ReDA:Reinforced Differentiable Attribute for 3D Face Reconstruction

Wenbin Zhu HsiangTao Wu Zeyu Chen Noranart Vesdapunt Baoyuan Wang
Microsoft Cloud and AI
{wenzh,musclewu,zech,noves,baoyuanw}@microsoft.com

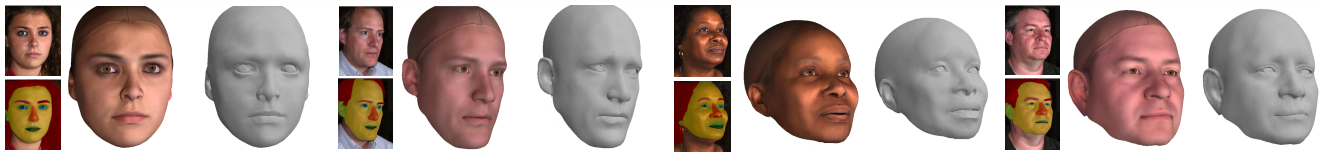

Figure 1: A few RGB-based 3D face reconstruction results by our proposed ReDA framework.

Abstract

The key challenge for 3D face shape reconstruction is to build the correct dense face correspondence between the deformable mesh and the single input image. Given the ill-posed nature, previous works heavily rely on prior knowledge (such as 3DMM [2]) to reduce depth ambiguity. Although impressive result has been made recently [42, 14, 8], there is still a large room to improve the correspondence so that projected face shape better aligns with the silhouette of each face region (i.e, eye, mouth, nose, cheek, etc.) on the image. To further reduce the ambiguities, we present a novel framework called "Reinforced Differentiable Attributes" ("ReDA") which is more general and effective than previous Differentiable Rendering ("DR"). Specifically, we first extend from color to more broad attributes, including the depth and the face parsing mask. Secondly, unlike the previous Z-buffer rendering, we make the rendering to be more differentiable through a set of convolution operations with multi-scale kernel sizes. In the meanwhile, to make "ReDA" to be more successful for 3D face reconstruction, we further introduce a new free-form deformation layer that sits on top of 3DMM to enjoy both the prior knowledge and out-of-space modeling. Both techniques can be easily integrated into existing 3D face reconstruction pipeline. Extensive experiments on both RGB and RGB-D datasets show that our approach outperforms prior arts.

1. Introduction

3D face shape reconstruction has been a hot research topic in both computer vision and graphics literature. Huge progress has been made in the past decade driven by a vast variety of important applications, such as face recognition [3, 24], face reenactment and visual dubbing [43, 33, 21], avatar creation and animation [29, 17, 6] and etc. Despite the impressive progress, face reconstruction is still an ill-posed problem for monocular images due to the depth ambiguity [38] and the albedo illumination ambiguity[9]. Given the nature of insufficient constraint, methods in previous works heavily rely on prior knowledge, such as 3D Morphable Model (3DMM) [2] or multi-linear tensor model [48, 5] to get a reasonable shape. Nevertheless, the key remaining challenge is to build a better dense correspondence between the input image and the deformable mesh.

The most popular way of reconstructing 3D face shape from single image is to use *"Analysis-by-Synthesis"* paradigm [2, 34, 43, 42, 41, 40, 46, 47] to minimize the visual difference between the input and the 2D synthesis of an estimated 3D face through a simplified image formulation model. A typical pixel-wise photo-metric loss is employed for the optimization. After convergence, each pixel will be assigned to a UV position of the template mesh. Hence, *"Analysis-by-Synthesis"* can be considered as an *implicit* way of building dense correspondence. Another typical way is to learn *explicit* dense correspondence first by directly regressing the per-pixel UV position[15] (or equivalent flow [51]), based upon which to fit the 3D face model afterwards. To achieve this, however, one has to use 3DMM fitting[52, 34] to get the ground-truth and then train the regression model through supervised learning [51, 15]. In either case, better correspondence means more accurate 3D reconstruction.

However, there still exist three fundamental issues in all the previous works. First, the capacity of the 3DMM significantly limits the representation power to support diverse geometry variations. Some works [15, 11] propose to directly learn dense correspondence through UV and claim

they are model-free, but it's arguable that their ground-truth space is still limited by the capacity of 3DMM. Recently, works like [4, 46, 40] attempt to represent the geometry in a free-form manner, although inspiring results were obtained, more discriminating constraint is still much desired to build better correspondence. Second, the differentiable rendering[42, 41, 46, 40] used in *"Analysis-by-Synthesis"* paradigm is not truly *"differentiable"*. Most of the existing works simply use Z-buffer rendering, which is not necessarily differentiable especially when the triangles that encloses each pixel are changing during the optimization. Lastly, the expressiveness of a pre-trained texture model used to synthesize the 2D image is another limiting factor. For example, If the texture is over-smooth, it cannot provide any discriminating constraint to drive the optimization and correct the correspondence. In short, we are lacking of extra forces (constraints) to drive the optimization out of the local minimum and steer the gradient towards the right direction.

To circumvent these issues, we present a novel 3D face reconstruction and fitting framework for monocular images. Concretely, we generalize from color to more broad attributes, including the depth and the face parsing mask, in the context of differentiable rendering. We call them as "differentiable attributes", which are supposed to be more effective for driving the correspondence learning thanks to their discriminating constraints. Secondly, we borrow the idea of the soft rasterization [26] and improve it to tailor for more efficient differentiable attributes for 3D face reconstruction. Specifically, we slice the mesh into a few pieces along the Z direction, each of which is rasterized with traditional approach, we then use a few stacked 2D convolution operations with various kernel sizes to aggregate along both the spatial and across the slices to achieve a truly differentiable render. At the end, we obtain a pyramid of rendered image, and then per-pixel attribute discrepancy loss can be employed between the rendered image and the corresponding ground-truth(color, mask label or depth) at each scale of the pyramid. We therefore call the whole process as "Reinforced Differentiable Attibutes (ReDA)", which is the key ingredient of our whole system. In addition, to resolve the limitation of 3DMM capacity and make "ReDA" more successful, we further propose a new free-form deformation layer that sits on top of 3DMM to ensure the mesh geometry has enough space to fit any input image. Unlike previous work [41], we optimize the residual per-vertex displacement in parallel with the 3DMM base mesh (Fig. 2). During training, we apply as-rigid-as-possible deformation constraint between the base mesh and the mesh after adding the residual. Both the free-form layer and ReDA can be easily used for fitting and learning-based 3D face reconstruction, and can be optimized jointly in an end-to-end manner. Our contributions can be summarized as follows:

1. We introduce "ReDA", a reinforced differentiable attribute framework, for better face reconstruction and accurate dense correspondence learning, which is more general and effective than traditional differentiable rendering.
2. We propose a free-form deformation layer with as-rigid-as-possible constraint to further increase the capacity of 3DMM, which is encouraged to be used jointly with ReDA.
3. Our end-to-end system supports both RGB and RGB-D, single image based fitting and deep learning based training. Extensive experiments indicate that our approach is both effective and efficient for 3D face reconstruction.

2. Related Works

2.1. 3D Face Reconstruction

There exist a large body of research in literature for 3D face reconstruction [34, 12, 20, 45, 35, 8], which can be divided into different groups depending on the input modality (RGB or RGB-D), single view or multi-view, optimization-based or learning based, different face models and different constraints being used. Refer to the latest survey [53, 10] for a full literature review. Recent years, there is a rich set of deep learning based 3D reconstruction works that either target only for geometry [44, 19, 25, 18, 49, 37] or both geometry and texture [42, 41, 14, 40, 46, 13] for monocular input. Most of those existing works try to boost the reconstruction accuracy by adding more prior knowledge (i.e., through a parametric face model) or adding more constraints, like sparse landmark loss, perception loss and photo-metric loss etc. Our work follows the line of adding more discriminating constraints to reduce the ambiguities. However, the key difference is that we significantly improve the differentiable rendering to leverage more attributes that beyond color. For example, to our knowledge, we are the first to apply face parsing for 3D reconstruction.

2.2. Differentiable Rendering

Inverse rendering is a long-standing research problem in computer graphics recent works [30, 16, 31, 26] on generic differentiable rendering have been receiving much attention from the community. Specific to face reconstruction, differentiable rendering is also now an indispensable component in the-state-of-the-art deep learning systems, such as [42, 41, 14, 46, 47, 13, 40]. However, the effectiveness of differentiable rendering is largely constrained by the expressiveness of the underlying texture model. Motivated by this, [13] proposes to learn a progressive GAN model to learn the highly nonlinear texture representation as opposed to use the traditional linear PCA model [2, 32]. However, the training requires 10K high quality 3D face texture scans

which are hard to acquire. Another limitation is, the previous differentiable render simply uses Z-buffer rasterization, which is not truly differentiable. This is because, each pixel will be only influenced by the three discrete vertices of its enclosed triangle. The recent work "SoftRas" [26] is fully differentiable and has shown impressive results on a few different 3D objects. However, it is not designed in particular for face reconstruction. As a comparison, our "ReDA" improve on top of "SoftRas" from three perspectives: (1) we add more constraints into the rendering process. Instead of just color, we also use face parsing masks [27, 23]. (2) We use multi-scale convolution operations to perform the soft aggregation across the mesh slices rather than triangles. (3) Instead of interpolating pixel attributes through vertex attributes, we interpolate through attributes on UV maps.

2.3. Semantic Face Segmentation

There exist a set of prior works [9, 36, 47, 11] that utilize semantic face segmentation for robust 3D face reconstructions. However, the way they use the segmentation information is different from ours. [36] proposed a real-time facial segmentation method, based on which to mask out the occluded facial regions before sending to the DDE [5] tracking model. Likewise, [47] leverages a face segmentation model to exclude the occluded areas by glasses, hand and hairs, so that they won't contribute to the optimization process. Similarly, [11] also uses segmentation information to give heuristically defined weights to different facial regions in their reconstruction loss function. However, no work has ever directly leveraged the face parsing mask to build the dense correspondence and to improve the reconstruction especially considering the rapid progress made for face parsing in recent works, such as [23].

2.4. Dense Face Correspondence

As discussed in introduction, a popular way of getting *explicit* dense correspondence is by directly regressing the per-pixel UV position [15] (or equivalent flow [51]). However, the per-pixel ground-truth UV was obtained through 3DMM fitting [52, 34], which certainly limits the expressiveness space due to the 3DMM capacity. Hence, any dense correspondence regression model trained through such supervised learning [51, 15] would be also limited, which is a typical chicken-and-egg problem. We improve the limit of capacity through adding a free-form deformation layer that can support out-of-space modeling, as well as a ReDA module to achieve truly differentiable rendering.

3. Overview

Like the standard "Analysis-by-Synthesis" pipeline, given an input image, our goal is to output the parameters of 3D face model so that the 2D projection matches with the input image. However, we optimize the pipeline by (1) replacing the differentiable rendering with our novel Reinforced Differnetiable Attribute (ReDA) (2) introducing a free-form deformation layer to expand the modeling capacity for better geometry representation. Fig.2 shows an example of the optimization pipeline. Unless otherwise specified, when applicable, we also use the photo-metric loss and 2D landmark loss on the rendered color image. However, our primary focus is to get better face shape through those constraints, so optimizing a photo-realistic texture [13] is out of the current scope.

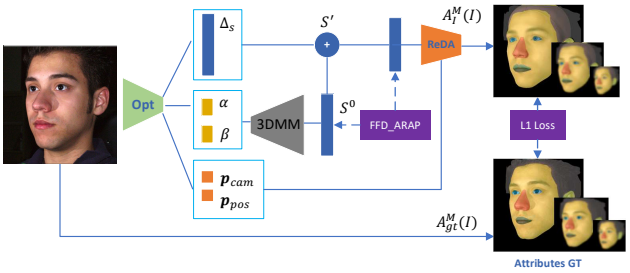

Figure 2: Illustration of the 3D face fitting pipeline. Coefficients are optimized by optimizer Opt. "FFD_ARAP" denotes the free-form deformation loss described in Sec.5.2. Other losses are omitted for simplicity.

4. ReDA:Reinforced Differentiable Attribute

To steer the mesh deformation towards the right shape until the final correspondence is fully achieved, we need an optimization framework. As introduced before (Sec.3), we improve the *"Analysis-by-Synthesis"* through ReDA framework. In the following, unless otherwise specified, we use $\mathcal{A}$ to represent all the differentiable attribute, including color, mask and depth, which are denoted as $\mathcal{A}^{\mathcal{C}}, \mathcal{A}^{\mathcal{M}}, \mathcal{A}^{\mathcal{D}}$ respectively. Of-course, $\mathcal{A}$ could be augmented with more attributes in the future.

4.1. Differentiable Attributes

We extend from color to more broad attributes when used for differentiable rendering. In particular, we propose to bring the face parsing mask into the differentiable procedure and use it to drive the correspondence learning. For an input image I, let us denote $\mathcal{A}_I^{\mathcal{M}}(I)$ as face parsing output from "$ReDA$" and $\mathcal{A}_{gt}^{\mathcal{M}}(I)$ as its face parsing mask ground-truth, which is obtained by either human annotation or a well-trained face parsing model such as [23]. Let us also denote $\mathcal{M}^{UV}$ as the mask UV map for our mesh template, defining the semantic label (i.e., eyebrow, upper lip etc see 3) of each geometry region. Similarly, when color is used as the differentiable attribute, say $\mathcal{A}^{\mathcal{C}}$, we need a corresponding texture UV map $\mathcal{C}^{UV}$. Suppose we use cylindrical unwarp function $\mathcal{F}$ to map a triangle vertex p into the corresponding position in the UV map, namely $\mathcal{UV}(p) = \mathcal{F}(p)$.

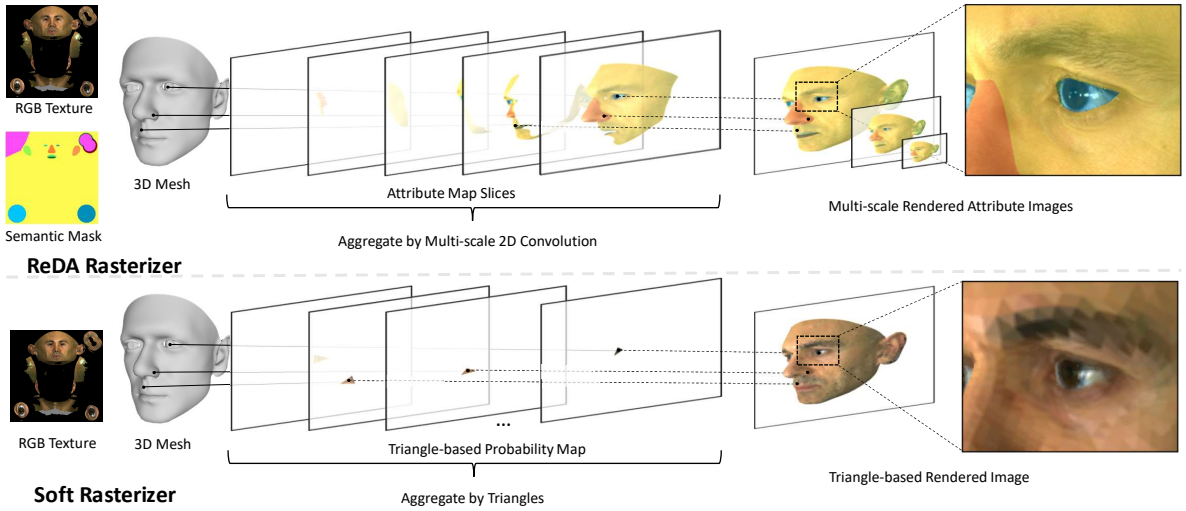

Figure 3: Illustration of our ReDA rasterizer and its comparison with "SoftRas"[26]

For any surface point V_s on the surface of shape S, its UV coordinates can be computed through:

$$\mathcal{UV}(V_s) = (u, v) = \sum_{p \in t} \lambda_p \mathcal{F}(p) \tag{1}$$

where $t = \{p_a, p_b, p_c\}$ represents the three vertices of the triangle that encloses point V_s and λ_p represents the barycentric coordinates of vertex p. When $\mathcal{A}^{\mathcal{M}}$ is used, the mask attribute value $\mathcal{A}_S^{\mathcal{M}}(p)$ for vertex $\mathrm{V_s}$ is computed via bi-linear sampling:

$$\mathcal{A}_S^{\mathcal{M}}(\mathrm{V_s}) = \sum_{\substack{u' \in \{\lfloor u \rfloor, \lceil u \rceil\} \\ v' \in \{\lfloor v \rfloor, \lceil v \rceil\}}} (1-|\mathrm{u}-\mathrm{u}'|)(1-|\mathrm{v}-\mathrm{v}'|) * \mathcal{M}^{\mathrm{UV}}(\mathrm{u}', \mathrm{v}') \tag{2}$$

Then to convert per-vertex attribute values on 3D shapes to per-pixel attribute values on 2D images, we have to go through rendering pipeline. Denote $\mathrm{P_{cam}}$ as the camera projection matrix and $\mathrm{P_{pos}}$ as the pose of the mesh in camera coordinate system. Assume the closest surface point to the image plane V_j(based on depth value) on the shape S maps to pixel I^i on the 2D image I after rendering, then the corresponding mask value $\mathcal{A}_I^{\mathcal{M}}(I^i)$ can be computed through a rendering function $\mathcal{R}$:

$$\mathcal{A}_I^{\mathcal{M}}(I^i) = \mathcal{R}(\mathrm{P_{pos}}, \mathrm{P_{cam}}, V_j, \mathcal{A}_S^{\mathcal{M}}(V_j))) \tag{3}$$

A similar process of equation 1, 2 and 3 can be applied for other attributes such as $\mathcal{A}^C$ for color if we replace $\mathcal{M}^{UV}$ with $\mathcal{C}^{UV}$ in the UV space.

In all previous work, $\mathcal{R}$ is simply defined as the Z-buffer rendering function, where each pixel I^i is only influenced by the nearest triangle to the image plane that encloses V_j, which is however not truly differentiable.

4.2. Soft Rasterization via Convolution Kernel

To remedy the Z-buffer limitation, we need to differentiate the discrete sampling (through an enclosed triangle) into a continuous probabilistic procedure inspired by [26]. That means, each pixel has to be influenced by all the vertices of the mesh with a corresponding weighted probability. Intuitively, after projection, the closer the pixel to the projected vertex, the higher probability the vertex is influenced. Before projection, the further the distance along the Z(depth) direction, the less the weight it should be imposed to its corresponding probability. To achieve this, one way is to project each triangle t onto the image plane and rasterize all the enclosed pixels to get an rendered image. In this way, each triangle t can only be influenced by those enclosed pixels and their corresponding attribute (color, mask or depth) values if the triangle is visible to the camera. To make it soft we can then apply a *convolution kernel* to *"blur"* the rendered image so that the attribute can be propagated outside of the triangle. Let us denote $\mathcal{A}_t^j$ and Z_t^j as the attribute and Z-value respectively for each enclosed pixel j within triangle t, denote $\mathcal{E}(t)$ as the enclosed pixel set of t, so $j \in \mathcal{E}(t)$, denote S as the whole triangle set. Then by following the similar heuristic formulation in [26], we can aggregate soft rendering results across all the triangles:

$$\mathcal{A}_I(I^i) = \sum_{t \in S} \sum_{j \in \mathcal{E}(t)} w_j^i \mathcal{A}_t^j \tag{4}$$

where $w_j^i = \frac{D_j^i \exp(Z_t^j/\gamma)}{\sum_k D_k^i \exp(Z_t^k/\gamma)}$ and $D_k^i = \mathrm{Sigmoid}(\frac{\|\mathrm{i}-\mathrm{k}\|_2)}{\sigma})(k \in \bigcup_{t=1} \mathcal{E}(t))$, both σ and γ are set $1\mathrm{x}10^{-4}$. Note that, each enclosed pixel attribute value $\mathcal{A}_t^j$ of triangle t is first obtained via per triangle traditional rasterization. The soft rasterization is then simply

implemented as the spatial Gaussian filtering operations with varying kernel sizes to help propagate the attribute values outside the triangle. In practice, it would be both computational intensive and memory inefficient to perform softening and aggregation on the per triangle basis, we therefore approximate on the mesh slices, as show in Fig.3, where we render all the triangle belong to the same depth zone into the same image, then aggregate across different slices. In our current experiments, we empirically slice the mesh along the Z direction into 5 pieces. Mathematically, Equation 4 can be easily implemented as a multi-channel 2D convolution operation, where the kernel size can be varied for different scales of softening. The bigger the kernel size, the broader impact each pixel would get from all the vertices. In practice, we simply stack the same convolution kernel a few times with stride 2 to get a pyramid of rendered attribute image. Then a photo-metric like loss can be applied at each scale of the pyramid between the rendered attribute image and the corresponding ground-truth image (color, mask or depth).

$$L_{ReDA} = \sum_{k} \parallel Pyd(\mathcal{A}_I(I), k) - Pyd(\mathcal{A}_{gt}(I), k) \parallel_1 \quad (5)$$

where Pyd is a function returning the $k-$th scale of the softening version.

5. Free-form Deformation

5.1. Parametric Base Model

Even though parametric base model, like 3DMM [2, 10], has limited modeling capacity, it still provides decent coarse-scale geometry. Compared with methods that learn everything from videos [46, 41, 40, 14], such structure prior would significantly reduce the burden of learning. Recently, more complex technique such as [22] shows better modeling capacity with more 3D scan data. Our method is friendly to any of those previous proposed models. Without loss of generality, assume we use the the following parametric face model to represent the basic face shape $S^0(\alpha, \beta)$:

$$S^0(\alpha, \beta) = \bar{S} + \sum_{k_s=1}^{m^s} \alpha_{k_s} B^s_{k_s} + \sum_{k_e=1}^{m^e} \beta_{k_e} B^e_{k_e} \quad (6)$$

where, $\bar{S} \in R^{3N}$ is the average facial geometry. Matrix $[B^s_1, ..., B^s_{m^s}]$ and $[B^e_1, ..., B^e_{m^e}]$ respectively represent the shape and expression PCA basis learned from high quality face scans [2]. The number of shape and expression basis are represented by m^s and m^e respectively. Given an face image I, one has to figure out the coefficients $[\alpha_1, ..., \alpha_{m^s}]$ and $[\beta_1, ..., \beta_{m^e}]$ to best explain the corresponding face shape. Note that, the reflectance model is defined similarly.

5.2. Shape Correction via Free-form Deformation

Unlike previous work [41] that models the correction in parameter space, we directly model the displacement in vertex space. As indicated in Fig. 2, the network outputs a corrective shape residual Δ_S in parallel with the 3DMM parameters. We use $S^{'}$ to represent the final deformed mesh, hence $S^{'} = S^0 + \Delta_S$. As we discussed before, we expect S^0 to model the coarse geometry which is roughly close to the ground-truth shape, and expect Δ_S to model whatever deformation is needed to fill the gap between S^0 and final correct shape $S^{'}$. As S^0 and $S^{'}$ has natural per-vertex correspondence, so we call the way going from S^0 to $S^{'}$ as free-form deformation.

As-rigid-as-possible Without proper regularization, it is hard to prevent it from deforming to non-sensible shape. Therefore, we impose as-rigid-as-possible (ARAP) constraint to further regularize the deformation. Let cell C_l represent all the triangles centered at vertex p_l, $C^{'}_l$ represent its deformed version; if the deformation is rigid, then there exist a rotation matrix R_l such that

$$p'_l - p'_m = R_l(p_l - p_m), \forall m \in \mathcal{N}(l) \quad (7)$$

for each edge emanating from vertex $p_l(p'_l)$ to its neighbor $p_m(p'_m)$ in the cell, where $\mathcal{N}(l)$ denotes the set of vertex indices connected to vertex p_l. Therefore, in the context of ARAP, we want to minimize to following loss function

$$L(C_l, C'_l) = \sum_{m \in \mathcal{N}(l)} w_{lm} \parallel (p'_l - p'_m) - R_l(p_l - p_m) \parallel \quad (8)$$

when it comes to the whole mesh, the total rigidity can be enhanced by summarizing over all the above loss for each cell, namely

$$L_{ARAP} = \sum_{l}^{n} w_l \sum_{m \in \mathcal{N}(l)} w_{lm} \parallel (p'_l - p'_m) - R_l(p_l - p_m) \parallel \quad (9)$$

where both w_l and w_{lm} are set according to prior work [39]. In addition to the above loss, we also add another smooth term to penalize the rotation difference between two adjacent cells. So our final free-form deformation layer has to minimize following losses (purple rectangle denoted as "FFD_ARAP" in Fig. 2)

$$L(R, \Delta_S) = L_{ARAP} + \lambda \sum_{l}^{n} \sum_{m \in \mathcal{N}(l)} \parallel R_l - R_m \parallel_2 \quad (10)$$

Here, R is the set of all $R_l, l \in [1, ..., n]$. λ is set empirically to 0.001 in all the current experiments. In our implementation, we initialize each R_l as identity matrix, then alternate

between optimizing Δ_S while fixing R and optimizing R [1] while fixing Δ_S. At the end, our entire system can be trained end-to-end by combining L_{DA} and $L(R, \Delta_S)$ together with 2D landmark loss.

6. Experiments and Results

6.1. Datasets

MICC [1] dataset consists of 53 subjects. We use the texture images from frontal pose scan for our fitting experiments. Since the texture images have both left-side and right-side views, we choose the left-side view images for all our experiments. We also follow [13] to crop scans $95mm$ radius around the tip of nose to better evaluate the reconstruction of inner face.

BU-3DFE [50] dataset provides scans of 100 subjects from diverse racial, age and gender groups. Each subject has 25 scans with different expressions. For our experiment, we use the scans and images from neutral faces. Similar to our experiment setup in MICC dataset, the left-side view texture images are used in our experiments.

6.2. Experiment Settings

To directly verify the effectiveness of our proposed method, we conduct experiments with a fitting-based method (Fig.2) Our proposed pipeline can also work for learning-based methods and we would like to experiment on learning in our future work.

Our fitting method adopts SGD optimization using ADAM optimizer and is composed of four stages. In all our experiments, similar to [13, 12], 2D landmark loss is used by default. First, we run landmark detection which includes the invisible line and face parsing on the input image to extract face landmarks and facial masks. Second, we apply landmark loss to optimize rigid pose $\mathrm{P}_{\mathrm{pose}}$ in Equation 4 so that the pose of the template mesh is roughly aligned with our input image. Then we apply our attribute loss (Equation 5) and landmark loss to jointly optimize rigid pose and other model parameters. In the last stage, we exclude the landmark loss if it falls within an empirical thresholds and jointly optimize all the model parameters. In addition to the optimization of model parameters, our free-form deformation is also in the last stage.

To measure the error between ground-truth and our predictions, we first perform iterative closest point (ICP) to automatically find the correspondence between meshes. We then calculate point-to-plane errors which are measured in millimeters. The results for MICC [1] are listed in Table 1 and the results for BU-3DFE [50] are listed in Table 2.

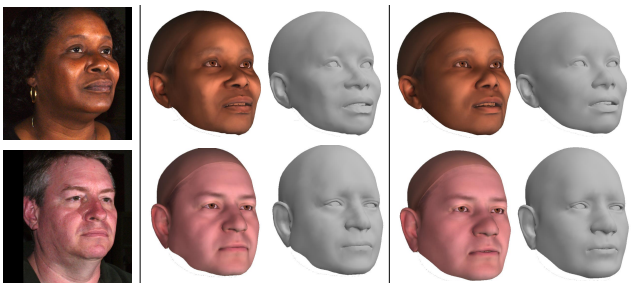

Figure 4: Two examples of showing the effectiveness of free-form deformation. **Left:** Input images; **Middle:** Results with free-form; **Right:** Results without free-from.

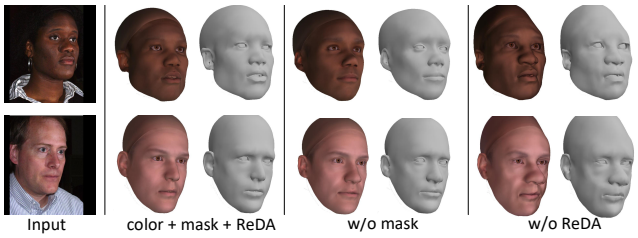

Figure 5: Visual ablation study of our system. As we can see, with the help of both mask and ReDA, the geometries look closer to the input identities.

6.3. Effectiveness of Differentiable Attributes

Similar to works [40, 13, 46] that apply photometric loss by enforcing the color consistency between images and the projected color from 3D shapes. We approximate our 3D shape color by utilizing a PCA texture model trained from 112 scans with lighting approximated by Spherical Harmonics Illumination. For mask attribute image, we first run face parsing model [23] on images to get the ground-truth face parsing masks. To enable facial parsing from 3D shape, we paint UV map as shown in Fig.3 in which each facial region (e.g., eyes, nose, ears and etc.) is painted with discrete color that corresponds to our ground-truth facial mask labels. Since both color and mask attributes have exact correspondence in UV space, we can directly render those attributes as images.

For images with depth information, we by default include the depth attribute in our experiments. To add depth attribute in our pipeline, we render the depth image for both ground-truth mesh and predicted mesh. The rendered depth image can be consumed in the same way as other attribute images by our pipeline in which we compute the loss between our predicted depth image with the ground-truth depth image.

We observe consistent improvements as we combine more attributes in our optimization pipeline. As the results in Table 1 and Table 2 show, by jointly optimize color and

[1] Refer to [39] for detailed derivation of optimizing rotation matrix R

mask attributes, we can achieve 5.1% and 16.1% relative improvement on MICC dataset comparing to optimize color attribute and mask attribute alone and 13.9% and 18.4% on BU-3DFE dataset with the same setting. With additional depth attribute, we can further improve our fitting error by 52.6%, 47.4% and 52.5% comparing to color attribute alone, mask attribute alone color+mask attributes settings respectively. Fig.5 shows the effectiveness of our proposed differentiable attributes in ReDA.

6.4. Effectiveness of ReDA Rasterization

As we have discussed in 4.2, our proposed ReDA Rasterization turns discrete sampling into a continuous probabilistic procedure that a change of one pixel can influence every vertex in a mesh. Our ablation study on MICC dataset Table 1 compares our ReDA Rasterization to traditional Z-buffer Rasterization. Our results shows that such a procedure can effectively reduce our numerical reconstruction error. We observe consistent improvement in reconstruction error on various of attributes constraints comparing to Z-buffer rasterization. ReDA Rasterization reduces our fitting error on MICC by 14.3%, 26.6% and 23.3% with color, mask and color + mask settings respectively relative to our Z-buffer rasterization baseline. Fig.5 also shows the effectiveness by a side-by-side comparison between the ReDA (second column) and the default Z-buffer rasterization (fourth column).

One factor that affects the effectiveness of our ReDA Rasterization is the number of pyramid levels. Our ablation study Table 3 shows that pyramid level of 6 gives the best results. We choose pyramid level of 6 in all our experiments with ReDA Rasterization.

Error(mm)		Attributes		ReDA Rasterization
Mean	SD	Color	Mask	
1.321	0.364	✓		
1.494	0.362		✓	
1.253	0.284	✓	✓	
1.131	0.234	✓		✓
1.097	0.160		✓	✓
0.962	0.146	✓	✓	✓

Table 1: Ablation Studies on MICC Dataset. Z-buffer rasterization is used if ReDA Rasterization is not specified.

6.5. Effectiveness of Free-form Deformation

To better leverage our image attributes, we propose using ARAP free-form deformation 5.2 to ensure that our fitting results are not limited by the capacity of face model. On the top of color, mask and depth attributes we add free-form deformation in the last stage of our fitting and obtained 11.7% relative improvement on BU-3DFE dataset. Fig.4 shows two examples of fitting results between with and without free-form deformation. As we can see, adding the free-form help add more geometry details on the important face regions to better convey the input identity, such as the details around the cheek and mouth.

Error(mm)		Attributes			Free-form
Mean	SD	Color	Mask	Depth	
1.546	0.384	✓			
1.632	0.396		✓		
1.331	0.346	✓	✓		
0.793	0.324	✓		✓	
0.858	0.331		✓	✓	
0.731	0.291	✓	✓	✓	
0.645	0.162	✓	✓	✓	✓

Table 2: Ablation Studies on BU-3DFE Dataset. We by default use ReDA Rasterization. We assume depth groundtruth is given when we use depth attribute

Errors(mm)	Pyramid Level(s)				
	1	2	3	4	6
Mean	2.059	1.940	1.669	1.357	1.331
SD	0.461	0.388	0.374	0.360	0.346

Table 3: We experiment the affect of pyramid level in our ReDA Rasterization on BU-3DFE dataset with both color and mask attributes. Our result shows that more levels of pyramid can help with our fitting results.

6.6. Compare with other Methods

Quantitatively, due to the slight difference in experimental setup, it is hard to do fair comparisons with other state-of-the-arts. Nevertheless, we still compare their fitting errors as a reference. On MICC dataset, GANFit [13] reports historically low fitting error (with mean: 0.94mm, SD:0.106mm) by using a high quality texture (GAN) model trained on a large scale 3D scans. Although the input images are different, our method achieves comparable mean point-to-plane error of 0.962mm with a SD of 0.146mm. On BU-3DFE dataset, we compare with FML [40] which is a learning based method taking multiple RGB images as input. We achieves better result of 1.331mm mean point-to-plane error with SD of 0.346mm comparing to their error of 1.78mm with SD of 0.45mm.

Qualitatively, we compare with FML [40] on VoxCeleb2 [7] and RingNet [37] on CelebA [28]. We show side-by-side comparisons in Fig. 6 and Fig. 7, as can be seen, in many cases, our results fits much closer to their input identities.

Figure 6: Results comparing with RingNet [37]. We use 0.7 alpha blending to show the alignment quality.

Figure 7: Results comparing with FML [40]. We use 0.7 alpha blending to show the alignment quality.

7. Conclusion and Future works

We present a novel framework that integrates multiple differentiable attributes including color, mask and depth into the rasterization and demonstrate their effectiveness for 3D face reconstruction. To fully utilize different image attributes, we propose ReDA rasterization which improves over the traditional Z-buffer render significantly. Lastly, to extend the fitting beyond the capacity of 3D face model, we show that applying free-form deformation could further improve the fitting results. Although our current framework is fitting-based, it can be easily adopt in a learning-based 3D face reconstruction pipeline. Lastly, finding other more differentiable attributes to be used in ReDA is also desirable. We leave all those as our future work.

References

[1] Andrew D Bagdanov, Alberto Del Bimbo, and Iacopo Masi. The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding*, pages 79–80. ACM, 2011. 6

[2] 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*, SIGGRAPH '99, pages 187–194, 1999. 1, 2, 5

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003. 1

[4] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *Int. J. Comput. Vision*, 126(2-4):233–254, Apr. 2018. 2

[5] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics*, 33:1–10, 07 2014. 1, 3

[6] Bindita Chaudhuri, Noranart Vesdapunt, and Baoyuan Wang. Joint face detection and facial motion retargeting for multiple faces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[7] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. 7

[8] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of IEEE Computer Vision and Pattern Recognition Workshop on Analysis and Modeling of Faces and Gestures*, 2019. 1, 2

[9] Bernhard Egger. Semantic morphable models. 2018. 1, 3

[10] Bernhard Egger, William Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhöfer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models – past, present and future. 09 2019. 2, 5

[11] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018. 1, 3

[12] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. In *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*, volume 32, pages 158:1–158:10, November 2013. 2, 6

[13] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: generative adversarial network fitting for high fidelity 3d face reconstruction. *CVPR*, 2019. 2, 3, 6, 7

[14] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3d morphable model regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 5

[15] Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg : Fully convolutional dense shape regression in the-wild rıza. In *CVPR*, 2017. 1, 3

[16] Yoshitaka Ushiku Hiroharu Kato and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[17] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6), Nov. 2017. 1

[18] 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 (CVPR)*, Salt Lake City, UT, 2018. 2

[19] Aaron S. Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[20] L. Jiang, J. Zhang, B. Deng, H. Li, and L. Liu. 3d face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing*, 27(10):4756–4770, Oct 2018. 2

[21] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Trans. Graph.*, 37(4):163:1–163:14, July 2018. 1

[22] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 5

[23] Jinpeng Lin, Hao Yang, Dong Chen, Ming Zeng, Fang Wen, and Lu Yuan. Face parsing with roi tanh-warping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 6

[24] F. Liu, Q. Zhao, x. Liu, and D. Zeng. Joint face alignment and 3d face reconstruction with application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 1

[25] Feng Liu, Ronghang Zhu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[26] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 2, 3, 4

[27] Sifei Liu, Jianping Shi, Ji Liang, and Ming-Hsuan Yang. Face parsing via recurrent propagation. *CoRR*, abs/1708.01936, 2017. 3

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018. 7

[29] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Trans. Graph.*, 37(4):68:1–68:13, July 2018. 1

[30] Matthew Loper and Michael Black. Opendr: An approximate differentiable renderer. In *ECCV'14*, 09 2014. 2

[31] Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7891–7901. 2018. 2

[32] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. Genova, Italy, 2009. IEEE. 2

[33] Albert Pumarola, Antonio Agudo, Aleix Martinez, A. Sanfeliu, and Francesc Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 09 2018. 1

[34] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:986–993 vol. 2, 2005. 1, 2, 3

[35] Joseph Roth, Yiying Tong, and Xiaoming Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[36] Shunsuke Saito, Tianye Li, and Hao Li. Real-time facial segmentation and performance capture from RGB input. *CoRR*, abs/1604.02647, 2016. 3

[37] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 7, 8

[38] William Smith. *The Perspective Face Shape Ambiguity*, pages 299–319. 01 2016. 1

[39] Olga Sorkine-Hornung and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, 2007. 5, 6

[40] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. FML: face model learning from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 6, 7, 8

[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 (CVPR)*, 2018. 1, 2, 5

[42] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2

[43] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016. 1

[44] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[45] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard Medioni. Extreme 3D face reconstruction: Seeing through occlusions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[46] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3D face morphable model. In *In Proceeding of IEEE Computer Vision and Pattern Recognition*, Long Beach, CA, June 2019. 1, 2, 5, 6

[47] Luan Tran and Xiaoming Liu. On learning 3d face morphable model from in-the-wild images. June 2019. 1, 2, 3

[48] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 426–433, 2005. 1

[49] Hongwei Yi, Chen Li, Qiong Cao, Xiaoyong Shen, Sheng Li, Guoping Wang, and Yu-Wing Tai. Mmface: A multi-metric regression network for unconstrained face reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[50] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006. 6

[51] Ronald Yu, Shunsuke Saito, Haoxiang Li, Duygu Ceylan, and Hao Li. Learning dense facial correspondences in unconstrained images. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4733–4742, 2017. 1, 3

[52] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. 2016. 1, 3

[53] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3D face reconstruction, tracking, and applications. *Comput. Graph. Forum*, 37:523–550, 2018. 2