, since ModelNet40 does not provide ground-truth textures. λ_c is set to 0.5 in all experiments. Rest experiment settings are the same as the shape reconstruction (see Sec. 4.1).

We compared our texture reconstruction approach with naive projective texture mapping, Screened Poisson Surface Reconstruction [2], PatchMatch [96], and a learning-based image completion method, i.e. Iizuka et al. [97]. Since the approach in Iizuka et al. [97] is built upon Generative Adversarial Networks (GAN) and hence designed to inpaint natural images, we apply it on re-rendered images, instead of directly on parameterized texture atlas images. More specifically, we set up 25 virtual cameras around the reconstructed subject (with projective texture applied), render the partial color images and occlusion masks at each camera view, employ Iizuka et al. [97] (trained on the Places2 dataset) to complete rendered images, and then recompute the final texture map from inpainted color images. For PatchMatch [96], we also follow the above scheme to get completed textures, as we find it performs better than operating directly on parameterized UV space in our experiments. Compared with Screened Poisson Surface Reconstruction, the proposed 3D-CFCN architecture is able to reconstruct geometric shapes with better completeness and accuracy, reducing distortion and ghosting artifacts on projected textures (see Fig. 6(b,f)). While PatchMatch [96] is able to fill in missing regions with reasonable colors, it sometimes fails to inpaint large holes (e.g., see the head and face regions of the last two subjects in Fig. 6(d)); also, it may have problems to make predictions around fine structures (see the regions around the hands and arms of the first two subjects in Fig. 6(d)). Iizuka et al. [97] fails to infer large missing regions as well, and it may produce ghosting effects (e.g., see Fig. 6(e), third, fifth, and seventh row). In comparison, our approach is able to complete missing areas with more consistent content.

We further evaluated the effect of texture blending in Fig. 7. We can observe some texture seams along the boundary between visible and occluded regions due to inconsistent resolutions and illumination conditions across different viewpoints (see Fig. 7(a)). These corresponding regions present more natural appearance after blending (see Fig. 7(b)).

In addition, Fig. 8 illustrates the effect of the auxiliary RGB input and color prediction branch on geometric reconstruction. Comparing shape reconstruction results with and without RGB inputs, it can be observed that color information helps to enhance the reconstruction of fine geometric details (e.g. see the facial regions, bottom opening and embossed letters on clothes in Fig. 8), as RGB channels provide complementary high frequency signals while the color reconstruction branch helps to improve training and generalization from the multi-task learning perspective.

4.3 Comparison with Existing Learning-based Approaches

Fig. 4 and Table 1 compare our 3D-CFCN architecture with three learning-based state-of-the-art alternatives for 3D shape reconstruction, i.e., OctNetFusion [3], 3D-EPN [54], and OGN [29], as well as the widely used geometric method Poisson surface reconstruction (PSR) [2].

OctNetFusion. Similar to our approach, OctNetFusion adopts OctNet as the building block and learns to denoise and complete input TSDFs in a multi-stage manner. However, each stage in OctNetFusion is designed to take an upsampled TSDF and refine it globally (i.e., each stage needs to process *all* the octants in the grid-octree at the current resolution), making it infeasible to reconstruct 3D shape at higher

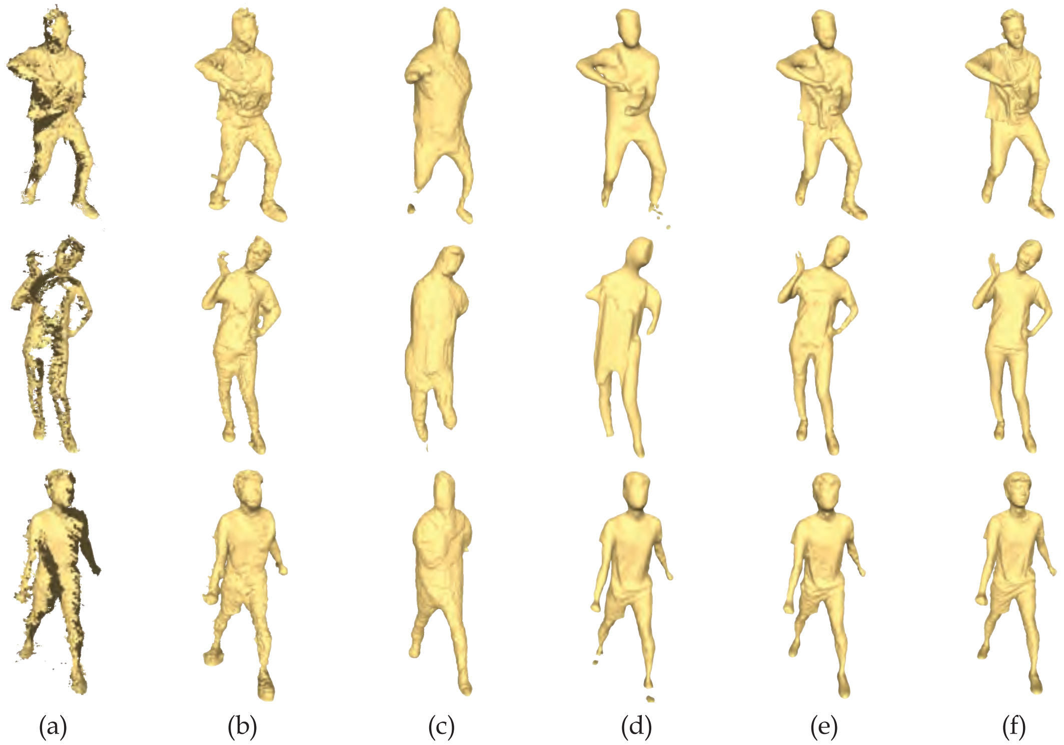

Fig. 4: Comparison of our shape reconstruction results with other state-of-the-art alternatives. (a): Input scans. (b): PSR [2]. (c): 3D-EPN [54]. (d): OctNetFusion [3]. (e): Ours. (f): Ground-truth references. Note the bulging artifacts on PSR's results.

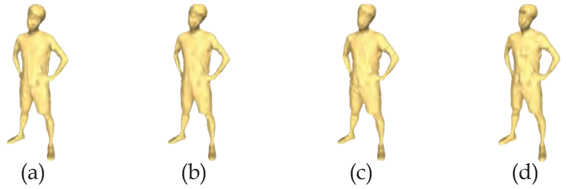

Fig. 5: Reconstruction results of the proposed 3D-CFCN under different levels of calibration error. (a): No error. (b): 2.5%. (c): 5%. (d): 10%.

resolutions, as learning at higher resolutions (e.g., 512^3) not only increases the memory cost at input and output layers, but also requires deeper network structures, which further challenges the limited computational resource. Fig. 4 and Table 1 summarize the comparison of our reconstruction results at 512^3 voxel resolution with OctNetFusion's results at 256^3.

3D-EPN. Without using octree-based data structures, 3D-EPN employs a hybrid approach, which first completes the input model at a low resolution (32^3) via a 3D CNN and then uses voxels from similar high-resolution models in the database to produce output distance volumes at 128^3 voxel resolution. However, as shown in Fig. 4, while being able to infer the overall shape of input models, this approach fails to recover fine geometric details due to the limited resolution.

OGN. As another relevant work to our 3D-CFCN architecture, OGN is a octree-based convolutional decoder. Although scales well to high resolution outputs, it remains challenging to recover accurate and detailed geometry information from encoded shape features via only deconvolution operations. To compare our approach with OGN, we trained the proposed 3D-CFCN on Human10 dataset to predict occupancy volumes, extracted 32^3 intermediate feature from the stage-1 3D FCN of our architecture, and used these feature maps to train an OGN. Fig. 9 compares the occupancy maps decoded by OGN with the corresponding occupancy volumes predicted by the proposed 3D-CFCN (both at 512^3 resolution), showing that our method performs significantly better than OGN with respect to fidelity and accuracy. Table 2 summarizes the results of quantitative comparisons between OGN and 3D-CFCN.

TABLE 2: Quantitative comparisons of occupancy reconstruction. We use average Hamming distance to measure the reconstruction accuracy, since the outputs are binary occupancy volumes. Our approach performs better than OGN on both dataset.

Method	Human10	ModelNet40
OGN (2 views)	0.1862	0.1391
OGN (4 views)	0.0844	0.0439
3D-CFCN (2 views)	0.0084	0.0019
3D-CFCN (4 views)	0.0033	0.0010

4.4 Generalization Ability

In Fig. 10, we demonstrate the reconstruction results of our approach on human scans that were not included in Human10. Per-vertex color predicted using 3D-CFCN is used in Fig. 10(c) for better evaluation of the behavior of the proposed network architecture. As shown in the figure, our approach generalize well on unseen data.

4.5 Computational Efficiency

Table 3 and Table 4 compare the runtime and memory consumption for OctNetFusion, OGN and the proposed 3D-CFCN at different output resolutions. We performed the experiment with batch size 1, and the iteration time

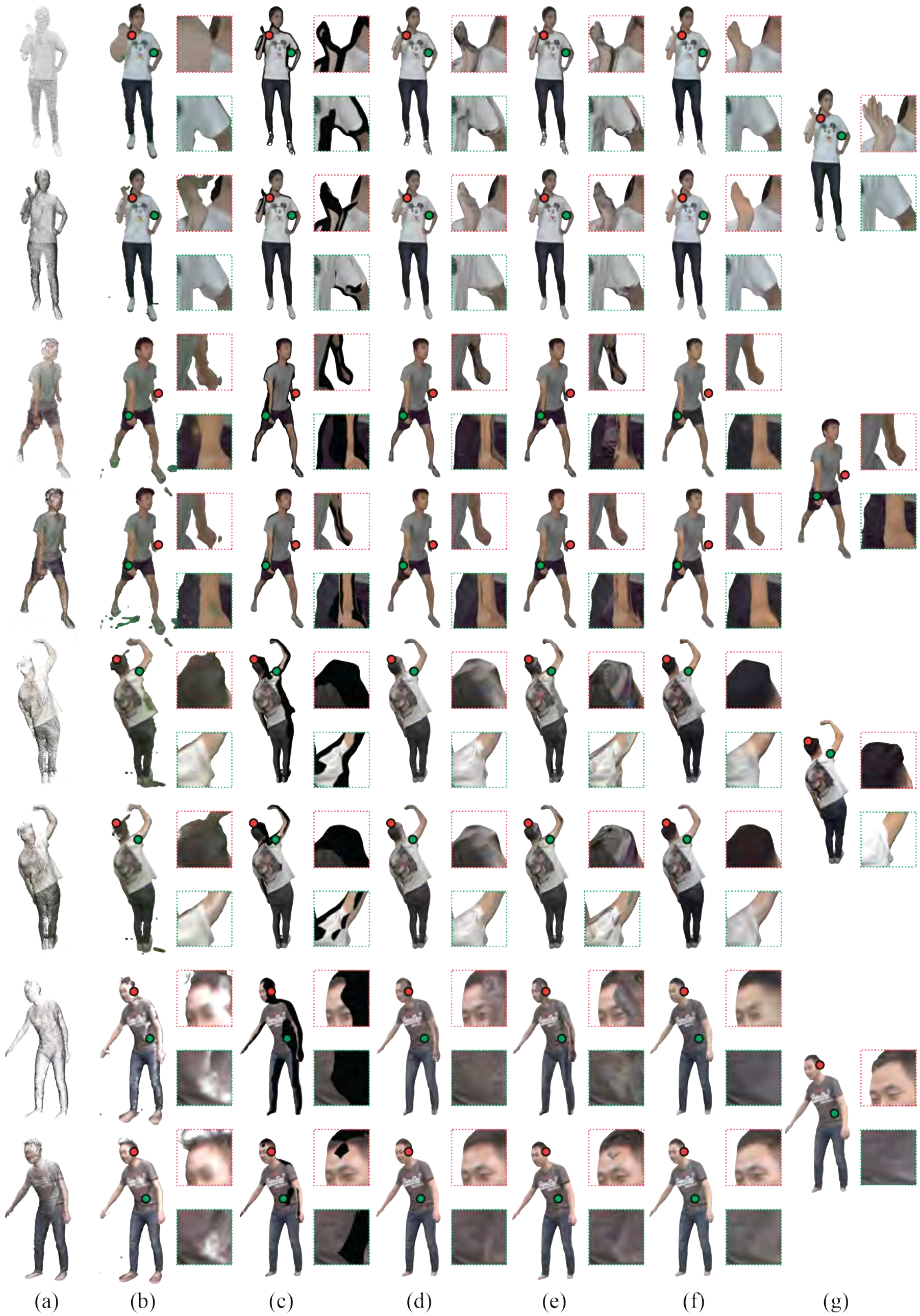

Fig. 6: Comparison of our texture reconstruction results with alternative approaches. Inset boxes highlight artifacts or details on reconstructed 3D models. (a): Input scans. (b): Screened PSR [2]. (c): Geometric shapes predicted using 3D-CFCN, textured with projective mapping. (d): Textured with projective mapping, inpainted using PatchMatch [96]. (e): Textured with projective mapping, inpainted using Iizuka et al. [97] (f): Our results. (g) Ground-truth references. The first row of each subject is fused from 2 views while the second row is fused from 4 views. Note that black regions in (c) indicate occluded areas in input views.

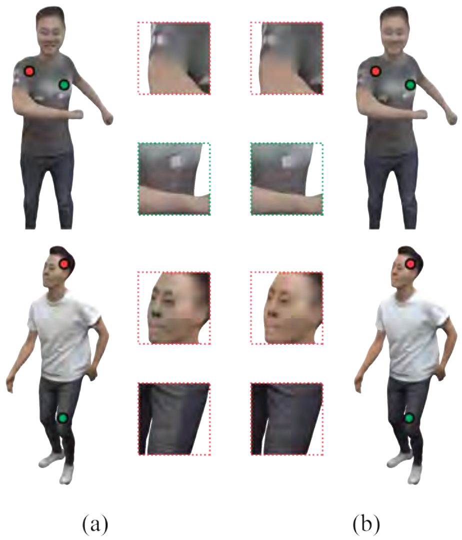

Fig. 7: Effect of texture blending. (a): Without blending. (b): With Poisson blending. Inset boxes highlight details on reconstructed models.

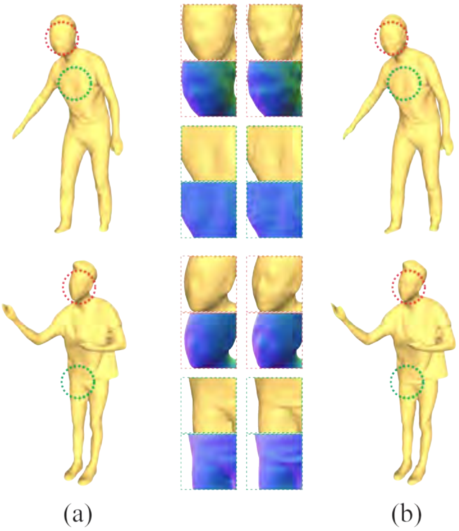

Fig. 8: Improved shape reconstruction with the auxiliary color prediction task. (a) Results without color reconstruction loss. (b) Results with color reconstruction loss. Inset boxes highlight the improvements (lower boxes visualize the normal direction).

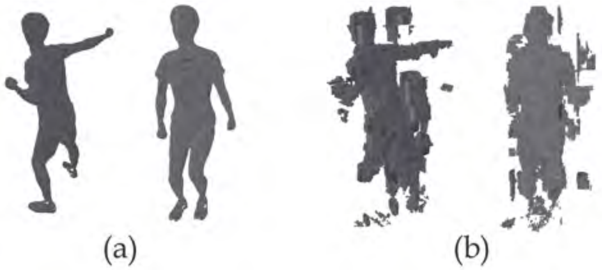

Fig. 9: Comparison with OGN. (a): Occupancy maps reconstructed by 3D-CFCN. (b): Occupancy maps decoded by OGN, using features learned by 3D-CFCN.

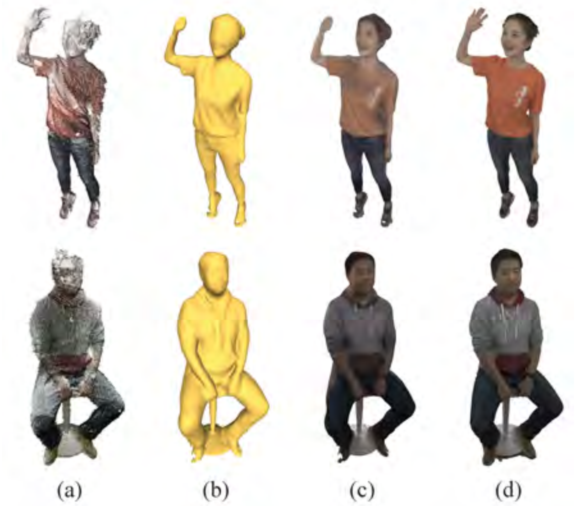

Fig. 10: Generalization ability of 3D-CFCN on unseen data. (a): Input scans (fused from 2 viewpoints). (b): Geometric shapes predicted using 3D-CFCN. (c): Jointly reconstructed shape and color using 3D-CFCN. (d): Ground-truth references.

TABLE 3: Iteration time (s) at training stage. 3D-CFCN (C) predicts both TSDF and color volumes while 3D-CFCN (O) predicts occupancy maps.

Method	128^3	256^3	512^3	1024^3
OctNetFusion	0.42	0.99	n/a	n/a
3D-CFCN	0.63	1.31	3.14	5.81
OGN	0.09	0.20	1.32	7.03
3D-CFCN (O)	0.41	0.81	1.45	2.72
3D-CFCN (C)	1.02	1.63	4.98	8.10

considers both forward and backward passes. For a fair comparison, when comparing with OGN, we trained our 3D-CFCN to predict occupancy maps (i.e., 3D-CFCN (O)). Note OctNetFusion networks for 512^3 or higher resolutions cannot be fitted into a single GPU. While both 3D-CFCN and OGN scale well for high resolutions, our approach performs much faster forward and backward passes than OGN at the highest resolution. Although training 3D-CFCN for joint shape and color prediction (i.e., 3D-CFCN (C)) introduces noticeable extra cost, the growth of the overall memory footprint and computation time of our method keeps sublinear.

5 Conclusion and Discussion

We have presented a cascaded 3D convolutional network architecture for efficient and high-fidelity shape and texture reconstruction at high resolutions. Our approach refines the volumetric representations of partial and noisy input models in a progressive and adaptive manner, which substantially simplifies the learning task and reduces computational cost. Experimental results demonstrate that the proposed method can produce high-quality reconstructions with accurate geometric details and visually plausible textures. We also believe that extending the proposed approach to reconstructing dynamic sequences is a promising direction.

Limitations. One limitation of the proposed hybrid textured shape reconstruction approach is that although it

TABLE 4: Memory comsuption (GB) at training stage. 3D-CFCN (C) predicts both TSDF and color volumes while 3D-CFCN (O) predicts occupancy maps.

Method	128^3	256^3	512^3	1024^3
OctNetFusion	0.72	1.58	n/a	n/a
3D-CFCN	2.05	2.16	3.59	6.48
OGN	0.39	0.45	0.75	2.74
3D-CFCN (O)	1.26	1.63	2.28	3.49
3D-CFCN (C)	2.22	2.76	4.43	7.89

can produce overall high-fidelity reconstructions, in large occluded regions, inferred textures at lower resolution may still cause noticeable blurring artifacts (see Fig. 7(b), first row). To reduce the blurring effect, we could design the network architecture to predict geometric shapes and textures at different spatial resolutions (e.g., appending more cascades for color refinement). Also, predicting color gradients instead of values could help to reduce the solution space and thus improve color reconstruction. Besides, we do not set constraints when blending textures across seams, which may lead to visible texture seams in some cases.

Acknowledgments

This work was supported by the Joint NSFC-DFG Research Program (project number 61761136018), the Natural Science Foundation of China (Project Number 61521002), and the China Postdoctoral Science Foundation.

References

[1] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Trans. Graph.*, vol. 36, no. 3, pp. 24:1–24:18, May 2017.

[2] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, Jul. 2013.

[3] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "Octnetfusion: Learning depth fusion from data," in *Proceedings of the International Conference on 3D Vision*, 2017.

[4] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Oct 2011, pp. 127–136.

[5] S. Choi, Q. Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5556–5565.

[6] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense rgb-d images," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov 2011, pp. 719–722.

[7] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3748–3754.

[8] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[9] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 500–516.

[10] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1996, pp. 303–312.

[11] A. C. Oeztireli, G. Guennebaud, and M. Gross, "Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression," *Computer Graphics Forum*, 2009.

[12] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz, "Occluding contours for multi-view stereo," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 4002–4009.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[14] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[15] C.-H. Shen, H. Fu, K. Chen, and S.-M. Hu, "Structure recovery by part assembly," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 180:1–180:11, Nov. 2012.

[16] K. Chen, Y. Lai, Y.-X. Wu, R. R. Martin, and S.-M. Hu, "Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information," *ACM Transactions on Graphics*, vol. 33, no. 6, 2014.

[17] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 628–644.

[18] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *European Conference on Computer Vision*. Springer, 2016, pp. 322–337.

[19] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Advances in Neural Information Processing Systems*, 2016, pp. 1696–1704.

[20] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in Neural Information Processing Systems*, 2016, pp. 82–90.

[21] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[22] A. Sinha, A. Unmesh, Q. Huang, and K. Ramani, "Surfnet: Generating 3d shape surfaces using deep residual networks," in *Proc. CVPR*, 2017.

[23] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3d neural network for multiview stereopsis," *arXiv preprint arXiv:1708.01749*, 2017.

[24] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow, "Structured prediction of unobserved voxels from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5431–5440.

[25] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 190–198.

[26] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, "3d object reconstruction from a single depth view with adversarial learning," *arXiv preprint arXiv:1708.07969*, 2017.

[27] A. Sharma, O. Grau, and M. Fritz, "Vconv-dae: Deep volumetric shape learning without object labels," in *European Conference on Computer Vision*. Springer, 2016, pp. 236–250.

[28] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3, 2017.

[29] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[30] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 72:1–72:11, Jul. 2017.

[31] Y.-P. Cao, Z.-N. Liu, Z.-F. Kuang, L. Kobbelt, and S.-M. Hu, "Learning to reconstruct high-quality 3d shapes with cascaded fully convolutional networks," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 626–643.

[32] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 301–329.

[33] K. Chen, Y.-K. Lai, and S.-M. Hu, "3d indoor scene modeling from rgb-d data: a survey," *Computational Visual Media*, vol. 1, no. 4, pp. 267–278, 2015.

[34] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun, "Voronoi-based variational reconstruction of unoriented point sets," in *Symposium on Geometry processing*, vol. 7, 2007, pp. 39–48.

[35] F. Calakli and G. Taubin, "Ssd: Smooth signed distance surface reconstruction," in *Computer Graphics Forum*, vol. 30, no. 7. Wiley Online Library, 2011, pp. 1993–2002.

[36] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 67–76.

[37] I. Macedo, J. P. Gois, and L. Velho, "Hermite radial basis functions implicits," in *Computer Graphics Forum*, vol. 30, no. 1. Wiley Online Library, 2011, pp. 27–42.

[38] G. Guennebaud and M. Gross, "Algebraic point set surfaces," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 23.

[39] A. C. Öztireli, G. Guennebaud, and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 493–501.

[40] S. Fuhrmann and M. Goesele, "Fusion of depth maps with multiple scales," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 148.

[41] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust tv-l 1 range image integration," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[42] R. Schnabel, P. Degener, and R. Klein, "Completion and reconstruction with primitive shapes," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 503–512.

[43] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1261–1268.

[44] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo, "An interactive approach to semantic modeling of indoor scenes with an rgbd camera," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 136, 2012.

[45] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 77–85.

[46] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 424–432.

[47] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, "Shape inpainting using 3d generative adversarial network and recurrent convolutional networks," *arXiv preprint arXiv:1711.06375*, 2017.

[48] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Computer Vision and Pattern Regognition (CVPR)*, 2018.

[49] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, "Bodynet: Volumetric inference of 3d human body shapes," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 20–38.

[50] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll, "Video based reconstruction of 3d people models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[51] Z. Huang, T. Li, W. Chen, Y. Zhao, J. Xing, C. LeGendre, L. Luo, C. Ma, and H. Li, "Deep volumetric video from very sparse multi-view performance capture," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 351–369.

[52] A. Gilbert, M. Volino, J. Collomosse, and A. Hilton, "Volumetric performance capture from minimal camera viewpoints," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 591–607.

[53] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[54] A. Dai, C. R. Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 3, 2017.

[55] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *IEEE International Conference on Computer Vision (ICCV)*, October 2017.

[56] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3d object reconstruction," *arXiv preprint arXiv:1704.00710*, 2017.

[57] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 69:1–69:13, Jul. 2015.

[58] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi, "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 741–754.

[59] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or, "Seamless montage for texturing models," *Computer Graphics Forum*, vol. 29, no. 2, pp. 479–486, 2010.

[60] K. Zhou, X. Wang, Y. Tong, M. Desbrun, B. Guo, and H.-Y. Shum, "Texturemontage," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1148–1155, Jul. 2005.

[61] Q.-Y. Zhou and V. Koltun, "Color map optimization for 3d reconstruction with consumer depth cameras," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 155:1–155:10, Jul. 2014.

[62] R. Du, M. Chuang, W. Chang, H. Hoppe, and A. Varshney, "Montage4d: Interactive seamless fusion of multiview video textures," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM, 2018, pp. 5:1–5:11.

[63] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1996, pp. 43–54.

[64] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 2001, pp. 425–432.

[65] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1996, pp. 11–20.

[66] P. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," in *Rendering Techniques '98*, G. Drettakis and N. Max, Eds. Vienna: Springer Vienna, 1998, pp. 105–116.

[67] M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. De Aguiar, N. Ahmed, C. Theobalt, and A. Sellent, "Floating Textures," *Computer Graphics Forum*, 2008.

[68] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 30:1–30:12, Jul. 2013.

[69] P. Hedman, T. Ritschel, G. Drettakis, and G. Brostow, "Scalable inside-out image-based rendering," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 231:1–231:11, Nov. 2016.

[70] Fitzgibbon, Wexler, and Zisserman, "Image-based rendering using image-based priors," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 1176–1183 vol.2.

[71] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *European Conference on Computer Vision*. Springer, 2016, pp. 286–301.

[72] E. Y. D. C. A. C. B. Eunbyung Park, Jimei Yang, "Transformation-grounded image generation network for novel 3d view synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 702–711.

[73] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," in *SIGGRAPH Asia 2018*. ACM, 2018, pp. 257:1–257:15.

[74] R. Martin-Brualla, R. Pandey, S. Yang, P. Pidlypenskyi, J. Taylor, J. Valentin, S. Khamis, P. Davidson, A. Tkach, P. Lincoln, A. Kowdle, C. Rhemann, D. B. Goldman, C. Keskin, S. Seitz, S. Izadi, and S. Fanello, "Lookingood: Enhancing performance capture with

real-time neural re-rendering," in *SIGGRAPH Asia 2018*. ACM, 2018, pp. 255:1–255:14.

[75] J. Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Nießner, "IGNOR: image-guided neural object rendering," *CoRR*, vol. abs/1811.10720, 2018.

[76] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3d reconstruction in dynamic scenes using point-based fusion," in *3D Vision-3DV 2013, 2013 International Conference on*. IEEE, 2013, pp. 1–8.

[77] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph," *Robotics: Science and Systems*, 2015.

[78] D. Gallup, M. Pollefeys, and J.-M. Frahm, "3d reconstruction using an n-layer heightmap," in *Joint Pattern Recognition Symposium*. Springer, 2010, pp. 1–10.

[79] M. Meilland and A. I. Comport, "On unifying key-frame and voxel-based dense visual slam at large scales," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3677–3683.

[80] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.

[81] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[82] Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, and P.-A. Heng, "3d deeply supervised network for automated segmentation of volumetric medical images," *Medical image analysis*, vol. 41, pp. 40–54, 2017.

[83] S. R. Fanello, J. Valentin, C. Rhemann, A. Kowdle, V. Tankovich, P. Davidson, and S. Izadi, "Ultrastereo: Efficient learning-based matching for active stereo systems," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 6535–6544.

[84] Q. Chen and V. Koltun, "Fast mrf optimization with application to depth reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3914–3921.

[85] P. McIlroy, S. Izadi, and A. Fitzgibbon, "Kinectrack: 3d pose estimation using a projected dense dot pattern," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 6, pp. 839–851, 2014.

[86] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *BMVC*, January 2011.

[87] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 694–711.

[88] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[89] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 169:1–169:11, Nov. 2013.

[90] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 362–371, Jul. 2002.

[91] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[93] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.

[94] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[95] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International journal of computer vision*, vol. 38, no. 3, pp. 199–218, 2000.

[96] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *ACM SIGGRAPH 2009 Papers*, ser. SIGGRAPH '09. New York, NY, USA: ACM, 2009, pp. 24:1–24:11.

[97] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 107:1–107:14, Jul. 2017.

Zheng-Ning Liu received his bachelor's degree in computer science from Tsinghua University in 2017. He is currently a PhD candidate in the Department of Computer Science and Technology, Tsinghua University. His research interests include 3D reconstruction, geometric modeling and processing.

Yan-Pei Cao received his bachelor's degree and Ph.D. degree in computer science from Tsinghua University in 2013 and 2018, respectively. He is currently a postdoctoral researcher in the Department of Computer Science and Technology, Tsinghua University. His research interests include geometric modeling and processing, 3D reconstruction, and 3D computer vision.

Zheng-Fei Kuang Zheng-Fei Kuang is currently an undergraduate student in the Department of Science and Technology at Tsinghua University. He is expected to receive his bachelor's degree in Computer Science in 2019. His research interests include computer vision and computer graphics.

Leif Kobbelt is a full professor and the head of the Computer Graphics Group at the RWTH Aachen University, Germany. His research interests include all areas of Computer Graphics and Geometry Processing with a focus on multiresolution and freeform modeling as well as the efficient handling of polygonal mesh data. He was a senior researcher at the Max-Planck- Institute for Computer Sciences in Saarbrücken, Germany from 1999 to 2000 and received his Habilitation degree from the University of Erlangen, Germany where he worked from 1996 to 1999. In 1995/96 he spent a postdoctorate year at the University of Wisconsin, Madison. He received his masters degree in (1992) and Ph.D. in (1994) from the University of Karlsruhe, Germany. Over the last few years he has authored many research papers in top journals and conferences and served on several program committees.

Shi-Min Hu is currently a professor in the department of Computer Science and Technology, Tsinghua University, Beijing, China. He received the PhD degree from Zhejiang University in 1996. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He has published more than 100 papers in journals and refereed conferences. He is Editor-in-Chief of Computational Visual Media, and on editorial boards of several journals, including IEEE Transactions on Visualization and Computer Graphics, Computer Aided Design and Computer & Graphics. He is a senior member of IEEE and ACM.