Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning

Shichen Liu[1,2], Tianye Li[1,2], Weikai Chen[1], and Hao Li[1,2,3]

[1]USC Institute for Creative Technologies
[2]University of Southern California
[3]Pinscreen

{lshichen, tli, wechen}@ict.usc.edu hao@hao-li.com

Abstract

Rendering bridges the gap between 2D vision and 3D scenes by simulating the physical process of image formation. By inverting such renderer, one can think of a learning approach to infer 3D information from 2D images. However, standard graphics renderers involve a fundamental discretization step called rasterization, which prevents the rendering process to be differentiable, hence able to be learned. Unlike the state-of-the-art differentiable renderers [30, 20], which only approximate the rendering gradient in the back propagation, we propose a truly differentiable rendering framework that is able to (1) directly render colorized mesh using differentiable functions and (2) back-propagate efficient supervision signals to mesh vertices and their attributes from various forms of image representations, including silhouette, shading and color images. The key to our framework is a novel formulation that views rendering as an aggregation function that fuses the probabilistic contributions of all mesh triangles with respect to the rendered pixels. Such formulation enables our framework to flow gradients to the occluded and far-range vertices, which cannot be achieved by the previous state-of-the-arts. We show that by using the proposed renderer, one can achieve significant improvement in 3D unsupervised single-view reconstruction both qualitatively and quantitatively. Experiments also demonstrate that our approach is able to handle the challenging tasks in image-based shape fitting, which remain nontrivial to existing differentiable renderers. Code is available at https://github.com/ShichenLiu/SoftRas.

1. Introduction

Understanding and reconstructing 3D scenes and structures from 2D images has been one of the fundamental goals in computer vision. The key to image-based 3D reasoning is to find sufficient supervisions flowing from the pixels to the 3D properties. To obtain image-to-3D correlations, prior approaches mainly rely on the matching losses based on 2D key points/contours [3, 36, 27, 33] or shape/appearance priors [1, 29, 6, 24, 50]. However, the above approaches are either limited to task-specific domains or can only provide weak supervision due to the sparsity of the 2D features. In contrast, as the process of producing 2D images from 3D assets, rendering relates each pixel with the 3D parameters by simulating the physical mechanism of image formulation. Hence, by inverting a renderer, one can obtain *dense* pixel-level supervision for *general-purpose* 3D reasoning tasks, which cannot be achieved by conventional approaches.

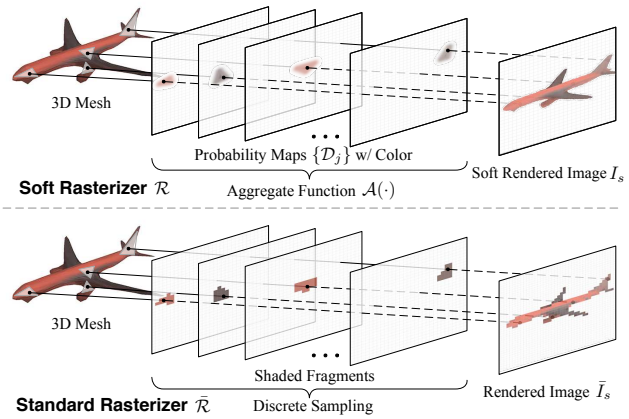


Figure 1: We propose Soft Rasterizer $\mathcal{R}$ (upper), a truly differentiable renderer, which formulates rendering as a differentiable aggregating process $\mathcal{A}(\cdot)$ that fuses per-triangle contributions $\{\mathcal{D}_i\}$ in a "soft" probabilistic manner. Our approach attacks the core problem of differentiating the standard rasterizer, which cannot flow gradients from pixels to geometry due to the discrete sampling operation (below).

However, the rendering process is not differentiable in conventional graphics pipelines. In particular, standard mesh renderer involves a discrete sampling operation, called *rasterization*, which prevents the gradient to be flowed into the mesh vertices. Since the forward rendering function is highly non-linear and complex, to achieve differentiable rendering, recent advances [30, 20] only approx-

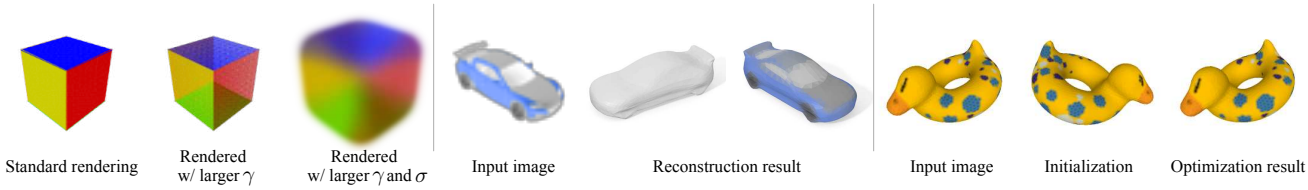


Figure 2: Forward rendering: various rendering effects generated by SoftRas (left). Different degrees of transparency Applications based on the backward gradients provided by SoftRas: (1) 3D unsupervised mesh reconstruction from a single input image (middle) and (2) 3D pose fitting to the target image by flowing gradient to the occluded triangles (right).

imate the backward gradient with hand-crafted functions while directly employing a standard graphics renderer in the forward pass. While promising results have been shown in the task of image-based 3D reconstruction, the inconsistency between the forward and backward propagations may lead to uncontrolled optimization behaviors and limited generalization capability to other 3D reasoning tasks. We show in Section 5.2 that such mechanism would cause problematic situations in image-based shape fitting where the 3D parameters cannot be efficiently optimized.

In this paper, instead of studying a better form of rendering gradient, we attack the key problem of differentiating the forward rendering function. Specifically, we propose a *truly differentiable* rendering framework that is able to render a colorized mesh in the forward pass (Figure 1). In addition, our framework can consider a variety of 3D properties, including mesh geometry, vertex attributes (color, normal *etc.*), camera parameters and illuminations and is able to flow efficient gradients from pixels to mesh vertices and their attributes. While being a universal module, our renderer can be plugged into either a neural network or a non-learning optimization framework without parameter tuning.

The key to our approach is the novel formulation, which views rendering as a "soft" probabilistic process. Unlike the standard rasterizer, which only selects the color of the closest triangle in the viewing direction (Figure 1 below), we propose that all triangles have probabilistic contributions to each rendered pixel, which can be modeled as probability maps on the screen space. While conventional rendering pipelines merge shaded fragments in a one-hot manner, we propose a differentiable aggregation function that fuses the per-triangle color maps based on the probability maps and the triangles' relative depths to obtain the final rendering result (Figure 1 upper). The novel aggregating mechanism enables our renderer to flow gradients to all mesh triangles, including the occluded ones. In addition, our framework can propagate supervision signals from pixels to far-range triangles because of its probabilistic formulation. We call our framework *Soft Rasterizer (SoftRas)* as it "softens" the discrete rasterization to enable differentiability.

Thanks to the consistent forward and backward propagations, SoftRas is able to provide high-quality gradient flows that supervise a variety of tasks on image-based 3D reasoning. To evaluate the performance of SoftRas, we show applications in 3D unsupervised single-view mesh reconstruction and image-based shape fitting (Figure 2, Section 5.1 and 5.2). In particular, as SoftRas provides strong error signals to the mesh generator simply based on the rendering loss, one can achieve mesh reconstruction from a single image without any 3D supervision. To faithfully texture the mesh, we further propose a novel approach that extracts representative colors from input image and formulates the color regression as a classification problem. Regarding the task of image-based shape fitting, we show that our approach is able to (1) handle occlusions using the aggregating mechanism that considers the probabilistic contributions of all triangles; and (2) provide much smoother energy landscape, compared to other differentiable renderers, that avoids local minima by using the smooth rendering (Figure 2 left). Experimental results demonstrate that our approach significantly outperforms the state-of-the-arts both quantitatively and qualitatively.

2. Related Work

Differentiable Rendering. To relate the changes in the observed image with that in the 3D shape manipulation, a number of existing techniques have utilized the derivatives of rendering [11, 10, 31]. Recently, Loper and Black [30] introduce an approximate differentiable renderer which generates derivatives from projected pixels to the 3D parameters. Kato et al. [20] propose to approximate the backward gradient of rasterization with a hand-crafted function to achieve differentiable rendering. More recently, Li et al. [25] introduce a differentiable ray tracer to realize the differentiability of secondary rendering effects. Insafutdinov et al. [17] propose a differentiable renderer for point clouds. Recent advances in 3D face reconstruction [40, 42, 41, 43, 9], material inference [28, 7] and other 3D reconstruction tasks [51, 38, 34, 14, 23, 35, 39] have leveraged some other forms of differentiable rendering layers to obtain gradient flows in the neural networks. However, these rendering layers are usually designed for special purpose and thus cannot be generalized to other applications. In this paper, we focus on a general-purpose differentiable rendering framework that is able to directly render a given mesh using differentiable functions instead of only approximating the backward derivatives.

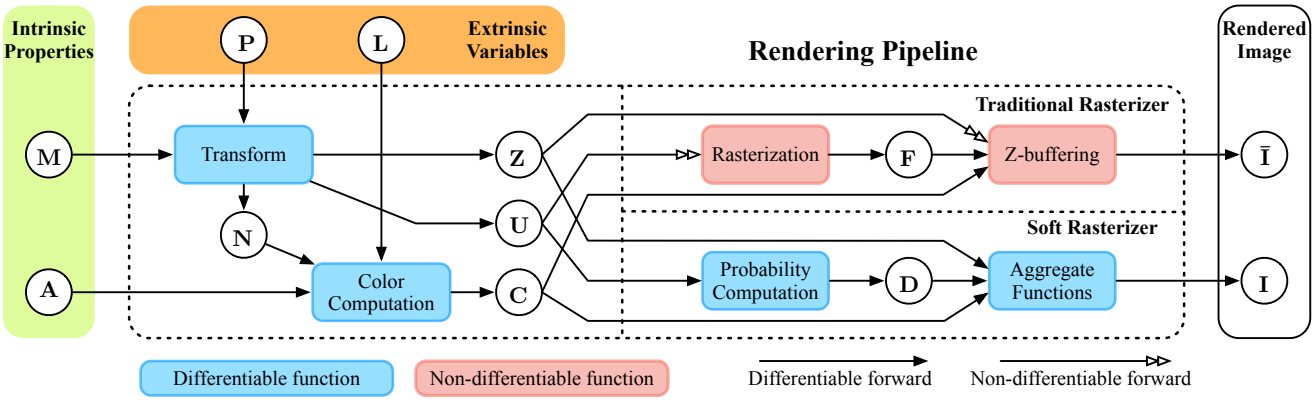


Figure 3: Comparisons between the standard rendering pipeline (upper branch) and our rendering framework (lower branch).

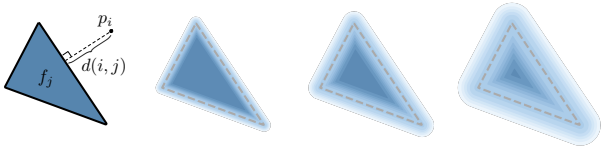
Image-based 3D Reasoning. 2D images are widely used as the media for reasoning 3D properties. In particular, image-based reconstruction has received the most attentions. Conventional approaches mainly leverage the stereo correspondence based on the multi-view geometry [13, 8] but are restricted to the coverage provided by the multiple views. With the availability of large-scale 3D shape dataset [5], learning-based approaches [45, 12, 15] are able to consider single or few images thanks to the shape prior learned from the data. To simplify the learning problem, recent works reconstruct 3D shape via predicting intermediate 2.5D representations, such as depth map [26], image collections [19], displacement map [16] or normal map [37, 46]. Pose estimation is another key task to understanding the visual environment. For 3D rigid pose estimation, while early approaches attempt to cast it as classification problem [44], recent approaches [21, 48] can directly regress the 6D pose by using deep neural networks. Estimating the pose of non-rigid objects, e.g. human face or body, is more challenging. By detecting the 2D key points, great progress has been made to estimate the 2D poses [32, 4, 47]. To obtain 3D pose, shape priors [1, 29] have been incorporated to minimize the shape fitting errors in recent approaches [3, 4, 18, 2]. Our proposed differentiable renderer can provide dense rendering supervision to 3D properties, benefitting a variety of image-based 3D reasoning tasks.

3. Soft Rasterizer

3.1. Differentiable Rendering Pipeline

As shown in Figure 3, we consider both extrinsic variables (camera $\mathbf{P}$ and lighting conditions $\mathbf{L}$) that define the environmental settings, and intrinsic properties (triangle meshes $\mathbf{M}$ and per-vertex appearance $\mathbf{A}$, including color, material *etc*.) that describe the model-specific properties. Following the standard rendering pipeline, one can obtain the mesh normal $\mathbf{N}$, image-space coordinate $\mathbf{U}$ and view-dependent depths $\mathbf{Z}$ by transforming input geometry $\mathbf{M}$ based on camera $\mathbf{P}$. With specific assumptions of illumination and material models (e.g. Phong model), we can compute color $\mathbf{C}$ given $\{\mathbf{A}, \mathbf{N}, \mathbf{L}\}$. These two modules are *naturally differentiable*. However, the subsequent operations: *rasterization* and *z-buffering*, in the standard graphics pipeline (Figure 3 red blocks) are not differentiable with respect to $\mathbf{U}$ and $\mathbf{Z}$ due to the discrete sampling operations.

Our differentiable formulation. We take a different perspective that the rasterization can be viewed as *binary masking* that is determined by the relative positions between the pixels and triangles, while z-buffering merges the rasterization results $\mathbf{F}$ in a pixel-wise *one-hot* manner based on the relative depths of triangles. The problem is then formulated as modeling the discrete binary masks and the one-hot merging operation in a soft and differentiable manner. We therefore propose two major components, namely probability maps $\mathbf{D} = \{\mathcal{D}_j\}$ that model the probability of each pixel staying inside a specific triangle f_j and aggregate function $\mathcal{A}(\cdot)$ that fuses per-triangle color maps based on $\{\mathcal{D}_j\}$ and the relative depths among triangles. With such formulation, all 3D properties, e.g. *camera*, *texture*, *material*, *lighting* and *geometry*, could receive gradients from the image.



(a) ground truth (b) $\sigma = 0.003$ (c) $\sigma = 0.01$ (d) $\sigma = 0.03$

Figure 4: Probability maps of a triangle under Euclidean metric. (a) definition of pixel-to-triangle distance; (b)-(d) probability maps generated with different σ.

3.2. Probability Map Computation

We model the influence of triangle f_j on image plane by probability map $\mathcal{D}_j$. To estimate the probability of $\mathcal{D}_j$ at pixel p_i, the function is required to take into account both the relative position and the distance between p_i and $\mathcal{D}_j$. To this end, we define $\mathcal{D}_j$ at pixel p_i as follows:

$$\mathcal{D}_j^i = sigmoid(\delta_j^i \cdot \frac{d^2(i,j)}{\sigma}), \tag{1}$$

where σ is a positive scalar that controls the sharpness of the probability distribution while δ_j^i is a sign indicator $\delta_j^i = \{+1, \text{if } p_i \in f_j; -1, \text{otherwise}\}$. We set σ as 1×10^{-4} unless otherwise specified. $d(i,j)$ is the closest distance from p_i to f_j's edges. A natural choice for $d(i,j)$ is the Euclidean distance. However, other metrics, such as barycentric or l_1 distance, can be used in our approach.

Intuitively, by using the *sigmoid* function, Equation 1 normalizes the output to $(0, 1)$, which is a faithful continuous approximation of binary mask with boundary landed on 0.5. In addition, the sign indicator maps pixels inside and outside f_j to the range of $(0.5, 1)$ and $(0, 0.5)$ respectively. Figure 4 shows $\mathcal{D}_j$ of a triangle with varying σ using Euclidean distance. Smaller σ leads to sharper probability distribution while larger σ tends to blur the outcome. This design allows controllable influence for triangles on image plane. As $\sigma \to 0$, the resulting probability map converges to the exact shape of the triangle, enabling our probability map computation to be a generalized form of traditional rasterization.

3.3. Aggregate Function

For each mesh triangle f_j, we define its color map C_j at pixel p_i on the image plane by interpolating vertex color using barycentric coordinates. We clip its barycentric coordinates to [0, 1] and normalize their sum amounts to 1, which prevents negative barycentric coordinate for color computation. We then propose to use an aggregate function $\mathcal{A}(\cdot)$ to merge color maps $\{C_j\}$ to obtain rendering output I based on $\{\mathcal{D}_j\}$ and the relative depths $\{z_j\}$. Inspired by the softmax operator, we define an aggregate function $\mathcal{A}_S$ as follows:

$$I^i = \mathcal{A}_S(\{C_j\}) = \sum_j w_j^i C_j^i + w_b^i C_b, \tag{2}$$

where C_b is the background color; the weights $\{w_j\}$ satisfy $\sum_j w_j^i + w_b^i = 1$ and are defined as:

$$w_j^i = \frac{\mathcal{D}_j^i \exp(z_j^i/\gamma)}{\sum_k \mathcal{D}_k^i \exp(z_k^i/\gamma) + \exp(\epsilon/\gamma)}, \tag{3}$$

where z_j^i denotes the normalized negative depth of the 3D point on f_i whose 2D projection is p_i; ϵ is small constant that enables the background color while γ (set as 1×10^{-4} unless otherwise specified) controls the sharpness of the aggregate function. Note that w_j is a function of two major variables: $\mathcal{D}_j$ and z_j. Specifically, w_j assigns higher weight to closer triangles that have larger z_j. As $\gamma \to 0$, the color aggregation function only outputs the color of nearest triangle, which exactly matches the behavior of z-buffering. In addition, w_j is robust to z-axis translations. $\mathcal{D}_j$ modulates the w_j along the x, y directions such that the triangles closer to p_i on screen space will receive higher weight.

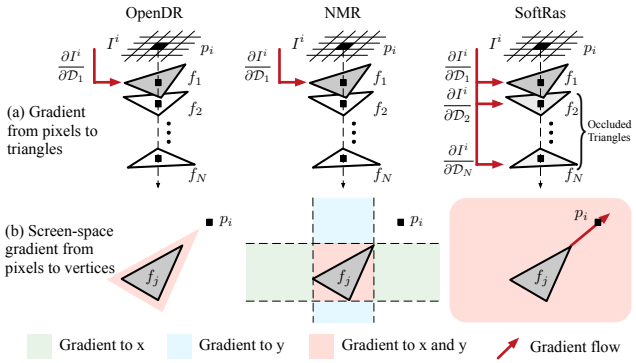


Figure 5: Comparisons with prior differentiable renderers in terms of gradient flow.

Equation 2 also works for shading images when the intrinsic vertex colors are set to constant ones. We further explore the aggregate function for silhouettes. Note that the silhouette of object is independent from its color and depth map. Hence, we propose a dedicated aggregation function $\mathcal{A}_O$ for the silhouette based on the binary occupancy:

$$I_s^i = \mathcal{A}_O(\{\mathcal{D}_j\}) = 1 - \prod_j (1 - \mathcal{D}_j^i). \tag{4}$$

Intuitively, Equation 4 models silhouette as the probability of having *at least* one triangle cover the pixel p_i. Note that there might exist other forms of aggregate functions. One alternative option may be using a universal aggregate function $\mathcal{A}_N$ that is implemented as a neural network. We provide an ablation study on this regard in Section 5.1.4.

3.4. Comparisons with Prior Works

In this section, we compare our approach with the state-of-the-art rasterization-based differential renderers: OpenDR [30] and NMR [20], in terms of gradient flows as shown in Figure 5. We provide detailed analysis on gradient computation in supplemental materials.

Gradient from pixels to triangles. Since both OpenDR and NMR utilize standard graphics renderer in the forward pass, they have no control over the intermediate rendering process and thus cannot flow gradient into the triangles that are occluded in the final rendered image (Figure 5(a) left and middle). In addition, as their gradients only operate on the image plane, both OpenDR and NMR are not able to optimize the depth value z of the triangles. In contrast, our approach has full control on the internal variables and is able to flow gradients to invisible triangles and the z coordinates of all triangles through the aggregation function (Figure 5(a) right).

Screen-space gradient from pixels to vertices. Thanks to our continuous probabilistic formulation, in our approach, the gradient from pixel p_j in screen space can flow gradient to all distant vertices (Figure 5(b) right). However, for OpenDR, a vertex can only receive gradients from

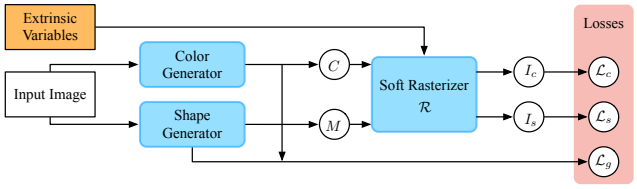


Figure 6: The proposed framework for single-view mesh reconstruction.

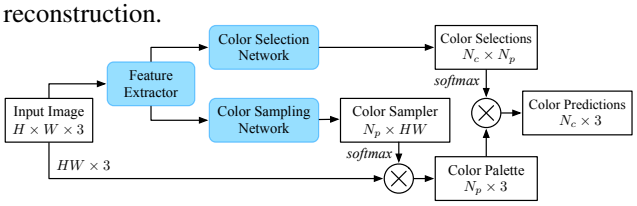


Figure 7: Network structure for color reconstruction.

neighboring pixels within a close distance due to the local filtering operation (Figure 5(b) left). Regarding NMR, there is no gradient defined from the pixels inside the white regions with respect to the triangle vertices ((Figure 5(b) middle). In contrast, our approach does not have such issue thanks to our orientation-invariant formulation.

4. Image-based 3D Reasoning

With direct gradient flow from image to 3D properties, SoftRas enables a variety of tasks on 3D reasoning.

4.1. Single-view Mesh Reconstruction

To demonstrate the effectiveness of soft rasterizer, we fix the extrinsic variables and evaluate its performance on single-view 3D reconstruction by incorporating it with a mesh generator. The direct gradient from image pixels to shape and color generators enables us to achieve *3D unsupervised* mesh reconstruction. Our framework is demonstrated in Figure 6. Given an input image, our shape and color generators generate a triangle mesh M and its corresponding colors C, which are then fed into the soft rasterizer. The SoftRas layer renders both the silhouette I_s and color image I_c and provide rendering-based error signal by comparing with the ground truths. Inspired by the latest advances in mesh learning [20, 45], we leverage a similar idea of synthesizing 3D model by deforming a template mesh. To validate the performance of soft rasterizer, the shape generator employ an encoder-decoder architecture identical to that of [20, 49]. The details of the shape and generators are described in supplemental materials.

Losses. The reconstruction networks are supervised by three losses: silhouette loss $\mathcal{L}_s$, color loss $\mathcal{L}_c$ and geometry loss $\mathcal{L}_g$. Let $\hat{I}_s$ and I_s denote the predicted and the ground-truth silhouette respectively. The silhouette loss is defined as $\mathcal{L}_s = 1 - \frac{||\hat{I}_s \otimes I_s||_1}{||\hat{I}_s \oplus I_s - \hat{I}_s \otimes I_s||_1}$, where $\otimes$ and $\oplus$ are the element-wise product and sum operators respectively. The color loss is measured as the l_1 norm between the rendered and input image: $\mathcal{L}_c = ||\hat{I}_c - I_c||_1$. To achieve appealing visual quality, we further impose a geometry loss $\mathcal{L}_g$ that regularizes the Laplacian of both shape and color predictions. The final loss is a weighted sum of the three losses:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_c + \mu \mathcal{L}_g. \quad (5)$$

4.1.1 Color Reconstruction

Instead of directly regressing the color value, our color generator formulates color reconstruction as a classification problem that learns to reuse the pixel colors in the input image for each sampling point. Let N_c denote the number of sampling points on M and H, W be the height and width of the input image respectively. However, the computational cost of a naive color selection approach is prohibitive, i.e. $O(HWN_c)$. To address this challenge, we propose a novel approach to colorize mesh using a color palette, as shown in Figure 7. Specifically, after passing input image to a neural network, the extracted features are fed into (1) a sampling network that samples the representative colors for building the palette; and (2) a selection network that combines colors from the palette for texturing the sampling points. The color prediction is obtained by multiplying the color selections with the learned color palette. Our approach reduces the computation complexity to $O(N_d(HW + N_c))$, where N_p is the size of color palette. With a proper setting of N_p, one can significantly reduce the computational cost while achieving sharp and accurate color recovery.

4.2. Image-based Shape Fitting

Image-based shape fitting has a fundamental impact in various tasks, such as pose estimation, shape alignment, model-based reconstruction, *etc*. Yet without direct correlation between image and 3D parameters, conventional approaches have to rely on coarse correspondences, e.g. 2D joints [3] or feature points [36], to obtain supervision signals for optimization. In contrast, SoftRas can directly back-propagate pixel-level errors to 3D properties, enabling dense image-to-3D correspondence for high-quality shape fitting. However, a differentiable renderer has to resolve two challenges in order to be readily applicable. (1) *occlusion awareness*: the occluded portion of 3D model should be able to receive gradients in order to handle large pose changes. (2) *far-range impact*: the loss at a pixel should have influence on distant mesh vertices, which is critical to dealing with local minima during optimization. While prior differentiable renderers [20, 30] fail to satisfy these two criteria, our approach handles these challenges simultaneously. (1) Our aggregate function fuses the probability maps from all triangles, enabling the gradients to be flowed to all vertices including the occluded ones. (2) Our soft approximation based on probability distribution allows the

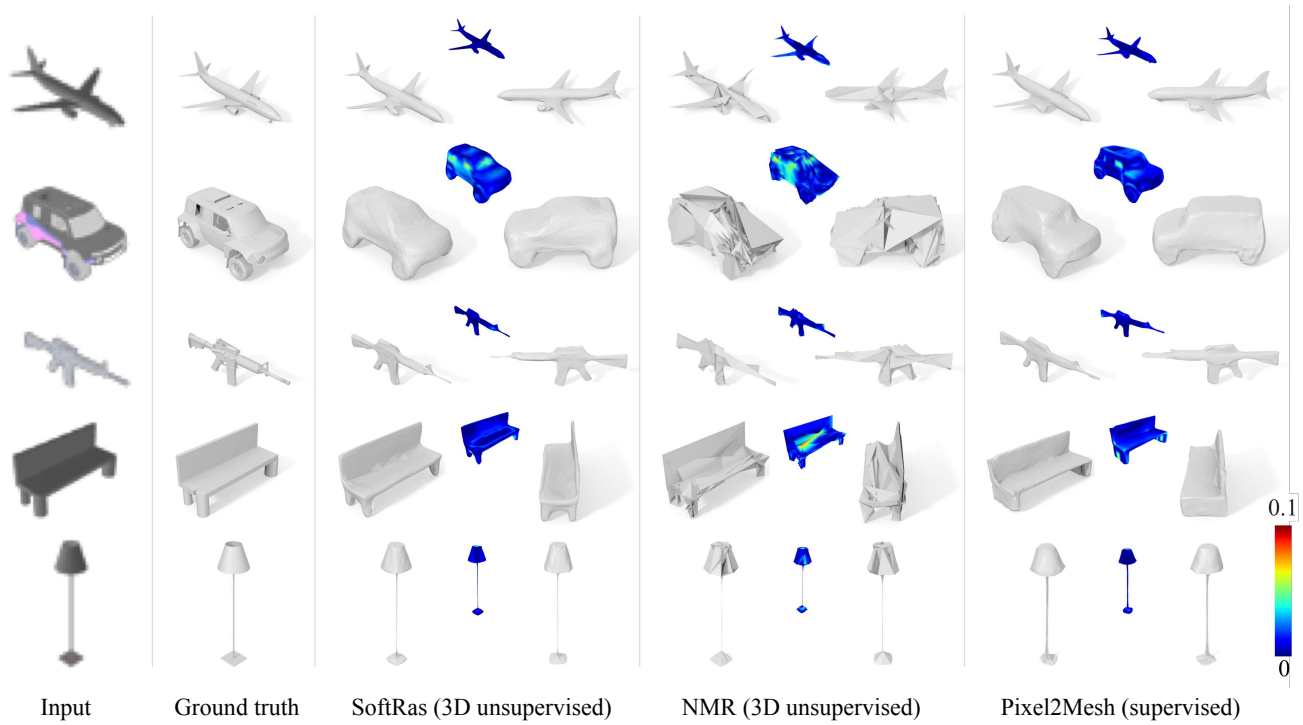


Figure 8: 3D mesh reconstruction from a single image. From left to right, we show input image, ground truth, the results of our method (SoftRas), Neural Mesh Renderer [20] and Pixel2mesh [45] – all visualized from 2 different views. Along with the results, we also visualize mesh-to-scan distances measured from reconstructed mesh to ground truth.

gradient to be propagated to the far end while the size of receptive field can be well controlled (Figure 4). To this end, our approach can faithfully solve the image-based shape fitting problem by minimizing the following energy objective:

$$\underset{\rho,\theta,t}{\operatorname{argmin}} ||R(M(\rho,\theta,t)) - I_t||_2, \tag{6}$$

where $R(\cdot)$ is the rendering function that generates a rendered image I from mesh M, which is parametrized by its pose θ, translation t and non-rigid deformation parameters ρ. The difference between I and the target image I_t provides strong supervision to solve the unknowns $\{\rho, \theta, t\}$.

5. Experiments

5.1. Single-view Mesh Reconstruction

5.1.1 Experimental Setup

Datasets and Evaluation Metrics. We use the dataset provided by [20], which contains 13 categories of objects from ShapeNet [5]. Each object is rendered in 24 different views with image resolution of 64×64. For fair comparison, we employ the same train/validate/test split on the same dataset as in [20, 49]. For quantitative evaluation, we adopt the standard reconstruction metric, 3D intersection over union (IoU), to compare with baseline methods.

Implementation Details. We use the same structure as [20, 49] for mesh generation. Our network is optimized using Adam [22] with $\alpha = 1 \times 10^{-4}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The training of our model takes 12 hours per category on a single NVIDIA 1080Ti GPU. Specifically, we set $\lambda = 1$ and $\mu = 1 \times 10^{-3}$ across all experiments unless otherwise specified. We train the network with multi-view images of batch size 64 and implement it using PyTorch.

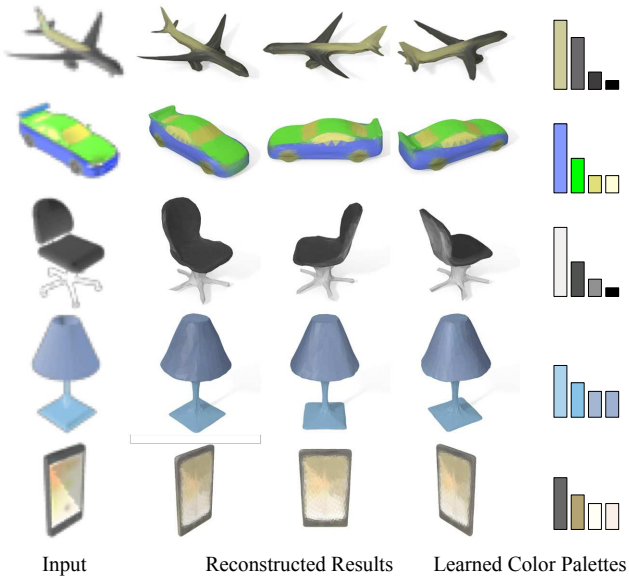


Figure 9: Results of colorized mesh reconstruction. The learned principal colors and their usage histogram are visualize on the right.

Category	Airplane	Bench	Dresser	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Vessel	Mean
retrieval [49]	0.5564	0.4875	0.5713	0.6519	0.3512	0.3958	0.2905	0.4600	0.5133	0.5314	0.3097	0.6696	0.4078	0.4766
voxel [49]	0.5556	0.4924	0.6823	0.7123	0.4494	0.5395	0.4223	0.5868	0.5987	0.6221	0.4938	0.7504	0.5507	0.5736
NMR [20]	0.6172	0.4998	0.7143	0.7095	0.4990	0.5831	0.4126	0.6536	0.6322	0.6735	0.4829	0.7777	0.5645	0.6015
Ours (sil.)	0.6419	0.5080	0.7116	0.7697	0.5270	0.6156	**0.4628**	0.6654	**0.6811**	0.6878	0.4487	0.7895	0.5953	0.6234
Ours (full)	**0.6670**	**0.5429**	**0.7382**	**0.7876**	**0.5470**	**0.6298**	0.4580	**0.6807**	0.6702	**0.7220**	**0.5325**	**0.8127**	**0.6145**	**0.6464**

Table 1: Comparison of mean IoU with other 3D unsupervised reconstruction methods on 13 categories of ShapeNet datasets.

5.1.2 Qualitative Results

Single-view Mesh Reconstruction. We compare the qualitative results of our approach with that of the state-of-the-art supervised [45] and 3D unsupervised [20] mesh reconstruction approaches in Figure 8. Though NMR [20] can recover the rough shape, the mesh surface is discontinuous and suffers from a considerable amount of self intersections. In contrast, our method can faithfully reconstruct fine details of the object, such as the airplane tail and the rifle barrel, while ensuring smoothness of the surface. Though trained without 3D supervision, our approach achieves results on par with the supervised method Pixel2Mesh [45]. In some cases, our approach can generate even more appealing details than that of [45], e.g. the bench legs, the airplane engine and the side of the car. Mesh-to-scan distance visualization also shows our results achieve much higher accuracy than [20] and comparable accuracy with that of [45].

Color Reconstruction. Our method is able to faithfully recover the mesh color based on the input image. Figure 9 presents the colorized reconstruction from a single image and the learned color palettes. Though the resolution of the input image is rather low (64×64), our approach is still able to achieve sharp color recovery and accurately restore the fine details, e.g. the subtle color transition on the body of airplane and the shadow on the phone screen.

5.1.3 Quantitative Evaluations

We show the comparisons on 3D IoU score with the state-of-the-art approaches in Table 1. We test our approach under two settings: one trained with silhouette loss only (sil.) and the other with both silhouette and shading supervisions (full). Our approach has significantly outperformed all the other 3D unsupervised methods on all categories. In addition, the mean score of our best setting has surpassed the state-of-the-art NMR [20] by more than 4.5 points. As we use the identical mesh generator and same training settings with [20], it indicates that it is the proposed SoftRas renderer that leads to the superior performance.

5.1.4 Ablation Study

Loss Terms and Alternative Functions. In Table 2, we investigate the impact of Laplacian regularizer and various forms of the distance function (Section 3.2) and the aggregate function. As the RGB color channel and the α channel (silhouette) have different candidate aggregate functions,

SoftRas settings			$\mathcal{L}_{lap}$	mIoU (%)
distance func.	aggregate func. (α)	aggregate func. (color)		
Barycentric	$\mathcal{A}_O$	-		60.8
Euclidean	$\mathcal{A}_O$	-		62.0
Euclidean	$\mathcal{A}_O$	-	✓	62.4
Euclidean	$\mathcal{A}_N$	-	✓	63.2
Euclidean	$\mathcal{A}_O$	$\mathcal{A}_S$	✓	**64.6**

Table 2: Ablation study of the regularizer and various forms of distance and aggregate functions. $\mathcal{A}_N$ is the aggregation function implemented as a neural network. $\mathcal{A}_S$ and $\mathcal{A}_O$ are defined in Equation 2 and 4 respectively.

Method	w/o scheduling	w/ scheduling
random guess	126.48°[1]	126.48°
NMR[20]	93.40°	80.94°
Li et al.[25]	95.02°	78.56°
SoftRas	**82.80°**	**63.57°**

Table 3: Comparison of cube rotation estimation error with NMR, measured in mean relative angular error.

we separate their lists in Table 2. First, by adding Laplacian constraint, our performance is increased by 0.4 point (62.4 *v.s.* 62.0). In contrast, NMR [20] has reported a negative effect of geometry regularizer on its quantitative results. The performance drop may be due to the fact that the ad-hoc gradient is not compatible with the regularizer. It is optional to have color supervision on the mesh generation. However, we show that adding a color loss can significantly improve the performance (64.6 *v.s.* 62.4) as more information is leveraged for reducing the ambiguity of using silhouette loss only. In addition, we also show that Euclidean metric usually outperforms the barycentric distance while the aggregate function based on neural network $\mathcal{A}_N$ performs slightly better than the non-parametric counterpart $\mathcal{A}_O$ at the cost of more computations.

5.2. Image-based Shape Fitting

Rigid Pose Fitting. We compare our approach with NMR in the task of rigid pose fitting. In particular, given a colorized cube and a target image, the pose of the cube needs to be optimized so that its rendered result matches the target image. Despite the simple geometry, the discontinuity of face colors, the non-linearity of rotation and the large occlusions make it particularly difficult to optimize. As shown in Figure 10, NMR is stuck in a local minimum while our

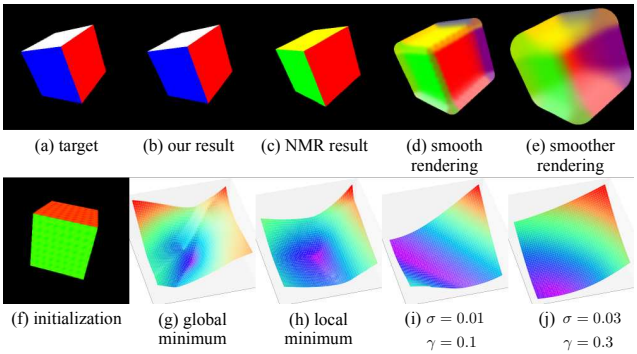


Figure 10: Visualization of loss function landscapes of NMR and SoftRas for pose optimization given target image (a) and initialization (f). SoftRas achieves global minimum (b) with loss landscape (g). NMR is stuck in local minimum (c) with loss landscape (h). At this local minimum, SoftRas produces the smooth and partially transparent rendering (d)(e), which smoothens the loss landscape (i)(j) with larger σ and γ, and consequently leads to better minimum.

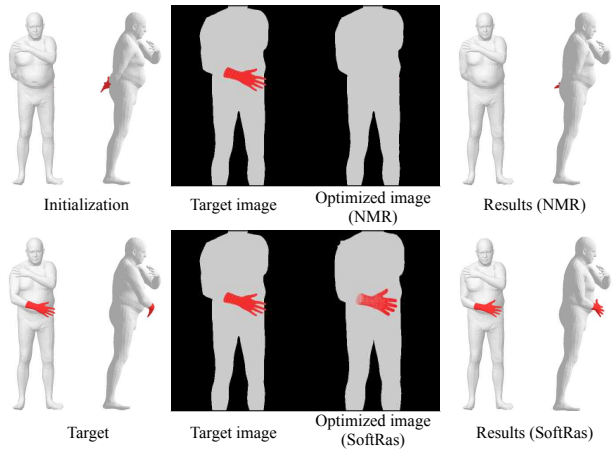


Figure 11: Results for optimizing human pose given single image target.

approach succeeds to obtain the correct pose. The key is that our method produces smooth and partially transparent renderings which "soften" the loss landscape. Such smoothness can be controlled by σ and γ, which allows us to avoid the local minimum. Further, we evaluate the rotation estimation accuracy on synthetic data given 100 randomly sampled initializations and targets. We compare methods w/ and w/o scheduling schemes, and summarize mean relative angle error in Table 3. Without optimization scheduling, our method outperforms the best baseline by 10.60°, demonstrating the effectiveness of the gradient flows provided by our method and the benefit of handling largely occluded triangles. Scheduling is a commonly used technique for solving non-linear optimization problems. For other methods, we solve with multi-resolution images in 5 levels; while for our method, we set schedules to decay σ and γ in 5 steps. While scheduling improves all methods, our approach still achieves better accuracy than the best baseline by 14.99°, indicating our consistent superiority.

[1]The expectation of uniform-sampled SO3 rotation angle is $\pi/2 + 2/\pi$

Non-rigid Shape Fitting. In Figure 11, we show that SoftRas can provide stronger supervision for non-rigid shape fitting even in the presence of part occlusions. We optimize the human body parametrized by SMPL model [29]. As the right hand (textured as red) is completely occluded in the initial view, it is extremely challenging to fit the body pose to the target image. To obtain correct parameters, the optimization should be able to (1) consider the impact of the occluded part on the rendered image and (2) back-propagate the error signals to the occluded vertices. NMR [20] fails to move the hand to the right position due to its incapability to handle occlusions. In comparison, our approach can faithfully complete the task as our novel probabilistic formulation and aggregating mechanism can take all triangles into account while being able to optimize the z coordinates (depth) of the mesh vertices.

6. Conclusions

In this paper, we have presented a truly differentiable rendering framework (SoftRas) that is able to directly render a given mesh in a fully differentiable manner. SoftRas can consider both extrinsic and intrinsic variables in a unified rendering framework and generate efficient gradients flowing from pixels to mesh vertices and their attributes (color, normal, *etc.*). We achieve this goal by reformulating the discrete operations including rasterization and z-buffering as differentiable probabilistic processes. Such novel formulation enables our renderer to flow gradients to unseen vertices and optimize the z coordinates of mesh triangles, leading to significant improvements in the tasks of single-view mesh reconstruction and image-based shape fitting. However, our approach, in current form, cannot handle shadows and topology changes, which are worth investigation in the future.

Acknowledgements

This research was conducted at USC and was funded by in part by the ONR YIP grant N00014-17-S-FO14, the CONIX Research Center, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the Andrew and Erna Viterbi Early Career Chair, the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, Adobe, and Sony. This project was not funded by Pinscreen, nor has it been conducted at Pinscreen or by anyone else affiliated with Pinscreen. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 1, 3

[2] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003. 3

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 1, 3, 5

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. 3

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3, 6

[6] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):681–685, 2001. 1

[7] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (TOG)*, 37(4):128, 2018. 2

[8] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. 3

[9] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018. 2

[10] Ioannis Gkioulekas, Anat Levin, and Todd Zickler. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision*, pages 685–701. Springer, 2016. 2

[11] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):162, 2013. 2

[12] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. *computer vision and pattern recognition*, 2018. 3

[13] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3

[14] Paul Henderson and Vittorio Ferrari. Learning to generate and reconstruct 3d meshes with only 2d supervision. In *British Machine Vision Conference (BMVC)*, 2018. 2

[15] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In *European Conference on Computer Vision*, pages 351–369. Springer, 2018. 3

[16] Loc Huynh, Weikai Chen, Shunsuke Saito, Jun Xing, Koki Nagano, Andrew Jones, Paul Debevec, and Hao Li. Mesoscopic facial geometry inference using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8407–8416, 2018. 3

[17] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2802–2812, 2018. 2

[18] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018. 3

[19] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. *arXiv preprint arXiv:1803.07549*, 2018. 3

[20] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 1, 2, 4, 5, 6, 7, 8

[21] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 3

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

[23] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018. 2

[24] Hendrik Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics (TOG)*, 22(2):234–257, 2003. 1

[25] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 2, 7

[26] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, 2016. 3

[27] Feng Liu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. Joint face alignment and 3d face reconstruction. In *European Conference on Computer Vision*, pages 545–560. Springer, 2016. 1

[28] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2280–2288. IEEE, 2017. 2

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015. 1, 3, 8

[30] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 1, 2, 4, 5

[31] Vikash K Mansinghka, Tejas D Kulkarni, Yura N Perov, and Josh Tenenbaum. Approximate bayesian image interpretation using generative probabilistic graphics programs. In *Advances in Neural Information Processing Systems*, pages 1520–1528, 2013. 2

[32] Iacopo Masi, Stephen Rawls, Gérard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4838–4846, 2016. 3

[33] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000. 1

[34] Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, H-P Seidel, and Tobias Ritschel. Deep shading: convolutional neural networks for screen space shading. In *Computer graphics forum*, volume 36, pages 65–78. Wiley Online Library, 2017. 2

[35] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. *arXiv preprint arXiv:1806.06575*, 2018. 2

[36] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017. 1, 5

[37] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. 3

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

[39] Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. A versatile scene model with differentiable visibility applied to generative pose estimation. In *Proceedings of the 2015 International Conference on Computer Vision (ICCV 2015)*, 2015. 2

[40] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5553–5562. IEEE, 2017. 2

[41] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2549–2559, 2018. 2

[42] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 5, 2017. 2

[43] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7346–7355, 2018. 2

[44] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 3

[45] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 3, 5, 6, 7

[46] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015. 3

[47] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 3

[48] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 3

[49] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 5, 6, 7

[50] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):690–706, 1999. 1

[51] Jacek Zienkiewicz, Andrew Davison, and Stefan Leutenegger. Real-time height map fusion using differentiable rendering. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4280–4287. IEEE, 2016. 2