Reconstructing 3D Shapes From Multiple Sketches Using Direct Shape Optimization

Zhizhong Han, Baorui Ma, Yu-Shen Liu, *Member, IEEE*, and Matthias Zwicker, *Member, IEEE*

***Abstract*— 3D shape reconstruction from multiple hand-drawn sketches is an intriguing way to 3D shape modeling. Currently, state-of-the-art methods employ neural networks to learn a mapping from multiple sketches from arbitrary view angles to a 3D voxel grid. Because of the cubic complexity of 3D voxel grids, however, neural networks are hard to train and limited to low resolution reconstructions, which leads to a lack of geometric detail and low accuracy. To resolve this issue, we propose to reconstruct 3D shapes from multiple sketches using direct shape optimization (DSO), which does not involve deep learning models for direct voxel-based 3D shape generation. Specifically, we first leverage a conditional generative adversarial network (CGAN) to translate each sketch into an attenuance image that captures the predicted geometry from a given viewpoint. Then, DSO minimizes a project-and-compare loss to reconstruct the 3D shape such that it matches the predicted attenuance images from the view angles of all input sketches. Based on this, we further propose a progressive update approach to handle inconsistencies among a few hand-drawn sketches for the same 3D shape. Our experimental results show that our method significantly outperforms the state-of-the-art methods under widely used benchmarks and produces intuitive results in an interactive application.**

***Index Terms*— 3D shape reconstruction, sketches, multiple angles, voxels, optimization.**

I. Introduction

Reconstructing 3D shapes from 2D hand-drawn sketches is an intriguing way to interactive 3D shape modeling. As a simple and intuitive representation [1]–[10], sketching allows users to describe the 3D shapes that they have in mind by just drawing lines in an interactive manner. Due to the lack of shading or texture information, however, 2D sketches are extremely ambiguous as a representation of 3D shapes. This makes it hard to learn plausible mappings from sketches to 3D shapes, which significantly limits the reconstruction accuracy.

To reduce this ambiguity, current methods [11], [12] allow users to draw sketches from multiple viewpoints for more detailed input, and they resort to deep learning models to obtain the complex mapping from sketches to 3D shapes. By allowing users to progressively draw sketches from different angles, deep learning models can incrementally update the 3D shape. Current techniques use 3D voxel grids as shape representations because they can represent arbitrary topologies and they enable the implementation of convolutional networks in a straightforward manner. However, 3D voxel grids lead to a cubic complexity for memory and computation time, which significantly limits this strategy to low resolution voxel representations. Therefore, it is still a research challenge to accurately reconstruct high resolution 3D shapes with more detailed geometry.

To overcome this challenge, we propose a method to directly optimize the reconstructed 3D shapes based on the understanding of sketches by leveraging a 2D deep learning model. Instead of directly generating 3D voxel representations using deep learning models, we only leverage the deep learning model to understand 2D sketches. Then, we perform direct shape optimization (DSO) to reconstruct a 3D shape without the involvement of a deep learning model, which avoids the computational burden of high resolution voxel grids. Specifically, we employ a conditional generative adversarial network (CGAN) [13] to understand a sketch by translating it into an image that reveals the shape geometry from a certain view angle. Regarding the predicted images from multiple view angles as targets, our DSO employs a project-and-compare loss to push the reconstructed shape to match the target image from the same view angle. In addition, DSO enables the progressive refinement of partial geometry as the user creates multiple sketches interactively. In summary, our contributions are as follows:

i) We propose a novel method to enable high resolution 3D shape reconstruction from multiple sketches. Our direct shape optimization technique uses a project-and-compare loss to avoid the direct involvement of deep learning models in 3D shape reconstruction, and it circumvents the computational burden of voxel-based deep learning models.
ii) We introduce a progressive update approach for reconstruction from multiple hand-drawn sketches. Our method robustly handles inconsistencies among multiple sketches and enables users to reconstruct high fidelity shapes using just a few sketches.

Manuscript received February 4, 2020; revised July 1, 2020; accepted August 5, 2020. Date of publication September 1, 2020; date of current version September 4, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB0505400 and in part by the NSF under Award 1813583. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Charith Abhayaratne. *(Corresponding author: Yu-Shen Liu.)*

Zhizhong Han is with the School of Software, BNRist, Tsinghua University, Beijing 100084, China, and also with the Department of Computer Science, University of Maryland, College Park, MD 20737 USA (e-mail: h312h@umd.edu).

Baorui Ma and Yu-Shen Liu are with the School of Software, BNRist, Tsinghua University, Beijing 100084, China (e-mail: liuyushen@tsinghua.edu.cn; mbr18@mails.tsinghua.edu.cn).

Matthias Zwicker is with the Department of Computer Science, University of Maryland, College Park, MD 20737 USA (e-mail: zwicker@cs.umd.edu).

Digital Object Identifier 10.1109/TIP.2020.3018865

iii) Our experimental results demonstrate that our method can reconstruct 3D shapes with more detailed and accurate geometry in higher resolution than the state-of-the-art.

II. Related Work

3D shape modeling from sketches has been drawing research attention in 3D computer vision for decades. Along with the development of deep learning models [14]–[16], there has been great progress in understanding the 3D world in different applications, such as 3D shape feature learning [17]–[25], shape completion [26]–[31], shape reconstruction [32]–[39], shape captioning [40], [41], and scene understanding [42], [43], where 3D shapes are represented by different 3D shape representations including triangle meshes [44]–[48], multiple views [22], [23], point clouds [24], [25], [31], [38], [49]–[52], and voxel grids [39], [53]. Here, we focus on work considering 3D shape reconstruction based on sketches. We classify the related works into two categories in terms of the involvement of deep neural networks.

A. Geometric Reasoning Based Methods

This category contains geometric reasoning based methods that first derive geometric constraints from the local shape features represented by lines or joints in sketches, and then leverage the derived constraints in a shape optimization framework [1]–[4]. The derived constraints may include convexity, symmetry, orthogonality, parallelism, discontinuity, surface orientation and surface developability [3], [54]–[56]. However, these methods are restricted to specific sketch drawings such as polyhedral scaffolds [57], curvature-aligned cross-sections [58], curvature flow lines [59], or cartoon isophotes [60], which are also required to be drawn by users as accurately as possible for better constraint derivation. These issues make it hard for untrained users to obtain successful results in an interactive sketch drawing system. In contrast, our method does not require users to provide accurate sketches to parse local geometry, and just reconstructs 3D shapes based on the global understanding of multiple sketches achieved using a CGAN.

B. Deep Learning Based Methods

Thanks to the powerful learning ability of deep neural networks, deep learning based methods are able to learn a mapping from 2D sketches to 3D shapes in an end-to-end manner. For example, encoder-decoder networks are a widely used deep learning architecture to achieve this. The encoder aims to extract features of the 2D sketches, and the decoder transforms these features into 3D shapes. Here, the reconstructed 3D shape could be represented as a voxel grid [11], [61] or template-based meshes [62], where the difference between the shape predicted by the deep learning model and the ground truth is minimized to train the network. The method [11] allows users to provide multiple sketches that are employed to iteratively refine the predicted voxel grid. However, this method is limited to low resolution 3D grids, due to the high computation cost in deep neural networks. Although this method was subsequently extended to predict normal information, which helps to smooth the surfaces of the reconstructed shapes [12], it still suffers from the low resolution problem. Recently, a system [63] was proposed to infer a set of parametric surfaces that realize the smooth drawing in 3D from a single bitmap sketch. However, this system does not have the ability of understanding multiple sketches drawn for the same 3D shape.

To resolve this issue, some methods represent 3D shapes as multiple depth images [64], [65] or normal images [66] that are predicted by deep learning models from multiple input sketches. Each predicted depth or normal image shows the characteristics of the 3D shape observed from a specific view angle, where all view angles cover the whole 3D shape. Although these methods avoid the involvement of voxel grids in deep learning models, each depth or normal image only represents a partial geometry of the 3D shape. Hence, this strategy additionally requires a multi-view fusion procedure to carefully register point clouds from different views. While our method also adopts a multi-view strategy to circumvent deep learning models based on voxel grids, our direct shape optimization allows us to directly predict 3D shapes without registration in a fusion process.

III. Method Overview

As shown in Fig. 1, our method aims to reconstruct a 3D shape $\boldsymbol{M}$ using multiple sketches $\boldsymbol{s}_i$ drawn from arbitrary view angles υ_i, where $i \in [1, V]$ and V is the number of view angles which can be chosen arbitrarily. Our method first employs a CGAN to understand each sketch $\boldsymbol{s}_i$ by predicting its geometry from the corresponding view angle. In our approach, the view-dependent geometry is represented as an attenuance image $\boldsymbol{d}'_i$, where each pixel contains the attenuance (that is, one minus the transmittance) along a ray through the shape. Then, we reconstruct the shape $\boldsymbol{M}$ according to the multiple images $\boldsymbol{d}'_i$ predicted by CGAN using direct shape optimization (DSO), which is cast as a tomographic reconstruction problem. Our DSO technique minimizes a project-and-compare loss to progressively refine the partial geometry of $\boldsymbol{M}$ such that its projections from view angles υ_i match the corresponding attenuance maps $\boldsymbol{d}'_i$.

IV. Recovering Partial Geometry From Sketches

A. Conditional GAN

Our method employs a CGAN to process and interpret the geometry of each sketch $\boldsymbol{s}_i$. A CGAN is a generative network formed by a GAN [67] with additional conditions. We aim to leverage the powerful generative ability of CGAN to predict geometric details from sketches. The CGAN employs a generator G to learn a mapping from sketches $\boldsymbol{s}_i$ and a random noise vector z to real attenuance images $\boldsymbol{d}_i$, such that $G : \{\boldsymbol{s}_i, z\} \rightarrow \boldsymbol{d}_i$ (we discuss the real attenuance images generation in Section V). The generator G is trained to predict a $\boldsymbol{d}'_i$ that cannot be distinguished from real images $\boldsymbol{d}_i$ by a discriminator D. At the same time, the discriminator D is trained to learn to distinguish predicted images from real ones,

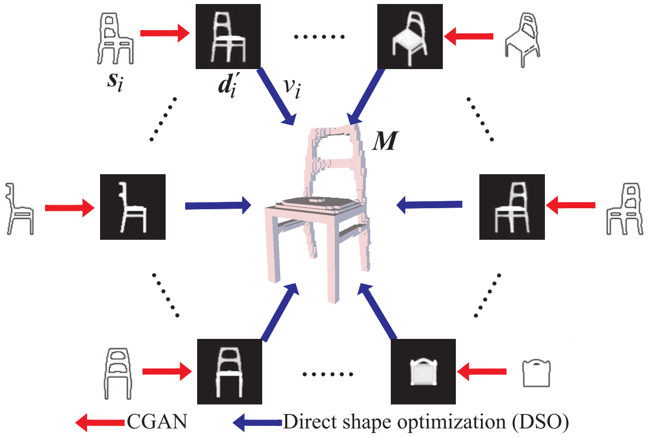


Fig. 1. Overview of our method. We first employ a CGAN (red arrows) to understand the view-dependent geometry of each sketch, which can be drawn from an arbitrary view angle. Specifically, the CGAN translates the sketch into an attenuance image, where each pixel represents the attenuance (one minus transmittance) of a ray that passes through the 3D shape. Then, we refine the reconstructed 3D shape using our direct shape optimization technique, which solves a tomographic reconstruction problem (blue arrows).

where D and G are trained adversarially, as illustrated in Fig 2.

B. Objective Function

We train the CGAN using the standard objective function, as defined below,

$$L_{CGAN}(G, D) = \mathbb{E}_{s_i,d_i}[\log(D(s_i, d_i))] + \mathbb{E}_{s_i,z}[\log(1 - D(s_i, d_i'))], \quad (1)$$

where $d_i' = G(s_i, z)$. D tries to distinguish predicted images d_i' from real images d_i by maximizing the objective function, while G tries to fool the discriminator by minimizing the objective function.

To prevent d_i' from being far away from the ground truth d_i, we also employ a L1 distance loss to further constrain the generator G, similar as pix2pix networks [68]. The L1 loss encourages less blurring in terms of pixel values as defined below,

$$L_{Pixel}(G) = \mathbb{E}_{s_i,d_i,z}[||d_i - d_i'||_1]. \quad (2)$$

Finally, we aim to train an optimal generator G^* to satisfy the objective function below,

$$G^* = \arg\min_G \max_D L_{CGAN}(G, D) + \lambda L_{Pixel}(G), \quad (3)$$

where λ is a weight to balance the two losses, and we set $\lambda = 100$ in our experiments.

C. Architecture

For the architecture of our CGAN, we employ a network similar to pix2pix [68] to implement the generator G and the discriminator D. Specifically, the generator G is a "U-Net" [69] which is an encoder-decoder network with skip connections between symmetric layers in the encoder and decoder stacks. To avoid the generation of blurry images, the discriminator D classifies whether each $N \times N$ patch on a predicted image is real or not, where the averaged responses across the whole image are regarded as the output. In addition, both generator and discriminator use the modules of the form convolution-BatchNorm-ReLu [70]. In each module, all convolutions employ 4×4 spatial filters with a stride of 2. All leaky ReLUs in the encoder are with a slope of 0.2, while ReLUs in the decoder are not leaky.

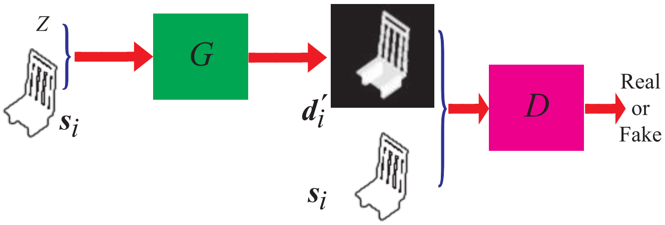


Fig. 2. Illustration of CGAN. The generator G is trained adversarially by competing with the discriminator D, which aims to enable G to generate d_i' that are indistinguishable from the real attenuance images d_i.

V. Direct Shape Optimization

Direct shape optimization (DSO) aims to refine the reconstructed 3D shape M using the predicted attenuance images d_i' from their corresponding view angles υ_i. Intuitively, DSO tries to minimize a project-and-compare loss to push M such that its projection from each view angle υ_i matches the predicted image d_i'.

A. Shape Projection

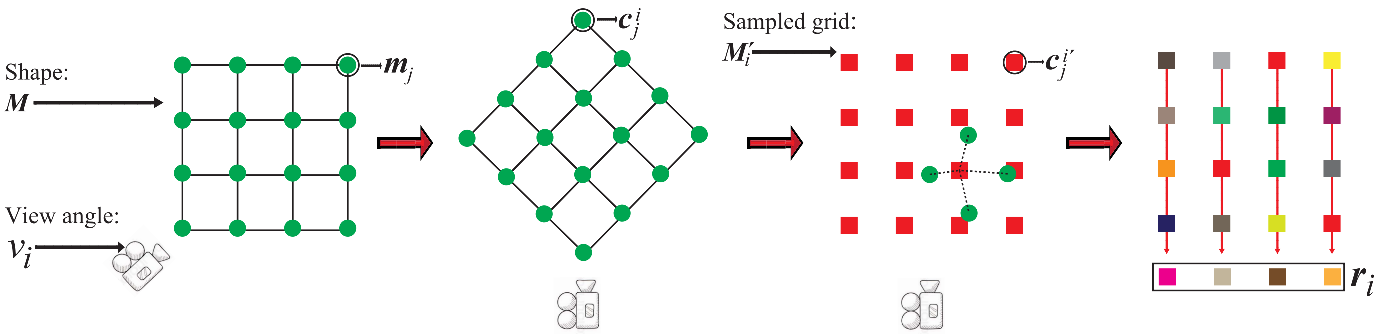
1) Shape Representation: In DSO, M is regarded as a voxel grid with a resolution of $R \times R \times R$, where each voxel m is a variable in a value range of $[0, 1]$ that represents the voxel's occupancy probability. In our projection model, we interpret the occupancy probabilities as absorption coefficients, and render images as attenuance maps where each pixel contains the attenuance (one minus transmittance) along a ray through the shape. Attenuance images can fully preserve the geometry of 3D shapes in the orthogonal projection introduced later, and more importantly, the generation of attenuance images is differentiable, which enables the DSO in the reconstruction.

To produce the appearance of M from view angle υ_i, we render M by projecting all variables m in the $R \times R \times R$ voxel grid to a 2D plane. We then leverage the rendered view to evaluate how similar it is to the predicted image d_i' from the trained CGAN.

For efficiency, we employ orthogonal projection rather than perspective projection to map the shape M onto a 2D plane. We represent the reconstructed 3D shape M in an object-centered coordinate system, and we use P to denote the entire projection procedure which produces the projected image r_i from shape M and view angle υ_i below,

$$r_i = P(M, \upsilon_i). \quad (4)$$

2) Coordinate Transformation: To project the voxel grid to a 2D plane, we first transform the coordinate of each voxel m from the object-centered coordinate system to a viewer-centered coordinate system, as shown from Fig. 3 (a) to



(a) 3D shape and view angle (b) Coordinate transformation (c) Resampling and interpolation (d) Orthogonal projection

Fig. 3. The shape projection process, for clarity illustrated from 2D to 1D. Starting from a 2D shape $\boldsymbol{M}$ as a voxel grid (the center of each voxel is represented as a green node) and a view angle v_i in (a), we first transform the voxels from the object-centered into the viewer-centered coordinate system in (b), then uniformly resample the volume into a new grid $\boldsymbol{M}'_i$ and get the values of sampled nodes (red squares) by interpolating the nearest 4 voxels (we only draw 4 voxel centers for clarity) in (c), and finally obtain the projected image $\boldsymbol{r}_i$ of $\boldsymbol{M}$ via orthogonal projection and computing attenuance values (Eq. (7)) using the sampled grid in (d).

Fig. 3 (b). We denote the coordinate of the center of the j-th voxel m_j as $\boldsymbol{m}_j$, where $j \in [1, R^3]$. According to the camera pose rotation matrix $\boldsymbol{T}_i$ from view angle v_i, the coordinate $\boldsymbol{c}^i_j$ of m_j in this viewer-centered coordinate system is

$$\boldsymbol{c}^i_j = \boldsymbol{T}_i \boldsymbol{m}_j. \tag{5}$$

Note that users can draw sketches while viewing a 3D shape as reference, so the view angle v_i of each sketch is captured when the user is drawing the sketch.

3) Resampling and Interpolation: To perform orthogonal projection, we resample the volume into a new $R \times R \times R$ grid $\boldsymbol{M}'_i$ that is aligned with the i-th viewer-centered coordinate system, as shown in Fig. 3 (c). The resampled grid $\boldsymbol{M}'_i$ bridges the optimization target $\boldsymbol{M}$ and the final projected image $\boldsymbol{r}_i$. Here, we denote each resampled voxel node as m'_j to differentiate it from the voxel m_j in shape $\boldsymbol{M}$, and denote the coordinate of m'_j as $\boldsymbol{c}^{i\,'}_j$ in the i-th viewer-centered coordinate system. We employ trilinear interpolation to get the value of each resampled voxel m'_j from the nearest 8 m_j in $\boldsymbol{M}$ as follows,

$$m'_j = \sum_{m_k \in \{m_j | \boldsymbol{c}^i_j \in 8\text{-NN}(\boldsymbol{c}^{i\,'}_j), m_j \in \boldsymbol{M}\}} w_k m_k, \tag{6}$$

where w_k is the weight which is a normalized ratio in terms of distance between the locations of $\boldsymbol{c}^{i\,'}_j$ and $\boldsymbol{c}^i_j$, as indicated by the dashed line in Fig. 3 (c).

4) Shape Projection: We finally obtain the projected image $\boldsymbol{r}_i$ by computing the attenuance along all rays orthogonal to the 2D projection plane. This is achieved by simply summing up all resampled voxels along the axis orthogonal to the 2D projection plane, as shown in Fig. 3 (d), and then exponentiating,

$$\boldsymbol{r}_i(u, v) = 1 - e^{-\sum_z \boldsymbol{M}'_i(u,v,z)}, \tag{7}$$

where u and v are the coordinates along the horizontal and vertical image axes, respectively.

B. Optimization

To push shape $\boldsymbol{M}$ to look the same as the attenuance image $\boldsymbol{d}'_i$ predicted by the CGAN from each v_i of V view angles, we optimize each voxel value m_j in $\boldsymbol{M}$ to minimize the L2 distance between $\boldsymbol{d}'_i$ and the image $\boldsymbol{r}_i$ projected from shape $\boldsymbol{M}$ in Eq. (7), as defined below,

$$L = \sum_{i \in [1,V]} ||\boldsymbol{r}_i - \boldsymbol{d}'_i||_2. \tag{8}$$

Eq. (8) defines a loss in a project and compares the process, hence we call it project-and-compare loss. By minimizing this project-and-compare loss, we can refine the partial geometry of shape $\boldsymbol{M}$. Hence, our objective function in DSO is defined as,

$$\boldsymbol{M}^* = \arg \min_{m_j \in \boldsymbol{M}} L. \tag{9}$$

which is essentially a tomographic reconstruction problem. We optimize Eq. (9) in an iterative manner to update each voxel m_j of shape $\boldsymbol{M}$ using the gradient obtained by the chain rule connecting Eq. (6), Eq. (7), and Eq. (8),

$$\frac{\partial L}{m_j} = \sum_{i \in [1,V]} \frac{\partial L}{\partial \boldsymbol{r}_i} \frac{\partial \boldsymbol{r}_i}{\partial m'_j} \frac{\partial m'_j}{\partial m_j}. \tag{10}$$

C. Pseudo Code

To make the direct shape optimization more clear, we further provide pseudo code below.

VI. Generating Training Sketches

We generate sketches for training our CGAN using synthetic 3D shapes from a 3D shape classification benchmark, as introduced in Sec. VIII-A later. For a 3D shape represented as a voxel grid, we generate V sketches from V view angles that are placed on a sphere around the 3D shape, where $V = 25$ in our experiments. In our setup, the x-z-plane is the ground plane, the y-axis points upwards, and the object is centered at the origin of the sphere. The V view angles are uniformly distributed on three rings around the y-axis, and the camera is

Algorithm 1 Direct Shape Optimization

```
Data: current reconstructed 3D shape $\boldsymbol{M}$, attenuance
      images $\boldsymbol{d}'_i$ predicted by CGAN, view angles $v_i$,
      where $i \in [1, V]$
Result: Reconstructed 3D shape $\boldsymbol{M}$
while $i \leq V$ do
    1. Coordinate transformation of $\boldsymbol{M}$ to the camera
    coordinate system specified by $v_i$ using Eq. (5);
    2. Calculate $\boldsymbol{M}_i$ by resampling and interpolation in
    the camera coordinate system using Eq. (6);
    3. Project $\boldsymbol{M}_i$ into an image $\boldsymbol{r}_i$ using Eq. (7);
    4. Calculate the error between $\boldsymbol{M}_i$ and the predicted
    attenuance image $\boldsymbol{d}'_i$ using $||\boldsymbol{r}_i - \boldsymbol{d}'_i||_2$;
    5. Optimize $\boldsymbol{M}$ by minimizing the error of
    $||\boldsymbol{r}_i - \boldsymbol{d}'_i||_2$;
```

always oriented towards the center of the sphere. We employ azimuth and elevation angles θ and ϕ to indicate the location of the cameras, which are placed at $\theta = \{0, 0.25\pi, 1.75\pi\}$, and $\phi = \{0, 0.25\pi, 0.5\pi, \ldots, 2.0\pi\}$. Beside these, the last camera is located at $\theta = 0.5\pi$ and $\phi = 0$, which is right above the 3D shape. The camera system is shown in Fig. 5 (a), where each node represents a camera location.

From each view angle v_i of V view angles shown in Fig. 4 (a), we first repeat the processes described in Sec. V-A to project the ground truth 3D shape to obtain the real attenuance image $\boldsymbol{d}_i$, as shown in Fig. 4 (b). Then, we extract the edge information from $\boldsymbol{d}_i$, which we will use as the training sketch $\boldsymbol{s}_i$, using the Moore-Neighbor tracing algorithm [71], as shown in Fig. 4 (c).

VII. Shape Reconstruction With Hand-Drawn Sketches

In a practical shape editing application, users should be able to reconstruct a 3D shape using just a few hand-drawn sketches, and the system should accurately and intuitively leverage the sketches to determine the desired shape. To achieve this goal, we propose a progressive update approach for 3D shape reconstruction, which also robustly handles inconsistencies among multiple hand-drawn sketches from different view angles. Our progressive update approach includes a retrieval process and an iterative drawing process, as illustrated in Fig. 5.

A. Initialization by Sketch Retrieval

Specifically, to reduce the required number of hand-drawn sketches, we first retrieve a 3D shape from a dataset by finding the shape that best matches the first sketch (black star with an index of 1). For this sketch-based shape retrieval step, we represent each 3D shape in the dataset by $V = 25$ sketches in the camera system in Fig. 5 (a). We then find the shape that contains the sketch that is closest to the user provided input sketch in the HOG [72] feature space.

Next, we use the retrieved 3D shape to initialize the voxel grid $\boldsymbol{M}$ that is to be reconstructed, and then, update it using DSO based on the first hand-drawn sketch in Fig. 5 (b). We employ the trained CGAN to predict the corresponding attenuance image of the first sketch, and we use this image for DSO. Subsequently, we provide the updated 3D shape $\boldsymbol{M}$ to users as a reference, so that users can draw the second sketch from a new view angle (black star with an index of 2) and further update the shape $\boldsymbol{M}$ using two predicted attenuance images in Fig. 5 (c). Users can iteratively draw additional sketches from new view angles, and use DSO to update the shape each time until they are satisfied with the result. For illustration, Fig. 5 (d) indicates the addition of a third sketch (black star with an index of 3).

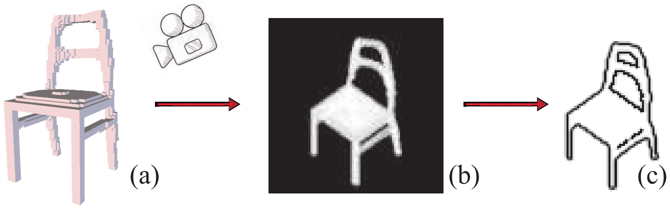


Fig. 4. Illustration of training sketch generation. Starting from a 3D shape and a view angle in (a), we first project the shape to an attenuance image in (b), and then extract a sketch from the attenuance image in (c) using the Moore-Neighbor tracing algorithm.

B. The Challenge of Sketch Consistency

One challenge for our approach is that it is hard for users to provide multiple sketches from different view angles that are consistent with a 3D shape. For example, a user who is trying to draw a chair may draw four legs from one view, and then change the view to provide more details. However, it is hard to draw the four legs with the exact same size and shape from another view. Such inconsistencies can dramatically degenerate the optimization result after each update. To resolve this issue, we propose two strategies to alleviate the effect of inconsistent user sketches.

C. Higher Confidence to the Last Sketch

The first strategy is that we give higher confidence to the last sketch than to the previous sketches. This makes sure that the shape $\boldsymbol{M}$ is immediately updated to match the sketch that the user just drew. To implement the higher confidence on the last sketch, we introduce a weighted project-and-compare loss based on Eq. (8) and add a larger weight on the predicted attenuance image corresponding to the last sketch, as shown below,

$$L_W = \sum_{x \in [1,X]} y_x ||\boldsymbol{r}_x - \boldsymbol{d}'_x||_2, \tag{11}$$

where x is the index of sketches drawn for the same shape and X is the number of sketches that have been drawn before each update. In our experiment, we only set the weight of the last sketch to 10, such that $y_X = 10$, and keep all the weights for the previous sketches to be 1.

D. Higher Confidence to Directly Inferred Voxels

The second strategy is that we give higher confidence to voxel variables that can be directly inferred by the predicted

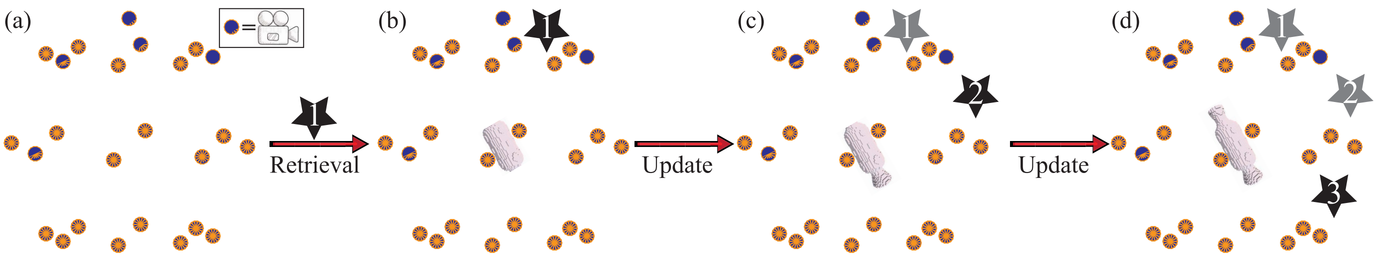


Fig. 5. Our process for updating reconstructed shapes using hand-drawn sketches: (a) The camera system we employ to generate training data, where each node represents a camera which is looking at the center of 3D space covered by the camera system. (b) According to the first sketch (black star with an index of 1) drawn by the user, we first retrieve a similar 3D shape, and update the shape according to the first sketch. (c) Repeating this process, we progressively reconstruct the shape by updating it using a second sketch. (d) Similarly, the third sketch is drawn and used to update the shape together with the two previous sketches. In each step, the last sketch (in dark) is given higher confidence than the previous ones (in light dark).

attenuance image $\boldsymbol{d}'_X$ corresponding to the last sketch. This is to achieve an intuitive and incremental editing step where we update the reconstructed shape $\boldsymbol{M}$ by revising only part of the geometry based on the last sketch, and keep the rest of the geometry unchanged. We can infer the geometry that the user wants to keep unchanged according to the predicted image $\boldsymbol{d}'_X$ of the last sketch, which we roughly split into foreground (pixels with non-zero values) and background (pixels with zero values). Intuitively, the current voxels with a value of 0 whose projections are also located in the background on the predicted image $\boldsymbol{d}'_X$ should keep the value of 0, since the user intends to keep them as background. On the other hand, the current voxels with a value of 1 whose projections are also located in the foreground on the predicted image $\boldsymbol{d}'_X$ should keep the value of 1, since the user intends to keep them as part of the 3D shape.

An example of this partial geometry update is demonstrated in Fig. 6. For example, a user wants to enlarge the current 3D shape (in blue) from a view shown in Fig. 6 (a), so the user draws a sketch (in white) on the view in Fig. 6 (b). The CGAN interprets the sketch by predicting its attenuance image in Fig. 6 (c). From the predicted image, we can infer that the user intends to keep some voxels with a value of 1 on the current shape and some voxels with a value of 0 unchanged, whose projections on this view are located in the red area shown in Fig. 6 (d).

Specifically, we denote the set of unchanged voxels as $\boldsymbol{U}$, and fix the voxel variables in $\boldsymbol{U}$ before the coming update using DSO. In addition, we just optimize the remaining voxel variables in the update to minimize Eq. (11). Based on Eq. (10), we further propose the conditional gradient for each voxel variable m_j as follows,

$$\frac{\partial L_W}{m_j} = \begin{cases} \sum\limits_{x\in[1,X]} \frac{\partial L_W}{\partial \boldsymbol{r}_x}\frac{\partial \boldsymbol{r}_x}{\partial m'_j}\frac{\partial m'_j}{\partial m_j} & m_j \in \boldsymbol{M} \cap m_j \notin \boldsymbol{U} \\ 0 & m_j \in \boldsymbol{M} \cap m_j \in \boldsymbol{U} \end{cases} \tag{12}$$

where x is the index of sketches drawn for the same shape and X is the number of sketches that have been drawn before each update.

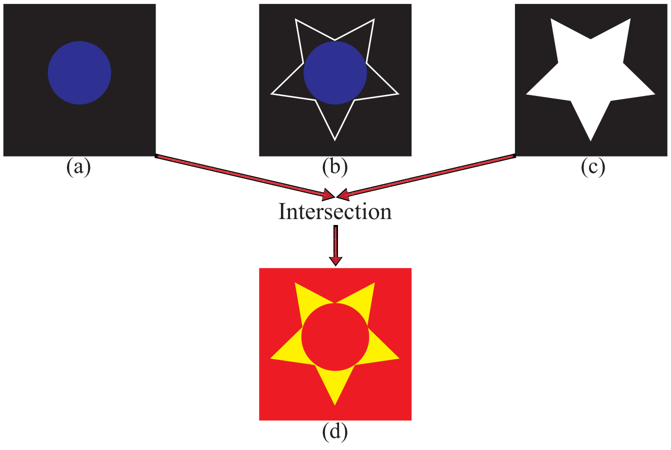


Fig. 6. Illustration of partial geometry update. For a view from the current reconstructed 3D shape (solid blue circle) in (a), a user draws a sketch (white star) in (b), CGAN predicts the attenuance image (background in black, front in white) in (c). The projections of unchanged voxel variables should be in the red area in (d).

VIII. Experiments

We evaluate our method in different experiments in this section. First, we explore the effects of key parameters on the performance. Then, we conduct a comparison with the state-of-the-art methods in automatic tests under widely used benchmarks. Finally, we demonstrate the effectiveness of our method by reconstructing 3D shapes in a practical application with freely hand-drawn sketches.

A. Dataset and Evaluation

We employ a widely used dataset including all the 400 chairs and 300 vases from [11] with the same training and test splitting, in addition to 3D shapes in the sofa, airplane, bed, monitor, table, and toilet classes from the ModelNet dataset [73], again with the standard training and test splitting. All 3D shapes are represented as voxel grids of resolution 64^3, such that $R = 64$. We employ binvox [74], [75] to get the ground truth 3D voxel grids by voxelizing 3D meshes. In addition, all our attenuance images and sketches are with a resolution of 64^2.

We evaluate the difference between the reconstructed 3D shapes and ground truth 3D shapes using the intersection-over-

TABLE I

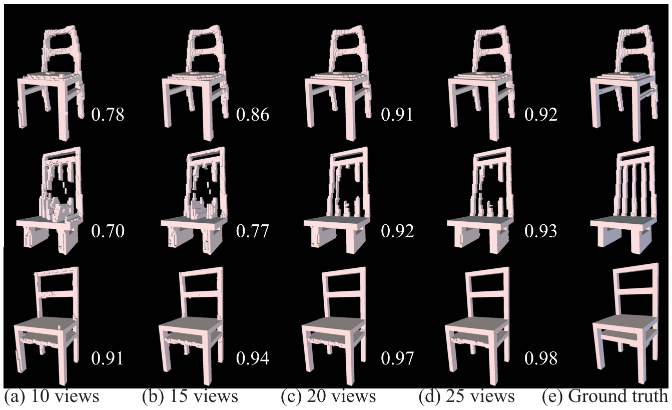
THE PATCH SIZE COMPARISON UNDER VASE CLASS IN TERMS OF IOU

Patch size N	50	100	150	200
IoU	0.65	**0.66**	0.65	0.65

TABLE II

COMPARISON OF DIFFERENT NUMBERS OF VIEWS UNDER THE VASE AND CHAIR CLASSES IN TERMS OF IOU

View number V	10	15	20	25
Vase	0.64	0.65	0.66	**0.66**
Chair	0.72	0.80	0.94	**0.95**



(a) 10 views (b) 15 views (c) 20 views (d) 25 views (e) Ground truth

Fig. 7. Chairs reconstructed by different numbers of sketches. The IoU provided for each 3D shape shows that more sketches provide more geometry details in the reconstruction.

union (IoU) metric, which is the ratio between the intersection and the union of the two voxel grids.

B. Parameters Studies

1) Patch Size N: The first parameter we explore is the patch size in the discriminator during CGAN training. We resize the predicted attenuance images $\boldsymbol{d}_i'$ or real attenuance images $\boldsymbol{d}_i$ images to 256 × 256 when training the discriminator. Then, we try several size candidates of $N \times N$ patches, such as $N = \{50, 100, 150, 200\}$, to train the CGAN for the prediction of attenuance images from sketches under the vase class. Finally, we test each trained CGAN to get the predicted attenuance images, which are further used to reconstruct 3D shapes.

As shown in Table I, the comparable results show that the CGAN is not sensitive to the patch size of the attenuance images in our method.

2) Number of Views V: We further explore the effect of the number of views V in DSO under the vase and chair classes. We uniformly sample views from the aforementioned 25 angles, and then use a subset of $V \in \{10, 15, 20, 25\}$ views. Finally, we employ DSO to reconstruct 3D shapes using these subsets with different numbers of views.

As shown in Table II, we always achieve better results with more input sketches. This is because more sketches provide more details to reconstruct 3D shapes. Visual comparison shown in Fig. 7 also justifies this point. In Fig. 8, we demonstrate some of the predicted attenuance images of the three shapes in Fig. 7 from the CGAN, which demonstrates the accurate interpretation of sketches by the CGAN.

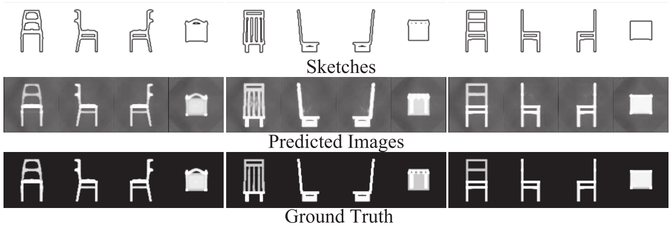


Fig. 8. Attenuance images predicted by the CGAN from sketches.

TABLE III

THE EFFECT OF BINARIZING PREDICTED ATTENUANCE IMAGES $\boldsymbol{d}_i'$ ON DSO IN TERMS OF IOU

Classes	Vase	Chair	Sofa	Airplane	Bed	Monitor	Table	Toilet
$\boldsymbol{d}_i'$	**0.71**	0.90	0.77	0.84	**0.74**	0.73	**0.55**	**0.75**
Binarized $\boldsymbol{d}_i'$	0.66	**0.95**	**0.79**	**0.87**	**0.74**	**0.74**	**0.55**	**0.75**

3) Binarizing Predicted Attenuance Images: In our experiments, we found an interesting factor that can slightly affect the accuracy of 3D shapes reconstructed by DSO. Currently, DSO is able to directly employ the predicted attenuance images $\boldsymbol{d}_i'$ from the CGAN to reconstruct 3D shapes. However, we found that we can slightly increase the accuracy of DSO further by using the binarized $\boldsymbol{d}_i'$ as input. As shown in Table III, the IoU obtained with binarized $\boldsymbol{d}_i'$ by DSO are higher than the IoU with $\boldsymbol{d}_i'$ under almost all tested shape classes. The reason for these results might be that it is still hard for the CGAN to accurately predict the attenuance information in most cases, although the CGAN can learn to distinguish between foreground (area with pixel values of 1) and background (area with pixel values of 0) very well. This inaccuracy would cause inconsistencies among predicted attenuance images for the same 3D shape, which could negatively affect the accuracy of reconstructed 3D shapes. Therefore, we employ binarized $\boldsymbol{d}_i'$ to produce results in our paper.

We show three reconstructed 3D shapes from each class in Fig. 9. These results demonstrate that DSO can reconstruct high fidelity 3D shapes from multiple predicted attenuance images based on the accurate interpretation of each one of multiple sketches. Here, we only show results with binarized $\boldsymbol{d}_i'$. As shown in Table III, the 3D shapes reconstructed from the original $\boldsymbol{d}_i'$ have a similar accuracy with comparable IoU.

According to the slight improvements with binarized $\boldsymbol{d}_i'$ in Table III, the question arises whether it is possible to further improve the results if we use binarized ground truth attenuance images $\boldsymbol{d}_i$ to train the CGAN. We conduct this experiment under the two classes in Table IV. From the comparison, however, we can see that this would lose some geometry information and lead to lower reconstruction accuracies.

C. Comparison to Space Carving

Based on the binarized attenuance images $\boldsymbol{d}_i'$ predicted from the CGAN, we could also use space carving to reconstruct a 3D shape. In the space carving process, only voxels whose projections on each view are located in the foreground (area with pixel values of 1) are considered occupied, while all the other voxels are removed. By employing the same binarized images $\boldsymbol{d}_i'$ as input, we compare DSO and space carving under

TABLE IV

THE EFFECT OF BINARIZING IMAGES $\boldsymbol{d}_i$ ON TRAINING THE CGAN IN TERMS OF IOU

Class	Vase	Chair
Train CGAN by $\boldsymbol{d}_i$	**0.66**	**0.95**
Train CGAN by binarized $\boldsymbol{d}_i$	0.64	0.92

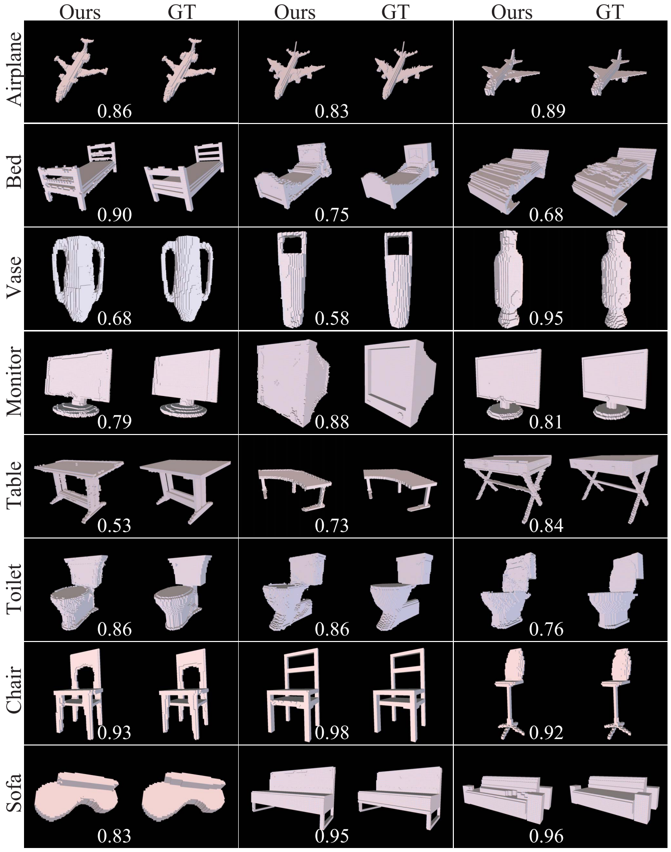


Fig. 9. Reconstructed 3D shapes under all shape classes involved in our experiments. Three high fidelity shapes are shown in each class with their IoU compared to the ground truth shapes.

four shape classes including vase, chair, sofa, and airplane in Table V, where our results significantly outperform the results obtained with space carving. Further comparison is demonstrated by error maps in multiple views in Fig. 10. In the two cases of airplane and chair, we first project the shape reconstructed by space carving and DSO into attenuance images from ten out of the $V = 25$ views by repeating the process described in Sec. V-A. Then, we get an error map between the attenuance image projected from the reconstructed shape and the ground truth shape in the same view. From the error maps, we can see that space carving produces larger error on the reconstructed shapes than DSO, especially around the boundaries of the reconstructed shapes (edges on the 2D error maps). The reason why space carving suffers from larger errors is that the predicted projected images $\boldsymbol{d}'_i$ from different views are not perfectly consistent, even after binarization. Because of its simple voting procedure, space carving is not very robust to these inconsistencies. On the other hand, DSO uses an optimization-based approach and handles these issues more successfully, leading to much higher accuracy.

TABLE V

COMPARISON TO SPACE CARVING UNDER FOUR CLASSES IN TERMS OF IOU

	Vase	Chair	Sofa	Airplane
Space carving	0.59	0.78	0.76	0.66
Ours	**0.66**	**0.95**	**0.79**	**0.87**

TABLE VI

COMPARISON TO OTHER METHODS UNDER FOUR CLASSES IN TERMS OF IOU

	Vase	Chair	Sofa	Airplane
MVDVP-S	0.4896	0.5688	0.6828	0.6918
MVDVP	0.4982	0.5732	0.6832	0.6938
MVDVP1	0.563	0.368	-	-
Ours	**0.66**	**0.946**	**0.787**	**0.872**

D. Comparison With the State-of-the-Art

We further compare our method with Multi-View Deep Volumetric Prediction (MVDVP) [11], a state-of-the-art approach for 3D reconstruction from multiple sketches. We use four shape classes as shown in Table VI. We denote the results reported in the original paper as "MVDVP1". In addition, we also train MVDVP ("MVDVP") and its single view version ("MVDVP-S") using the same training data as ours, i.e., the same sketches and voxelized 3D shapes. The comparison shows that our method significantly outperforms MVDVP. In addition, we also visually compare the three shapes reconstructed by these methods with the shapes reconstructed by our method in each one of the four shape classes in Fig. 11, which further demonstrates the effectiveness and significance of our method.

We also conduct a visual comparison with DepthFusion [65] in Fig. 12, where we also show our results as meshes by transforming voxel grids into meshes using marching cubes for a fair comparison. DepthFusion reconstructs 3D shapes by fusing multiple depth images that are predicted from sketches in fixed view angles by neural networks. We use the trained model of DepthFusion from the authors to produce the reconstructed 3D shapes in airplane and chair classes. Besides the advantage of the sketch input with arbitrary numbers and view angles, our method can also reveal more accurate structures of 3D shapes without any post processing procedure such as the fusion process.

E. Resolution of Reconstructed Shapes

As a view-based approach, another advantage of our method over voxel-based deep learning is the flexibility to adjust the resolution of reconstructed shapes R. Besides suffering from the low resolution of reconstructed shapes caused by the computational cost of voxel grids, voxel-based deep learning methods also cannot change the resolution of reconstructed shapes after training networks for a specific resolution. In contrast, our method is able to flexibly adjust the resolution of reconstructed shapes even when being trained at a specific resolution in 2D. Specifically, while we train our CGAN using sketches from shapes with $R = 64$, we can then resize the predicted attenuance images $\boldsymbol{d}'_i$ from the CGAN to higher or lower resolutions, such as $R = \{32, 128, 256\}$. Using resized $\boldsymbol{d}'_i$ as input, DSO can then produce reconstructed shapes with

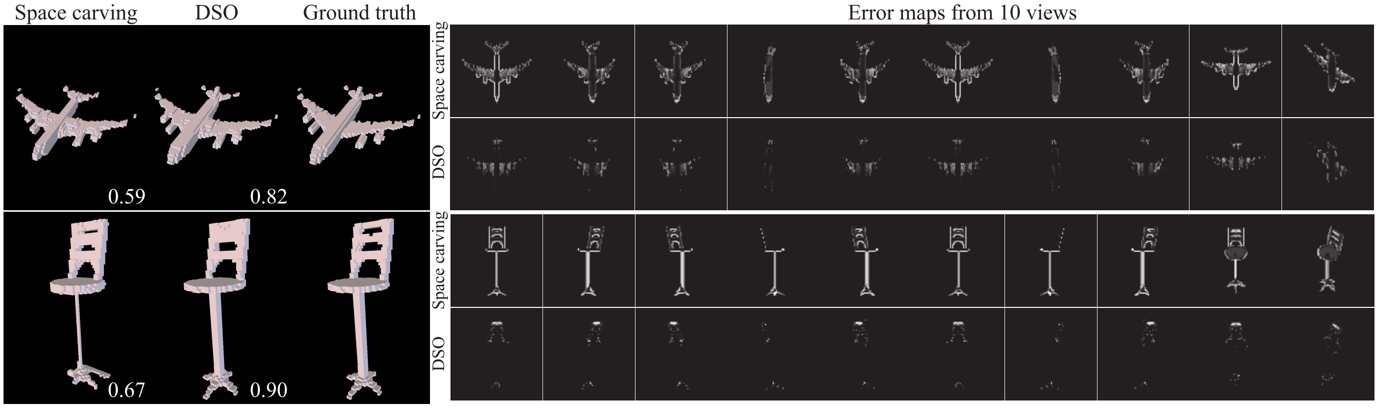


Fig. 10. Comparison between DSO and space curving. The error maps in multiple views show that DSO can achieve higher accuracies on 3D reconstruction.

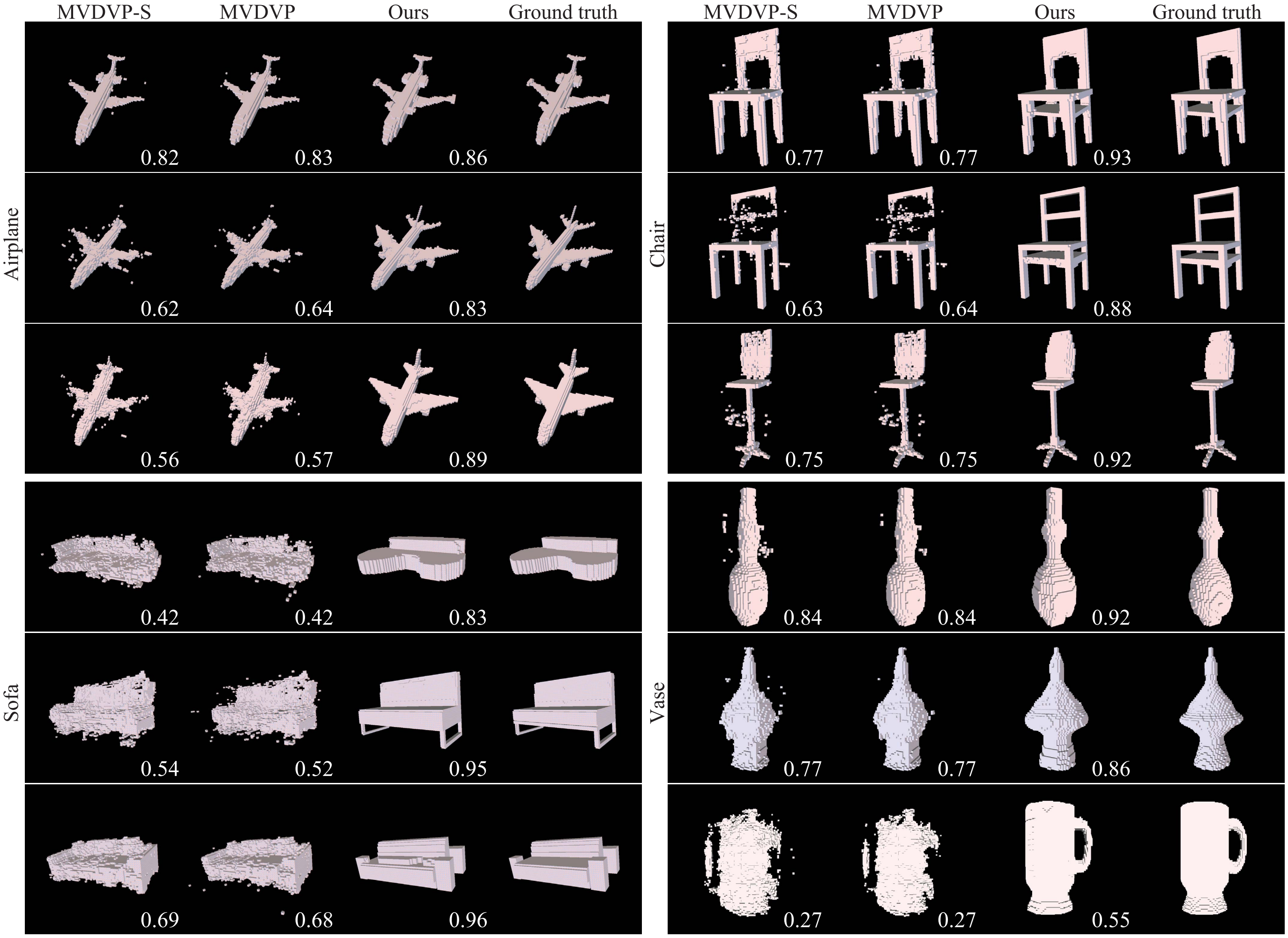


Fig. 11. Comparison between Multi-View Deep Volumetric Prediction (MVDVP) in four shape classes. MVDVP can be trained with single view (MVDVP-S) or multiple views (MVDVP) from each 3D shape. Our method outperforms both MVDVP-S and MVDVP, where the IoU is also shown below.

higher or lower resolutions. We conduct this experiment using the vase shape class, and compare the reconstructed shapes with different resolutions in Table VII. Although the IoU decreases at higher resolutions, the visual results of three shapes are still good, even at a high resolution of 256^3, as shown in Fig. 13. The reason for the lower IoU at higher resolutions is that it is harder to reconstruct voxels to accurately match the much larger number of voxels in the ground truth shapes.

F. Shape Reconstruction From Hand-Drawn Sketches

We evaluate our method using hand-drawn sketches in this experiment. We provide a user interface for users to draw

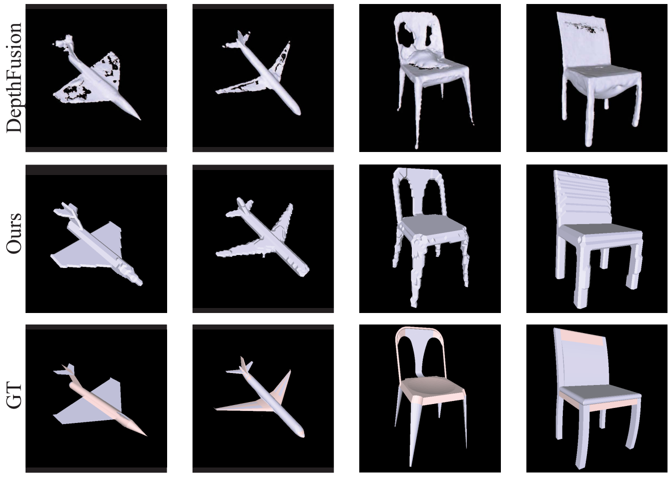


Fig. 12. Visual comparison with depth image fusion based method.

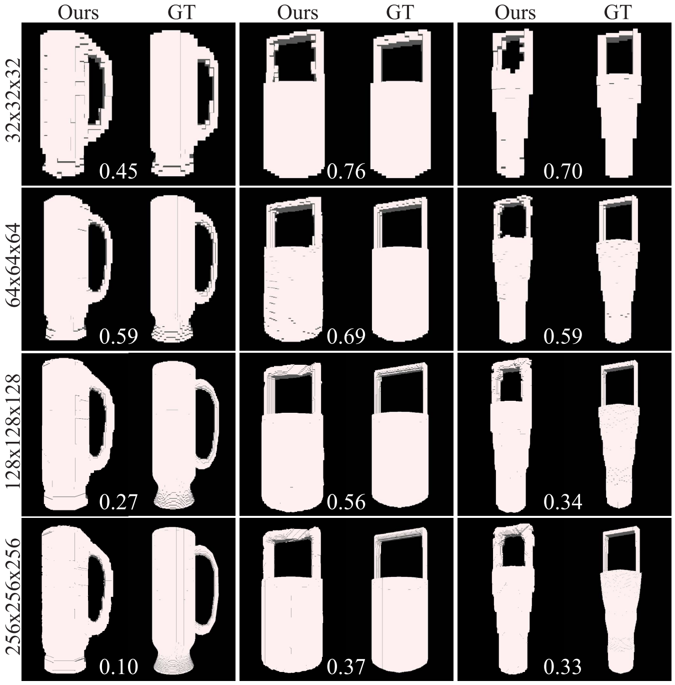


Fig. 13. Reconstructed 3D shapes from the vase class at different resolutions.

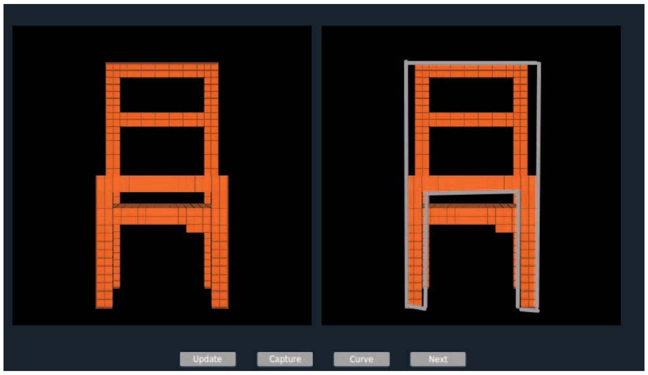


Fig. 14. The user interface to draw sketches. The currently reconstructed shape is shown in the left window, where the user can view the shape from different view angles by arbitrarily rotating the shape. The user can draw a sketch on the screen shot of the left window in the right window to update the currently reconstructed shapes from a viewpoint specified by rotating the shape in the left window. Please watch our demonstration video in the submitted multimedia file for more detail.

TABLE VII

COMPARISON OF DIFFERENT RESOLUTIONS UNDER THE VASE CLASS IN TERMS OF IOU

Resolution	32^3	64^3	128^3	265^3
IoU	0.62	0.66	0.47	0.39

sketches, as shown in Fig. 14. The user interface shows the currently reconstructed 3D shape in the left window, and the user can draw a new sketch for the next update in the right window. The user can freely rotate the current 3D shape in the left window to provide additional sketches from desired view angles.

We employ hand-drawn sketches to reconstruct some novel 3D shapes in each one of the eight shape classes involved in the experiments before, as shown in Fig. 15. To reconstruct each shape, we employ no more than three sketches, and leverage the first sketch to retrieve a 3D shape as an initialization. We also demonstrate the sketch and its corresponding predicted attenuance image for each update. The plausible results show that our method can first correctly understand the hand-drawn sketches that probably never appear in the training set, and then, robustly handle the inconsistencies among multiple sketches for the same shape.

G. Ablation Studies

Next, we conduct ablation studies to justify some important elements in our method.

1) Balance Weight and Trilinear Interpolation: Under the chair class, we first explore the effectiveness of the loss with the balance weight λ when training CGAN in Eq. (3) and the trilinear interpolation in DSO, as shown in Table VIII.

Specifically, we remove the supervision from the ground truth attenuance images by setting $\lambda = 0$. We found that the performance of 3D shape reconstruction significantly degenerates, as shown by the result of "$\lambda = 0$". Without supervision from ground truth attenuance images, the predicted attenuance images may be any images that appear real but not the one for the specific sketches, which negatively affects the reconstructed 3D shapes using DSO. Moreover, we emphasize the effect of trilinear interpolation by replacing it with nearest neighbor interpolation when resampling the 3D space in DSO. The result of "No trilinear" shows that trilinear interpolation can better optimize the reconstructed 3D shapes.

2) Project-and-Compare Loss: In this experiment, we compare the L1 and L2 distance for the project-and-compare loss in Eq 8. We employ the same predicted attenuance images for each 3D shape as the input of DSO, while using the L1 and L2 distance as the project-and-compare loss, respectively. Under the same experimental conditions, we show the results under vase and chair classes in Table IX, where the visual comparison is demonstrated in Fig. 16. This comparison indicates that L2 distance is more suitable than L1 distance for our project-and-compare loss.

3) Progressive Update Approach: We further conduct another ablation study to justify the effectiveness of each

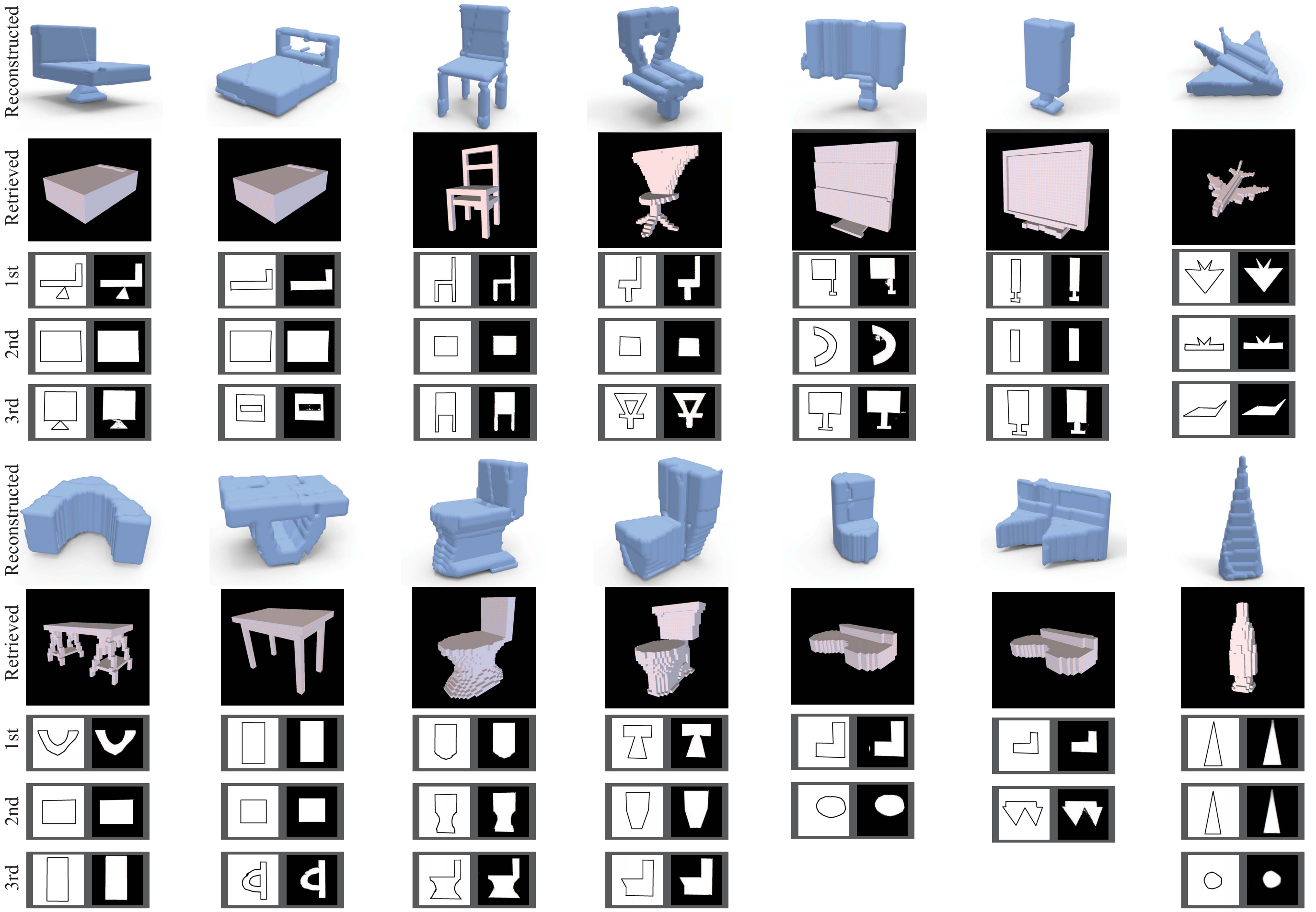


Fig. 15. Reconstructed shapes from hand-drawn sketches. Each shape is reconstructed from no more than three sketches, and we also show the corresponding predicted image for each sketch.

TABLE VIII

ABLATION STUDIES UNDER THE CHAIR CLASS IN TERMS OF IOU

	$\lambda = 0$	No trilinear	Ours
IoU	0.48	0.82	**0.95**

TABLE IX

COMPARISON OF L1 AND L2 DISTANCE IN PROJECT-AND-COMPARE LOSS IN TERMS OF IOU

Class	Vase	Chair
L1 distance	0.57	0.49
L2 distance	**0.66**	**0.95**

element in our progressive update approach for reconstruction from hand-drawn sketches. In Fig. 17 (b) to (e), we compare the reconstructed shapes obtained by the project-and-compare loss, the weighted project-and-compare loss but without conditional gradient, the project-and-compare loss with conditional gradient but without weights on views, and both weighted project-and-compare loss and conditional gradient. The degenerated results without either weights on views or conditional gradient indicate that both of them are effective in our progressive update approach to produce plausible shapes.

H. Further Analysis

1) DSO Loss: We first visualize the optimization process of the DSO by plotting the loss in Eq. (8). Fig. 18 shows that the loss converges rapidly, and DSO effectively optimizes the reconstructed 3D shape by minimizing the distance between the attenuance images rendered from the reconstructed 3D shape and the ones predicted from the trained CGAN. While we show the loss over 100 epoches, the plot reveals that no more than 50 epoches are adequate to obtain a plausible 3D shape.

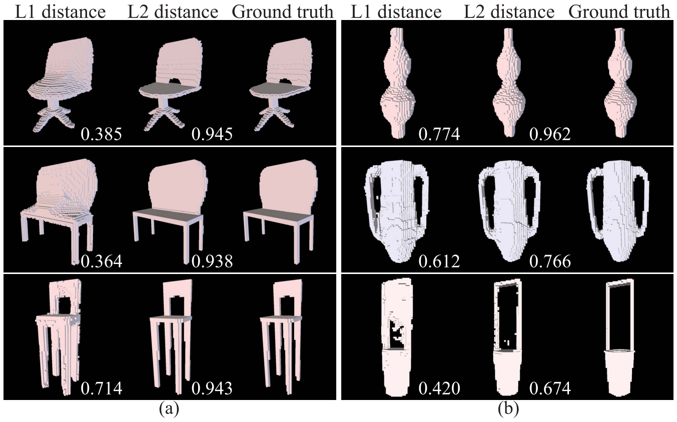


Fig. 16. Comparison between L1 and L2 distance in project-and-compare loss under the chair class in (a) and the vase class in (b).

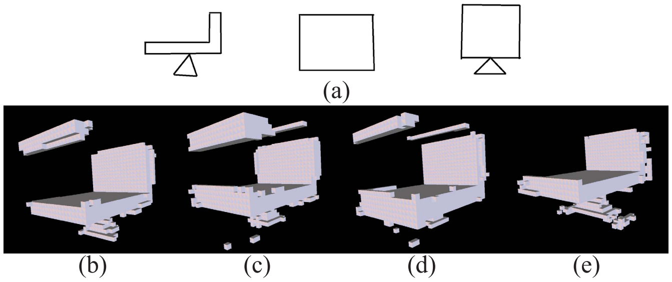


Fig. 17. Justification of our progressive update approach by reconstructed shapes in a resolution of 32^3. (a) The input sketches. (b) Result obtained by project-and-compare loss only. (c) Result with weighted project-and-compare loss but without conditional gradients. (d) Result with conditional gradients but without weighted project-and-compare loss. (e) Result obtained by our full progressive update approach.

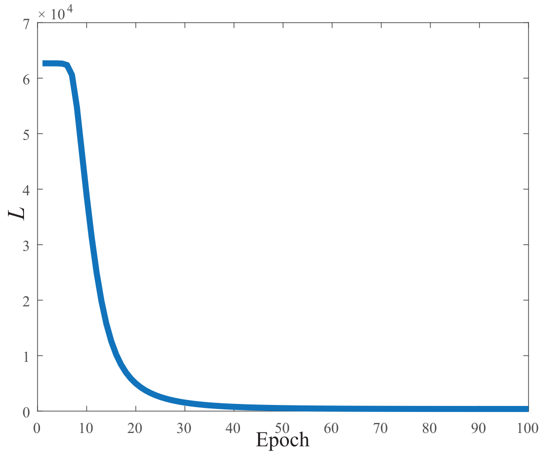


Fig. 18. Visualization of the loss in Eq. (8) in the optimization process of the DSO.

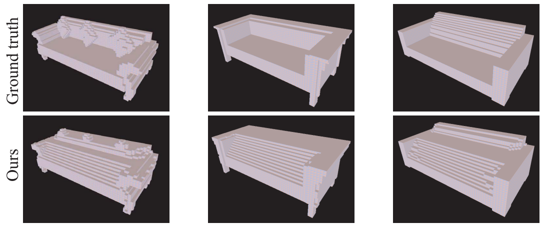


Fig. 19. Some failure cases of our method. Since the CGAN can not precisely predict the attenuance information from sketches, we binarize the predicted attenuance images for the reconstruction, which cannot reveal the detailed structures of sofas.

2) Computational Complexity: We additionally provide computational complexity comparisons with other methods including [11] and [65] in Table X. We compare the training time used for one batch with the same batch size and the testing time for one shape. Although our adversarial training is slower than [11], we require less GPU memory to reconstruct 3D shapes in higher resolution than the 32^3 from [11], since we avoid direct 3D reconstruction using neural networks.

TABLE X
ABLATION STUDIES UNDER THE CHAIR CLASS IN TERMS OF IoU

	[11] (Caffe)	[65] (Tensorflow)	Ours (Tensorflow)
Training time(s/batch)	0.2	1.8	0.5
Testing time (s/shape)	0.05	0.04	0.05
GPU (MB)	5400	10200	3500

3) Failure Cases: Finally, we show some failure cases of our method. Because we are using the binarized attenuance images from the trained CGAN to reconstruct 3D shapes in DSO, we can not reveal the detailed structures, such as the sofas in Fig. 19. As mentioned earlier, although the CGAN can learn to distinguish between foreground (areas with pixel values of 1) and background (areas with pixel values of 0) very reliably, it is not able to precisely predict the attenuance information from sketches, which may affect the reconstruction using DSO. Hence we binarize the predicted attenuance images for the reconstruction in the DSO process.

IX. Conclusion

With current approaches to reconstruct 3D shapes from hand-drawn sketches, it is still challenging to generate high fidelity and high resolution 3D voxelized shapes, because it is hard to handle the computational costs caused by the cubic complexity of 3D voxels in deep learning models. In contrast, our method successfully resolves this issue by reconstructing 3D shapes without the involvement of voxel-based deep learning models. Instead, our method effectively leverages a CGAN to understand sketches by translating them into attenuance images with more geometric details. We then use a direct shape optimization technique to generate the reconstructed 3D shape based on the predicted attenuance images. In addition, our progressive update approach is able to robustly handle inconsistencies among multiple hand-drawn sketches for the same 3D shape, which enables users to reconstruct high fidelity 3D shapes using just a few sketches. Our reconstructed results show that our method outperforms the state-of-the-art methods.

References

[1] D. Waltz, "Understanding line drawings of scenes with shadows," in *The Psychology of Computer Vision*. New York, NY, USA: McGraw-Hill, 1975.

[2] J. Malik, "Interpreting line drawings of curved objects," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 73–103, 1987.

[3] H. Lipson and M. Shpitalni, "Optimization-based reconstruction of a 3D object from a single freehand line drawing," *Comput.-Aided Des.*, vol. 28, no. 8, pp. 651–663, Aug. 2007.

[4] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes, "Sketch: An interface for sketching 3D scenes," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 163–170.

[5] P. Navarro, J. I. Orlando, C. Delrieux, and E. Iarussi, "SketchZooms: Deep multi-view descriptors for matching line drawings," 2019, *arXiv:1912.05019*. [Online]. Available: http://arxiv.org/abs/1912.05019

[6] B. Wailly and A. Bousseau, "Line rendering of 3D meshes for data-driven sketch-based modeling," *Journées Francaises d'Informatique Graphique et de Réalité virtuelle*, 2019.

[7] Y. Gryaditskaya, M. Sypesteyn, J. W. Hoftijzer, S. Pont, F. Durand, and A. Bousseau, "OpenSketch: A richly-annotated dataset of product design sketches," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–16, Nov. 2019.

[8] Y. Shen, C. Zhang, H. Fu, K. Zhou, and Y. Zheng, "DeepSketchHair: Deep sketch-based 3D hair modeling," 2019, *arXiv:1908.07198*. [Online]. Available: http://arxiv.org/abs/1908.07198

[9] A. Bonnici *et al.*, "Sketch-based interaction and modeling: Where do we stand?" *Artif. Intell. Eng. Des., Anal. Manuf.*, vol. 33, no. 4, pp. 370–388, 2019.

[10] D. Yu *et al.*, "SketchDesc: Learning local sketch descriptors for multi-view correspondence," 2020, *arXiv:2001.05744*. [Online]. Available: http://arxiv.org/abs/2001.05744

[11] J. Delanoy, M. Aubry, P. Isola, A. A. Efros, and A. Bousseau, "3D sketching using multi-view deep volumetric prediction," *ACM Comput. Graph. Interact. Techn.*, vol. 1, no. 1, pp. 21:1–21:22, 2018.

[12] J. Delanoy, D. Coeurjolly, J.-O. Lachaud, and A. Bousseau, "Combining voxel and normal predictions for multi-view 3D sketching," *Comput. Graph.*, vol. 82, pp. 65–72, Aug. 2019.

[13] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[14] S. Kuanar, V. Athitsos, D. Mahapatra, K. R. Rao, Z. Akhtar, and D. Dasgupta, "Low dose abdominal CT image reconstruction: An unsupervised learning based approach," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1351–1355.

[15] S. Kuanar, V. Athitsos, N. Pradhan, A. Mishra, and K. R. Rao, "Cognitive analysis of working memory load from eeg, by a deep recurrent neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2576–2580.

[16] S. Kuanar, K. R. Rao, M. Bilas, and J. Bredow, "Adaptive CU mode selection in HEVC intra prediction: A deep learning approach," *Circuits, Syst., Signal Process.*, vol. 38, no. 11, pp. 5081–5102, Nov. 2019.

[17] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[18] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[19] Z. Han *et al.*, "SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by RNN with attention," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 672–685, Feb. 2019.

[20] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker, "View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions," in *Proc. AAAI*, 2019, pp. 8376–8384.

[21] Z. Han, X. Liu, Y.-S. Liu, and M. Zwicker, "Parts4Feature: Learning 3D global features from generally semantic parts in multiple views," in *Proc. IJCAI*, 2019, pp. 766–773.

[22] Z. Han *et al.*, "3D2SeqViews: Aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3986–3999, Aug. 2019.

[23] Z. Han, X. Wang, C. M. Vong, Y.-S. Liu, M. Zwicker, and C. L. P. Chen, "3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 758–765.

[24] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, "Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network," in *Proc. AAAI*, 2019, pp. 8778–8785.

[25] X. Liu, Z. Han, X. Wen, Y.-S. Liu, and M. Zwicker, "L2G auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 989–997.

[26] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (DV)*, Sep. 2018, pp. 728–737.

[27] T. Hu, Z. Han, and M. Zwicker, "3D shape completion with multi-view consistent inference," 2019, *arXiv:1911.12465*. [Online]. Available: http://arxiv.org/abs/1911.12465

[28] T. Hu, Z. Han, A. Shrivastava, and M. Zwicker, "Render4Completion: Synthesizing multi-view depth maps for 3D shape completion," 2019, *arXiv:1904.08366*. [Online]. Available: http://arxiv.org/abs/1904.08366

[29] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1939–1948.

[30] T. Hu, Z. Han, and M. Zwicker, "3D shape completion with multi-view consistent inference," in *Proc. AAAI*, 2020, pp. 1–8.

[31] Z. Han, X. Wang, Y.-S. Liu, and M. Zwicker, "Multi-angle point cloud-VAE: Unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction," in *Proc. ICCV*, 2019, pp. 10441–10450.

[32] S. Tulsiani, A. A. Efros, and J. Malik, "Multi-view consistency as supervisory signal for learning shape and pose prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Regognition*, Jun. 2018, pp. 2897–2905.

[33] E. Insafutdinov and A. Dosovitskiy, "Unsupervised learning of shape and pose with differentiable point clouds," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2807–2817.

[34] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 165–174.

[35] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4460–4470.

[36] Y. Jiang, D. Ji, Z. Han, and M. Zwicker, "SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization," 2019, *arXiv:1912.07109*. [Online]. Available: http://arxiv.org/abs/1912.07109

[37] T. Hu, G. Lin, Z. Han, and M. Zwicker, "Learning to generate dense point clouds with textures on multiple categories," 2019, *arXiv:1912.10545*. [Online]. Available: http://arxiv.org/abs/1912.10545

[38] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images," in *Proc. ICML*, 2020, pp. 1–12.

[39] Z. Han, G. Qiao, Y.-S. Liu, and M. Zwicker, "SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates," 2020, *arXiv:2003.05559*. [Online]. Available: http://arxiv.org/abs/2003.05559

[40] Z. Han, M. Shang, X. Wang, Y.-S. Liu, and M. Zwicker, "Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences," in *Proc. AAAI*, 2019, pp. 126–133.

[41] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 1–10.

[42] S. Zhang, Z. Han, Y.-K. Lai, M. Zwicker, and H. Zhang, "Stylistic scene enhancement GAN: Mixed stylistic enhancement generation for 3D indoor scenes," *Vis. Comput.*, vol. 35, nos. 6–8, pp. 1157–1169, Jun. 2019.

[43] A. Dai, C. Diller, and M. Nießner, "SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans," 2019, *arXiv:1912.00036*. [Online]. Available: http://arxiv.org/abs/1912.00036

[44] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. L. P. Chen, "Mesh convolutional restricted Boltzmann machines for unsupervised learning of features with structure preservation on 3-D meshes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2268–2281, Oct. 2017.

[45] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and X. Li, "Unsupervised 3D local feature learning by circle convolutional restricted Boltzmann machine," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5331–5344, Nov. 2016.

[46] Z. Han *et al.*, "BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3707–3720, Aug. 2017.

[47] Z. Han *et al.*, "Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3049–3063, Jun. 2018.

[48] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2Mesh++: Multi-view 3D mesh generation via deformation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1042–1051.

[49] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[50] X. Wen, Z. Han, X. Liu, and Y.-S. Liu, "Point2SpatialCapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules," 2019, *arXiv:1908.11026*. [Online]. Available: http://arxiv.org/abs/1908.11026

[51] X. Wen, Z. Han, G. Youk, and Y.-S. Liu, "CF-SIS: Semantic-instance segmentation of 3D point clouds by context fusion with self-attention," in *Proc. ACM Int. Conf. Multimedia*, 2020.

[52] X. Liu, Z. Han, F. Hong, Y.-S. Liu, and M. Zwicker, "LRC-Net: Learning discriminative features on point clouds by encoding local region contexts," *Comput. Aided Geometric Des.*, vol. 79, May 2020, Art. no. 101859.

[53] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. L. P. Chen, "Unsupervised learning of 3-D local features from raw voxels based on a novel permutation voxelization strategy," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 481–494, Feb. 2019.

[54] J. Malik, "Recovering three dimensional shape from a single image of curved objects," in *Proc. 10th Int. Joint Conf. Artif. Intell. (IJCAI)*, 1987, pp. 734–737.

[55] F. Cordier, H. Seo, M. Melkemi, and N. Sapidis, "Inferring mirror symmetric 3D shapes from sketches," *Comput.-Aided Des.*, vol. 45, pp. 301–311, Feb. 2013.

[56] A. Jung, S. Hahmann, D. Rohmer, A. Begault, L. Boissieux, and M.-P. Cani, "Sketching folds: Developable surfaces from non-planar silhouettes," *ACM Trans. Graph.*, vol. 34, no. 5, pp. 155:1–155:12, 2015.

[57] R. Schmidt, A. Khan, K. Singh, and G. Kurtenbach, "Analytic drawing of 3D scaffolds," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 149:1–149:10, 2009.

[58] B. Xu, W. Chang, A. Sheffer, A. Bousseau, J. McCrae, and K. Singh, "True2Form: 3D curve networks from 2D sketches via selective regularization," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 131:1–131:13, 2014.

[59] H. Pan, Y. Liu, A. Sheffer, N. Vining, C.-J. Li, and W. Wang, "Flow aligned surfacing of curve networks," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 127:1–127:10, 2015.

[60] Q. Xu, Y. Gingold, and K. Singh, "Inverse toon shading: Interactive normal field modeling with isophotes," in *Proc. Workshop Sketch-Based Interfaces Modeling*, 2015, pp. 15–25.

[61] L. Wang, C. Qian, J. Wang, and Y. Fang, "Unsupervised learning of 3D model reconstruction from hand-drawn sketches," in *Proc. ACM Multimedia Conf. (MM)*, 2018, pp. 1820–1828.

[62] X. Han, C. Gao, and Y. Yu, "DeepSketch2Face: A deep learning based sketching system for 3D face and caricature modeling," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[63] D. Smirnov, M. Fisher, V. G. Kim, R. Zhang, and J. Solomon, "Deep parametric shape predictions using distance fields," 2019, *arXiv:1904.08921*. [Online]. Available: http://arxiv.org/abs/1904.08921

[64] C. Li, H. Pan, Y. Liu, A. Sheffer, and W. Wang, "Robust flow-guided neural prediction for sketch-based freeform surface modeling," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 238:1–238:12, 2018.

[65] Z. Lun, M. Gadelha, E. Kalogerakis, S. Maji, and R. Wang, "3D shape reconstruction from sketches via multi-view convolutional networks," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 67–77.

[66] W. Su, D. Du, X. Yang, S. Zhou, and H. Fu, "Interactive sketch-based normal map generation with deep neural networks," *ACM Comput. Graph. Interact. Techn.*, vol. 1, no. 1, pp. 22:1–22:17, 2018.

[67] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[68] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[69] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351. 2015, pp. 234–241.

[70] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[71] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[72] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893

[73] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

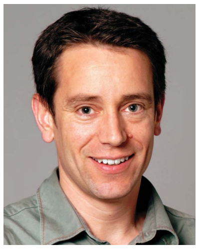
[74] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 2, pp. 191–205, Apr. 2003.

[75] P. Min. (2019). *Binvox*. [Online]. Available: http://www.patrickmin.com/binvox and https://www.google.com/search?q=binvox

Zhizhong Han received the Ph.D. degree from Northwestern Polytechnical University, China, in 2017. He is currently a Postdoctoral Researcher with the Department of Computer Science, University of Maryland, College Park, MD, USA. He is also a Research Member of the BIM Group, Tsinghua University, China. His research interests include machine learning, pattern recognition, feature learning, and digital geometry processing.

Baorui Ma received the B.S. degree in computer science and technology from Jilin University, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Software, Tsinghua University. His research interests include deep learning and 3D reconstruction.

Yu-Shen Liu (Member, IEEE) received the B.S. degree in mathematics from Jilin University, China, in 2000, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2006. He spent three years as a Postdoctoral Researcher with Purdue University from 2006 to 2009. He is currently an Associate Professor with the School of Software, Tsinghua University. His research interests include shape analysis, pattern recognition, machine learning, and semantic search.



Matthias Zwicker (Member, IEEE) received the Ph.D. degree from ETH, Zurich, Switzerland, in 2003. He was an Assistant Professor with the University of California at San Diego, San Diego, CA, USA, and a Professor with the University of Bern, Switzerland. He is currently a Professor with the Department of Computer Science, University of Maryland, College Park, MA, USA, where he holds the Reginald Allan Hahne Endowed E-nnovate Chair. His research in computer graphics covers signal processing for high-quality rendering, point-based methods for rendering and modeling, 3D geometry processing, and data-driven modeling and animation.