LEARNING 3D MODEL FROM 2D IN-THE-WILD IMAGES

By

Luan Quoc Tran

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science — Doctor of Philosophy

2020

ABSTRACT

LEARNING 3D MODEL FROM 2D IN-THE-WILD IMAGES

By

Luan Quoc Tran

Understanding 3D world is one of computer vision's fundamental problems. While a human has no difficulty understanding the 3D structure of an object upon seeing its 2D image, such a 3D inferring task remains extremely challenging for computer vision systems. To better handle the ambiguity in this inverse problem, one must rely on additional prior assumptions such as constraining faces to lie in a restricted subspace from a 3D model. Conventional 3D models are learned from a set of 3D scans or computer-aided design (CAD) models, and represented by two sets of PCA basis functions. Due to the type and amount of training data, as well as, the linear bases, the representation power of these model can be limited. To address these problems, this thesis proposes an innovative framework to learn a nonlinear 3D model from a large collection of in-the-wild images, without collecting 3D scans. Specifically, given an input image (of a face or an object), a network encoder estimates the projection, lighting, shape and albedo parameters. Two decoders serve as the nonlinear model to map from the shape and albedo parameters to the 3D shape and albedo, respectively. With the projection parameter, lighting, 3D shape, and albedo, a novel analytically-differentiable rendering layer is designed to reconstruct the original input. The entire network is end-to-end trainable with only weak supervision. We demonstrate the superior representation power of our models on different domains (face, generic objects), and their contribution to many other applications on facial analysis and monocular 3D object reconstruction.

This thesis is dedicated to my beautiful wife My Nhat Nguyen, whose encouragement along the way was paramount to making it thus far.

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Xiaoming Liu for the continuous support of my Ph.D study. His desire to see me succeed has pushed me to obtain far more than I could imagine alone. The late nights spent writing papers together, attention to the smallest details, and desire the push the bounds of knowledge have inspired my dedication to excellence. I am deeply indebted for his refinement of my writing and presentation skills.

I would also like to thank the remainder of my committee members, Dr. Arun Ross, Dr. Jiayu Zhou and Dr. Daniel Morris for their valuable insights and contributions along the way.

I am grateful to my Computer Vision Lab members, both present and past, Dr. Joseph Roth, Dr. Xi Yin, Dr. Amin Jourabloo, Dr. Morteza Safdarnejad, Dr. Yousef Atoum, Yaojie Liu, Garrick Brazil, Adam Terwilliger, Joel Stehouwer, Bangjie Yin, Hieu Nguyen, Shengjie Zhu and Masa Hu for the excellent working atmosphere. The willingness to answer any questions and late nights working together causes all of our work to flourish. I will also never forget the memories we have together, from boardgame nights to travel trips, that made my PhD a very pleasant journey.

A word of appreciation to Brenda Hodge, Katherine Trinklein, Steven Smith and Amy King for their administrative assistance.

Finally, I would like to thank my family - my parents and my sister - who have provided me through moral and emotional support in my life. The largest thanks for my beautiful wife My Nhat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction and Contributions

Understanding 3D structure is a long-standing problem with much interest in computer vision. A human has no difficulty understanding the 3D structure of an object upon seeing its 2D image. Even without geometric cues (motion or stereopsis), our visual system can still infer detailed surfaces or plausibly hidden parts. Meanwhile, such a 3D inferring task remains extremely challenging for computer vision systems.

One object in particular, the face, is highly studied, since obtaining a user-specific 3D face surface model is useful for many applications including but not limited to face recognition [6, 102, 185], video editing [47, 155], avatar puppeteering [20, 23, 189] or virtual make-up [48, 83].

Inferring a 3D face mesh from a single photograph is arduous and ill-posed since the image formation process blends multiple components (shape, albedo) as well as environment (lighting) into a single color for each pixel. To better handle the ambiguity, one must rely on additional prior assumptions, such as constraining 3D objects to lie in a restricted subspace, e.g., 3D Morphable Models (3DMM) [13] learned from a small 3D scans collection.

Traditionally, 3DMM is learnt through *supervision* by performing dimension reduction, typically Principal Component Analysis (PCA), on a training set of co-captured 3D face scans and 2D images. To model highly variable 3D face shapes, a large amount of high-quality 3D face scans is required. However, this requirement is expensive to fulfill as acquiring face scans is very laborious, in both data capturing and post-processing stage. The first 3DMM [13] was built from scans of

200 subjects with a similar ethnicity/age group. They were also captured in well-controlled conditions, with only neutral expressions. Hence, it is fragile to large variances in the face identity. The widely used Basel Face Model (BFM) [121] is also built with only 200 subjects in neutral expressions. Lack of expression can be compensated using expression bases from FaceWarehouse [24] or BD-3FE [183], which are learned from the offsets to the neutral pose. After more than a decade, almost all existing models use no more than 300 training scans. Such small training sets are far from adequate to describe the full variability of human faces [19]. Until recently, with a significant effort as well as a novel automated and robust model construction pipeline, Booth *et al*. [19] build the first large-scale 3DMM from scans of $\sim 10,000$ subjects, which is still restricted to the public.

Second, the texture model of 3DMM is normally built with a small number of 2D face images *co-captured* with 3D scans, under well-controlled conditions. Despite there is a considerable improvement of 3D acquisition devices in the last few years, these devices still cannot operate in arbitrary in-the-wild conditions. Therefore, all the current 3D facial datasets have been captured in the laboratory environment. Hence, such models are only learnt to represent the facial texture in similar, rather than in-the-wild, conditions. This substantially limits application scenarios of 3DMM.

Finally, the representation power of 3DMM is limited by not only the size or type of training data but also its *formulation*. The facial variations are nonlinear in nature. E.g., the variations in different facial expressions or poses are nonlinear, which violates the linear assumption of PCA-based models. Thus, a PCA model is unable to interpret facial variations sufficiently well. This is especially true for facial texture. For all current 3DMM models, their low-dimension albedo subspace faces the same problem of lacking facial hair, e.g., beards. To reduce the fitting error, it compensates unexplainable texture by alternating surface normal, or shrinking the face shape [198]. Either way, linear 3DMM-based applications often degrade their performances when

handling out-of-subspace variations.

Given the barrier of 3DMM in its data, supervision and linear bases, this thesis aims to revolutionize the paradigm of learning 3DMM by answering a fundamental question:

> *Whether and how can we learn a nonlinear 3D Morphable Model of face shape and albedo from a set of in-the-wild 2D face images, without collecting 3D face scans?*

If the answer were yes, this would be in sharp contrast to the conventional 3DMM approach, and remedy all aforementioned limitations. Fortunately, we have developed approaches to offer positive answers to this question. With the recent development of deep neural networks, we view that it is the right time to undertake this new paradigm of 3DMM learning. Therefore, the core of this thesis is regarding how to learn this new 3DMM, what is the representation power of the model, and what is the benefit of the model to facial analysis.

1.1 Thesis Contributions

In this thesis, we propose a novel paradigm to *learn a nonlinear 3DMM model from a large in-the-wild 2D face image collection, without acquiring 3D face scans*, by leveraging the power of deep neural networks captures variations and structures in complex face data. The framework is also further extended to generic objects, with substantially larger shape deformation, thanks to a novel representation. In summary, this dissertation makes the following contributions:

◇ To overcome the shortage of annotated 3D data, we develop a framework to jointly learn the 3D model and the model fitting algorithm via weak supervision, by leveraging a large collection of 2D images without 3D scans. Two modules are optimized end-to-end with the objective to reconstruct the input image. This objective allows us to use any photographs for model training

without any 3D labels.

◇ Different from previous methods that focus on modeling only 3D shape, the proposed nonlinear 3DMM fully models shape, albedo and lighting, which enables us to train the model in weak supervision fashion.

◇ By using neural networks to represent all model components, our model can better model nonlinear shape/albedo variations. Hence our model has greater representation power than its traditional linear counterpart.

◇ In realization that the strong regularization and global-based modeling are the roadblocks to achieve high-fidelity 3DMM model, we propose to relax regularization by using proxies and propose a global-local network architecture.

◇ To extend the learning framework to generic objects which usually has large shape deformation as well as inconsistent shape topology, we propose a novel representation, colored occupancy field, in which each 3D spatial point is classified as inside/outside the 3D shape as well as assigned with an albedo color.

1.2 Thesis Organization

The rest of this dissertation is organized as follows. Chapter 2 gives more background introduction and reviews related work on 3D reconstruction. Chapter 3 develops the learning framework on nonlinear 3DMM. Chapter 4 improves the model in both learning objective and architecture. Chapter 5 presents the extension of the framework to generic objects with a novel representation, colored occupancy field. Chapter 6 concludes this dissertation.

Chapter 2

Background and Related Work

Now that a basic understanding of the problem is known, I will present some background information and related work necessary for fully understanding this thesis.

2.1 3D Morphable Model

The 3D Morphable Model (3DMM) [13] and its 2D counterpart, Active Appearance Model [37, 94, 91], provide parametric models for synthesizing faces, where faces are modeled using two components: shape and albedo (skin reflectant).

Blanz and Vetter [13] propose the first generic 3D face model learned from scan data. They define a linear subspace to represent shape and albedo using principal component analysis (PCA) and show how to fit the model to data. The 3D face space can represented with PCA as:

$$\mathbf{S} = \bar{\mathbf{S}} + \mathbf{G}\alpha, \tag{2.1}$$

where $\mathbf{S} \in \mathbb{R}^{3Q}$ is a 3D face mesh with Q vertices, $\bar{\mathbf{S}} \in \mathbb{R}^{3Q}$ is the mean shape, $\alpha \in \mathbb{R}^{ls}$ is the shape parameter corresponding to a 3D shape bases $\mathbf{G}$. The shape bases can be further split into $\mathbf{G} = [\mathbf{G}_{id}, \mathbf{G}_{exp}]$, where $\mathbf{G}_{id}$ is trained from 3D scans with neutral expression, and $\mathbf{G}_{exp}$ is from the offsets between expression and neutral scans.

The albedo of the face $\mathbf{A} \in \mathbb{R}^{3Q}$ is defined within the mean shape $\bar{\mathbf{S}}$, which describes the R,

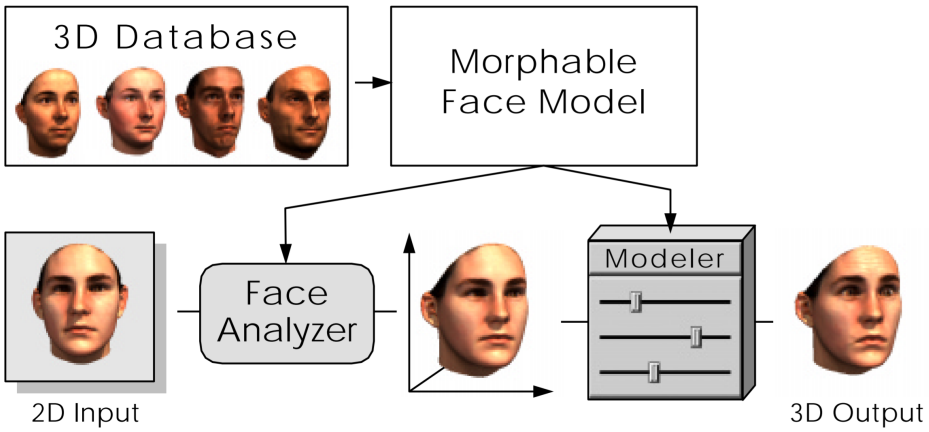


Figure 2.1: The visual abstract of the seminal work by Blanz and Vetter [13]. It proposes a statistical model for faces to perform 3D reconstruction from 2D images and a parametric face space which enables controlled manipulation

G, B colors of Q corresponding vertices. $\mathbf{A}$ is also formulated as a linear combination of basis functions:

$$\mathbf{A} = \bar{\mathbf{A}} + \mathbf{R}\beta, \tag{2.2}$$

where $\bar{\mathbf{A}}$ is the mean albedo, $\mathbf{R}$ is the albedo bases, and $\beta \in \mathbb{R}^{l_T}$ is the albedo parameter.

The 3DMM can be used to synthesize novel views of the face. Firstly, a 3D face is projected onto the image plane with the weak perspective projection model:

$$\mathbf{V} = \mathbf{R} * \mathbf{S}, \tag{2.3}$$

$$g(\mathbf{S}, \mathbf{m}) = \mathbf{V}^{2\mathrm{D}} = f * \mathbf{Pr} * \mathbf{V} + \mathbf{t}_{2d} = M(\mathbf{m}) * \begin{bmatrix} \mathbf{S} \\ \mathbf{1} \end{bmatrix}, \tag{2.4}$$

where $g(\mathbf{S}, \mathbf{m})$ is the projection function leading to the 2D positions $\mathbf{V}^{2\mathrm{D}}$ of 3D rotated vertices $\mathbf{V}$, f is the scale factor, $\mathbf{Pr} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is the orthographic projection matrix, $\mathbf{R}$ is the rotation matrix constructed from three rotation angles (pitch, yaw, roll), and $\mathbf{t}_{2d}$ is the translation vector. While the project matrix M is of the size of 2×4, it has six degrees of freedom, which is parameterized by a

6-dim vector **m**. Then, the 2D image is rendered using texture and an illumination model such as Phong reflection model [122] or Spherical Harmonics [125].

Since Blanz and Vetter's seminal work [13], there has been a large amount of effort on improving 3DMM modeling mechanism. In [13], the dense correspondence between facial mesh is solved with a regularised form of optical flow. However, this technique is only effective in a constrained setting, where subjects share similar ethnicities and ages. To overcome this challenge, Patel and Smith [120] employ a Thin Plate Splines (TPS) warp [16] to register the meshes into a common reference frame. Alternatively, Paysan *et al*. [121] use a Nonrigid Iterative Closest Point [7] to directly align 3D scans. In a different direction, Amberg *et al*. [6] extended Blanz and Vetter's PCA-based model to emotive facial shapes by adopting an additional PCA modeling of the residuals from the neutral pose. This results in a single linear model of both identity and expression variation of 3D facial shape. Vlasic *et al*. [166] use a multilinear model to represent the combined effect of identity and expression variation on the facial shape. Later, Bolkart and Wuhrer [15] show how such a multilinear model can be estimated directly from the 3D scans using a joint optimization over the model parameters and groupwise registration of 3D scans

2.2 Improving Linear 3DMM

With PCA bases, the statistical distribution underlying 3DMM is Gaussian. Koppen *et al*. [77] argue that single-mode Gaussian can't well represent real-world distribution. They introduce the Gaussian Mixture 3DMM that models the global population as a mixture of Gaussian subpopulations, each with its own mean, but shared covariance. Booth *et al*. [17, 18] aim to improve texture of 3DMM to go beyond controlled settings by learning "in-the-wild" feature-based texture model. On another direction, Tran *et al*. [158] learn to regress robust and discriminative 3DMM represen-

tation, by leveraging multiple images from the same subject. However, all works are still based on statistical PCA bases. Duong *et al*. [112] address the problem of linearity in face modeling by using Deep Boltzmann Machines. However, they only work with 2D face and sparse landmarks; and hence cannot handle faces with large-pose variations or occlusion well. Concurrent to our work, Tewari *et al*. [152] learn a (potentially non-linear) corrective model on top of a linear model. The final model is a summation of the base linear model and the learned corrective model, which contrasts to our unified model. Furthermore, our model has an advantage of using 2D representation of both shape and albedo, which maintains spatial relations between vertices and leverages CNN power for image synthesis. Finally, thanks for our novel rendering layer, we are able to employ perceptual, adversarial loss to improve the reconstruction quality.

2.3 2D Face Alignment

2D Face Alignment [172, 90] can be cast as a regression problem where 2D landmark locations are regressed directly [42]. For large-pose or occluded faces, strong priors of 3D model face shape have been shown to be beneficial [67]. Hence, there is increasing attention in conducting face alignment by fitting a 3D face model to a single 2D image [68, 193, 195, 86, 106, 71, 69]. Among the prior works, iterative approaches with cascade of regressors tend to be preferred. At each cascade, there is a single [165, 67] or even two regressors [175] used to improve its prediction. Recently, Jourabloo and Liu [71] propose a CNN architecture that enables the end-to-end training ability of their network cascade. Contrasted to aforementioned works that use a fixed 3DMM model, our model and model fitting are learned jointly. This results in a more powerful model: a single-pass encoder, which is learned jointly with the model, achieves state-of-the-art face alignment performance on different benchmark datasets.

2.4 3D Face Reconstruction

Face reconstruction creates a 3D face model from an image collection [130, 131] or even with a single image [128, 139]. This long-standing problem draws a lot of interest because of its wide applications. 3DMM also demonstrates its strength in face reconstruction, especially in the monocular case. This problem is a highly under-constrained, as with a single image, present information about the surface is limited. Hence, 3D face reconstruction must rely on prior knowledge like 3DMM [132]. Statistical PCA linear 3DMM is the most commonly used approach. Besides 3DMM fitting methods [14, 55, 190, 43, 153, 88], recently, Richardson *et al*. [129] design a refinement network that adds facial details on top of the 3DMM-based geometry. However, this approach can only learn 2.5D depth map, which loses the correspondence property of 3DMM. The follow up work by Sela *et al*. [139] try to overcome this weakness by learning a correspondence map. Despite having some impressive reconstruction results, both these methods are limited by training data synthesized from the linear 3DMM model. Hence, they fail to handle out-of-subspace variations, e.g., facial hair.

2.5 3D Object Modeling and Reconstruction

Recently, autoencoder has been widely used for 3D object modeling [65, 126, 85, 8, 38, 146] due to its efficient feature representation. These methods can be naturally applied to single-image 3D reconstruction. The reconstruction process encodes the input image with deep convolutional networks, and then uses the trained decoder to reconstruct the corresponding 3D shapes from the shape latent vectors. However, most of these methods suffer from the domain mismatch issue since the models are trained on *synthetic* data.

Another related direction, e.g., MarrNet [173] and ShapeHD [174], is to develop a two-step

pipeline. They first recover 2.5D sketches (depth and normal maps), from which a voxelized 3D shape can be further inferred. VON [192] method also benefits from this two-step process for realistic image synthesis. However, despite the use of 2.5D sketches can relax the burden on domain transfer and constrain the reconstructed 3D shape to be consistent with 2D observations, they still have two limitations: 1) Even with high-resolution voxel, they are far from producing visually compelling shapes; 2) They do not learn disentangled and interpretable latent vectors that allow image manipulation under different conditions (e.g., pose and lighting).

Chapter 3

Learning 3D Face Morphable Model from In-the-wild Images

3.1 Introduction

The 3D Morphable Model (3DMM) is a statistical model of 3D facial shape and texture in a space where there are explicit correspondences [13]. The morphable model framework provides two key benefits: first, a point-to-point correspondence between the reconstruction and all other models, enabling "morphing", and second, modeling underlying transformations between types of faces (male to female, neutral to smile, etc.). 3DMM has been widely applied in numerous areas including computer vision [13, 186, 159], computer graphics [5, 141, 154, 155], human behavioral analysis [6, 185] and craniofacial surgery [145].

Given the barrier of 3DMM in its data, supervision and linear bases, we propose a novel paradigm to *learn a nonlinear* 3*DMM model from a large in-the-wild* 2*D face image collection, without acquiring* 3*D face scans*. As shown in Fig. A1, starting with an observation that the linear

This chapter is adapted from following publications:
[1] Luan Tran and Xiaoming Liu, "Nonlinear 3D Face Morphable Model" in CVPR, 2018.
[2] Luan Tran and Xiaoming Liu, "On Learning 3D Face Morphable Model From In-the-wild images" in TPAMI, 2019.

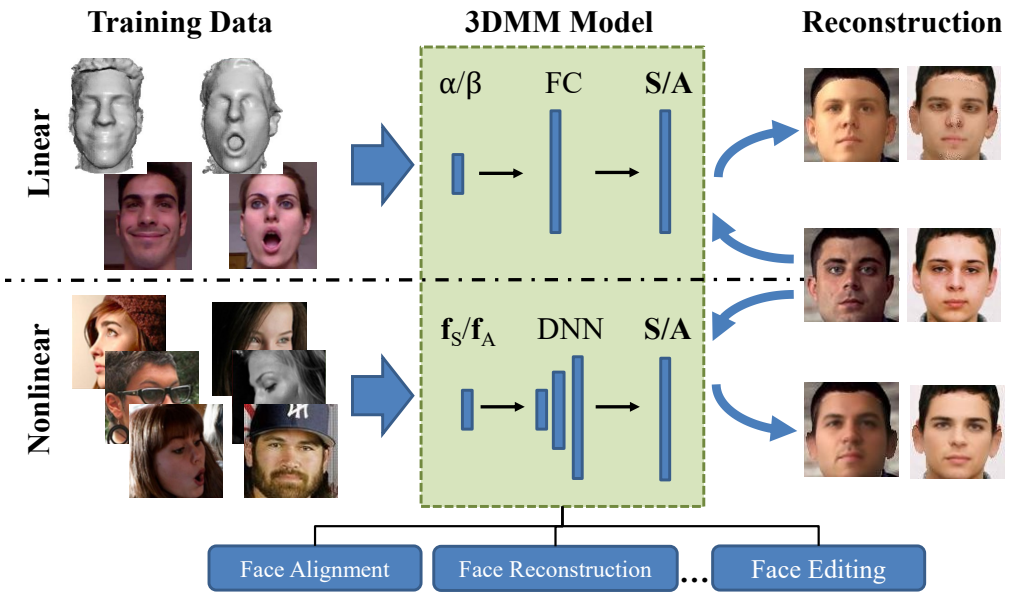


Figure 3.1: Conventional 3DMM employs linear bases models for shape/albedo, which are trained with 3D face scans and associated controlled 2D images. We propose a nonlinear 3DMM to model shape/albedo via deep neural networks (DNNs). It can be trained from in-the-wild face images without 3D scans, and also better reconstruct the original images due to the inherent nonlinearity.

3DMM formulation is equivalent to a single layer network, using a deep network architecture naturally increases the model capacity. Hence, we utilize two convolution neural network decoders, instead of two PCA spaces, as the shape and albedo model components, respectively. Each decoder will take a shape or albedo parameter as input and output the dense 3D face mesh or a face skin reflectance. These two decoders are essentially the nonlinear 3DMM.

Further, we learn the fitting algorithm to our nonlinear 3DMM, which is formulated as a CNN encoder. The encoder network takes a face image as input and generates the shape and albedo parameters, from which two decoders estimate shape and albedo.

The 3D face and albedo would *perfectly* reconstruct the input face, if the fitting algorithm and 3DMM are well learnt. Therefore, we design a differentiable rendering layer to generate a reconstructed face by fusing the 3D face, albedo, lighting, and the camera projection parameters estimated by the encoder. Finally, the end-to-end learning scheme is constructed where the encoder

and two decoders are learnt jointly to minimize the difference between the reconstructed face and the input face. Jointly learning the 3DMM and the model fitting encoder allows us to leverage the large collection of *in-the-wild* 2D images without relying on 3D scans. We show significantly improved shape and facial texture representation power over the linear 3DMM. Consequently, this also benefits other tasks such as 2D face alignment, 3D reconstruction, and face editing.

In summary, this chapter makes the following main contributions.

◇ We learn a *nonlinear* 3DMM model, fully models shape, albedo and lighting, that has greater representation power than its traditional linear counterpart.

◇ Both shape and albedo are represented as 2D images, which help to maintain spatial relations as well as leverage CNN power in image synthesis.

◇ We jointly learn the model and the model fitting algorithm via *weak supervision*, by leveraging a large collection of 2D images without 3D scans. The novel rendering layer enables the end-to-end training.

◇ The new 3DMM further improves performance in related facial analysis tasks: face alignment, face reconstruction.

3.2 The Proposed Nonlinear 3DMM

3.2.1 Nonlinear 3DMM

As mentioned in Sec. 3.1, the linear 3DMM has the problems such as requiring 3D face scans for supervised learning, unable to leverage massive in-the-wild face images for learning, and the limited representation power due to the linear bases. We propose to learn a nonlinear 3DMM model using only large-scale in-the-wild 2D face images.

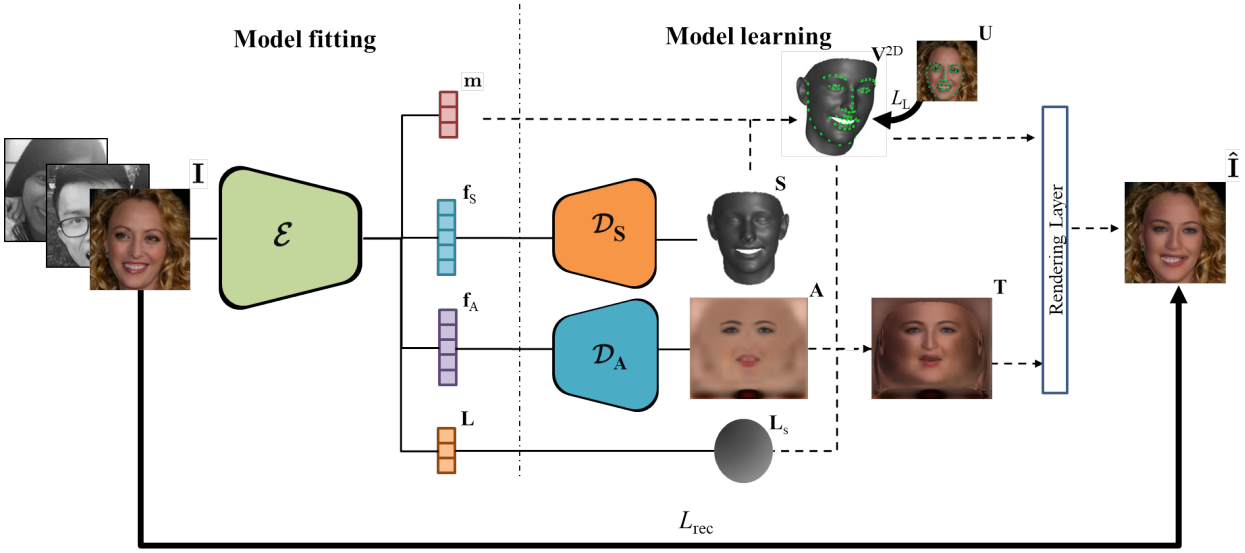


Figure 3.2: Jointly learning a nonlinear 3DMM and its fitting algorithm from unconstrained 2D in-the-wild face image collection, in a weakly supervised fashion. $\mathbf{L}_S$ is a visualization of shading on a sphere with lighting parameters $\mathbf{L}$.

3.2.1.1 Problem Formulation

In linear 3DMM (Sec 2.1), the factorization of each of components (shape, albedo) can be seen as a matrix multiplication between coefficients and bases. From a neural network's perspective, this can be viewed as a shallow network with only *one fully connected layer* and no activation function. Naturally, to increase the model's representation power, the shallow network can be extended to a deep architecture. In this work, we design a novel learning scheme to joint learn a deep 3DMM model and its inference (or fitting) algorithm.

Specifically, as shown in Fig. 3.2, we use two deep networks to decode the shape, albedo parameters into the 3D facial shape and albedo respectively. To make the framework end-to-end trainable, these parameters are estimated by an encoder network, which is essentially the fitting algorithm of our 3DMM. Three deep networks join forces for the ultimate goal of reconstructing the input face image, with the assistant of a physically-based rendering layer. Fig. 3.2 visualizes the architecture of the proposed framework. Each component will be present in following sections.

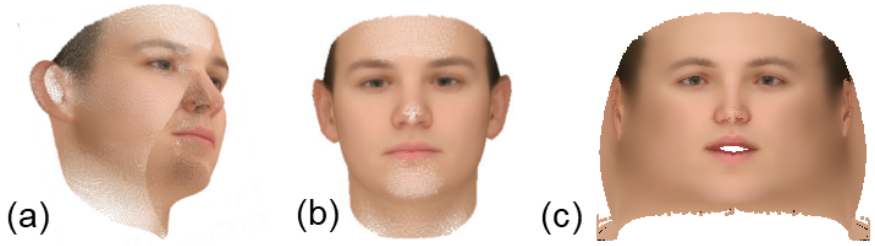


Figure 3.3: Three albedo representations. (a) Albedo value per vertex, (b) Albedo as a 2D frontal face, (c) UV space 2D unwarped albedo.

Formally, given a set of K 2D face images $\{\mathbf{I}_i\}_{i=1}^{K}$, we aim to learn an encoder $\mathcal{E}$: $\mathbf{I}\rightarrow\mathbf{P},\mathbf{L},\mathbf{f}_S,\mathbf{f}_A$ that estimates the projection matrix $\mathbf{P}$, lighting parameter $\mathbf{L}$, shape parameters $\mathbf{f}_S \in \mathbb{R}^{l_S}$, and albedo parameter $\mathbf{f}_A \in \mathbb{R}^{l_A}$, a 3D shape decoder $\mathcal{D}_S$: $\mathbf{f}_S\rightarrow\mathbf{S}$ that decodes the shape parameter to a 3D shape $\mathbf{S} \in \mathbb{R}^{3Q}$, and an albedo decoder $\mathcal{D}_A$: $\mathbf{f}_A\rightarrow\mathbf{A}$ that decodes the albedo parameter to a realistic albedo $\mathbf{A} \in \mathbb{R}^{3Q}$, with the objective that the rendered image with $\mathbf{P}$, $\mathbf{L}$, $\mathbf{S}$, and $\mathbf{A}$ can well approximate the original image. Mathematically, the objective function is:

$$\underset{\mathcal{E},\mathcal{D}_S,\mathcal{D}_A}{\arg\min} \sum_{i=1}^{K} \left\|\hat{\mathbf{I}}_i - \mathbf{I}_i\right\|_1, \tag{3.1}$$

$$\hat{\mathbf{I}} = \mathcal{R}\left(\mathcal{E}_P(\mathbf{I}), \mathcal{E}_L(\mathbf{I}), \mathcal{D}_S(\mathcal{E}_S(\mathbf{I})), \mathcal{D}_A(\mathcal{E}_A(\mathbf{I}))\right),$$

where $\mathcal{R}(\mathbf{P},\mathbf{L},\mathbf{S},\mathbf{A})$ is the rendering layer (Sec. 5.1.2).

3.2.1.2 Albedo & Shape Representation

Fig. 3.3 illustrates three possible albedo representations. In traditional 3DMM, albedo is defined per vertex (Fig. 3.3(a)). This representation is also adopted in recent work such as [153, 152]. There is an albedo intensity value corresponding to each vertex in the face mesh. Despite widely used, this representation has its limitations. Since 3D vertices are not defined on a 2D grid, this representation is mostly parameterized as a vector, which not only loses the spatial relation of its

vertices, but also prevents it to leverage the convenience of deploying CNN on 2D albedo. In contrast, given the rapid progress in image synthesis, it is desirable to choose a 2D image, e.g., a frontal-view face image in Fig. 3.3(b), as an albedo representation. However, frontal faces contain little information of two sides, which would lose many albedo information for side-view faces.

In light of these consideration, we use an unwrapped 2D texture as our texture representation (Fig. 3.3(c)). Specifically, each 3D vertex $\mathbf{v}$ is projected onto the UV space using cylindrical unwarp. Assuming that the face mesh has the top pointing up the y axis, the projection of $\mathbf{v} = (x, y, z)$ onto the UV space $\mathbf{v}^{\text{uv}} = (u, v)$ is computed as:

$$v \rightarrow \alpha_1 . \arctan\left(\frac{x}{z}\right) + \beta_1, \quad u \rightarrow \alpha_2 . y + \beta_2, \tag{3.2}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2$ are constant scale and translation scalars to place the unwrapped face into the image boundaries. Here, per-vertex albedo $\mathbf{A} \in \mathbb{R}^{3Q}$ could be easily computed by sampling from its UV space counterpart $\mathbf{A}^{\text{uv}} \in \mathbb{R}^{U \times V}$:

$$\mathbf{A}(\mathbf{v}) = \mathbf{A}^{\text{uv}}(\mathbf{v}^{\text{uv}}). \tag{3.3}$$

Usually, it involves sub-pixel sampling via bilinear interpolation:

$$\mathbf{A}(\mathbf{v}) = \sum_{\substack{u' \in \{\lfloor u \rfloor, \lceil u \rceil\} \\ v' \in \{\lfloor v \rfloor, \lceil v \rceil\}}} \mathbf{A}^{\text{uv}}(u', v')(1 - |u - u'|)(1 - |v - v'|), \tag{3.4}$$

where $\mathbf{v}^{\text{uv}} = (u, v)$ is the UV space projection of $\mathbf{v}$ via Eqn. 3.2.

Albedo information is naturally expressed in the UV space but spatial data can be embedded in the same space as well. Here, a 3D facial mesh can be represented as a 2D image with three

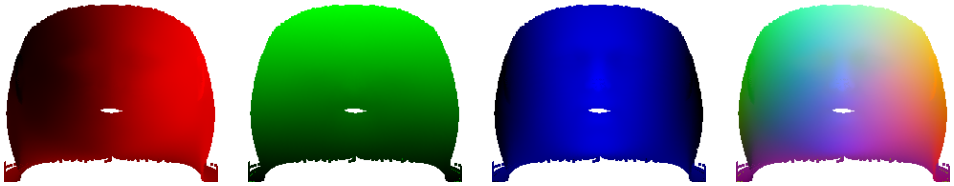


Figure 3.4: UV space shape representation. From left to right: individual channels for x, y and z spatial dimension and final combined shape image.

Table 3.1: The architectures of E, $\mathcal{D}_A$ and $\mathcal{D}_S$ networks.

E			$\mathcal{D}_A/\mathcal{D}_S$		
Layer	Filter/Stride	Output Size	Layer	Filter/Stride	Output Size
			FC		6×7×320
Conv11	7×7/2	112×112×32	FConv52	3×3/2	12×14×160
Conv12	3×3/1	112×112×64	FConv51	3×3/1	12×14×256
Conv21	3×3/2	56×56×64	FConv43	3×3/2	24×28×256
Conv22	3×3/1	56×56×64	FConv42	3×3/1	24×28×128
Conv23	3×3/1	56×56×128	FConv41	3×3/1	24×28×192
Conv31	3×3/2	28×28×128	FConv33	3×3/2	48×56×192
Conv32	3×3/1	28×28×96	FConv32	3×3/1	48×56×96
Conv33	3×3/1	28×28×192	FConv31	3×3/1	48×56×128
Conv41	3×3/2	14×14×192	FConv23	3×3/2	96×112×128
Conv42	3×3/1	14×14×128	FConv22	3×3/1	96×112×64
Conv43	3×3/1	14×14×256	FConv21	3×3/1	96×112×64
Conv51	3×3/2	7×7×256	FConv13	3×3/2	192×224×64
Conv52	3×3/1	7×7×160	FConv12	3×3/1	192×224×32
Conv53	3×3/1	7×7×(l_S+l_A+64)	FConv11	3×3/1	192×224×3
AvgPool	7×7/1	1×1×(l_S+l_A+64)			
$\mathbf{FC_m}$	64×6	6			
$\mathbf{FC_L}$	64×27	27			

channels, one for each spatial dimension x, y and z. Fig 3.4 gives an example of this UV space shape representation $\mathbf{S}^{\text{uv}} \in \mathbb{R}^{U\times V}$.

Representing 3D face shape in UV space allow us to use a CNN for shape decoder $\mathcal{D}_S$ instead of using a multi-layer perceptron (MLP) as in our preliminary version [160]. Avoiding using wide

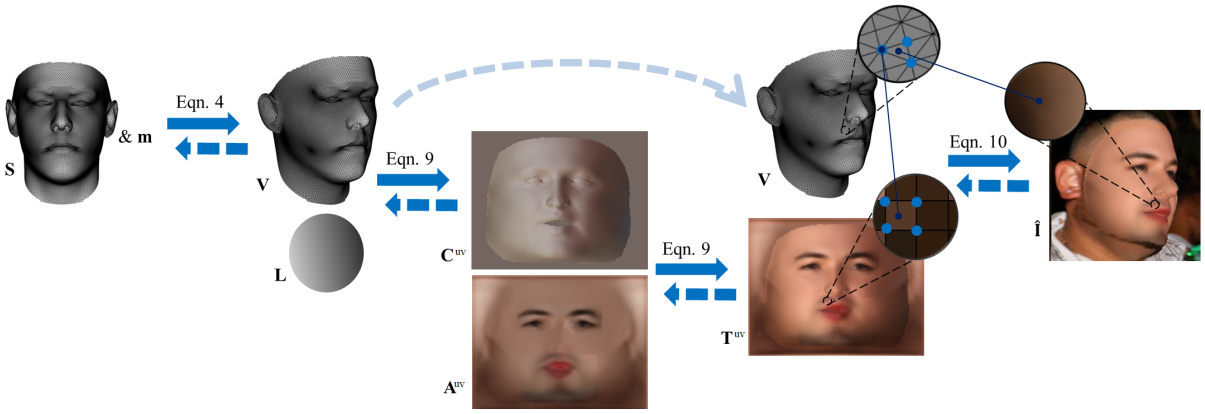


Figure 3.5: Forward and backward pass of the rendering layer.

fully-connected layers allow us to use deeper network for $\mathcal{D}_S$, potentially model more complex shape variations. This results in better fitting results as being demonstrated in our experiment (Sec. 3.3.1.2).

The reference shape used has the mouth open. This change helps the network to avoid learning a large gradient near the two lips' borders in the vertical direction when the mouth is open.

To regress these 2D representation of shape and albedo, we can employ CNNs as shape and albedo networks respectively. Specifically, $\mathcal{D}_S$, $\mathcal{D}_A$ are CNN constructed by multiple fractionally-strided convolution layers. After each convolution is batchnorm and eLU activation, except the last convolution layers of encoder and decoders. The output layer has a *tanh* activation to constraint the output to be in the range of $[-1,1]$. The detailed network architecture is presented in Tab. 3.1.

3.2.1.3 In-Network Physically-Based Face Rendering

To reconstruct a face image from the albedo $\mathbf{A}$, shape $\mathbf{S}$, lighting parameter $\mathbf{L}$, and projection parameter $\mathbf{m}$, we define a rendering layer $\mathcal{R}(\mathbf{m},\mathbf{L},\mathbf{S},\mathbf{A})$ to render a face image from the above parameters. This is accomplished in three steps, as shown in Fig. 3.5. Firstly, the facial texture is computed using the albedo $\mathbf{A}$ and the surface normal map of the rotated shape $N(\mathbf{V}) = N(\mathbf{P},\mathbf{S})$. Here, following [169], we assume distant illumination and a purely *Lambertian* surface reflectance.

Hence the incoming radiance can be approximated using spherical harmonics (SH) basis functions $H_b : \mathbb{R}^3 \to \mathbb{R}$, and controlled by coefficients $\mathbf{L}$. Specifically, the texture in UV space $\mathbf{T}^{\text{uv}} \in \mathbb{R}^{U \times V}$ is composed of albedo $\mathbf{A}^{\text{uv}}$ and shading $\mathbf{C}^{\text{uv}}$:

$$\mathbf{T}^{\text{uv}} = \mathbf{A}^{\text{uv}} \odot \mathbf{C}^{\text{uv}} = \mathbf{A}^{\text{uv}} \odot \sum_{b=1}^{B^2} L_b H_b(N(\mathbf{m}, \mathbf{S}^{\text{uv}})), \tag{3.5}$$

where B is the number of spherical harmonics bands. We use $B = 3$, which leads to $B^2 = 9$ coefficients in $\mathbf{L}$ for each of three color channels. Secondly, the 3D shape/mesh $\mathbf{S}$ is projected to the image plane via Eqn. 2.4. Finally, the 3D mesh is then rendered using a Z-buffer renderer, where each pixel is associated with a single triangle of the mesh,

$$\begin{aligned} \hat{\mathbf{I}}(m,n) &= \mathcal{R}(\mathbf{P}, \mathbf{L}, \mathbf{S}^{\text{uv}}, \mathbf{A}^{\text{uv}})_{m,n} \\ &= \mathbf{T}^{\text{uv}}\Big(\sum_{\mathbf{v}_i \in \Phi^{\text{uv}}(g,m,n)} \lambda_i \mathbf{v}_i\Big), \end{aligned} \tag{3.6}$$

where $\Phi(g,m,n) = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is an operation returning three vertices of the triangle that encloses the pixel (m,n) after projection g; $\Phi^{\text{uv}}(g,m,n)$ is the same operation with resultant vertices mapped into the referenced UV space using Eqn. 3.2. In order to handle occlusions, when a single pixel resides in more than one triangle, the triangle that is closest to the image plane is selected. The final location of each pixel is determined by interpolating the location of three vertices via barycentric coordinates $\{\lambda_i\}_{i=1}^3$.

There are alternative designs to our rendering layer. If the texture representation is defined per vertex, as in Fig. 3.3(a), one may warp the input image $\mathbf{I}_i$ onto the vertex space of the 3D shape $\mathbf{S}$, whose distance to the per-vertex texture representation can form a reconstruction loss. This design is adopted by the recent work of [153, 152]. In comparison, our rendered image is defined on a

2D grid while the alternative is on top of the 3D mesh. As a result, our rendered image can enjoy the convenience of applying the perceptual loss or adversarial loss, which is shown to be critical in improving the quality of synthetic texture. Another design for rendering layer is image warping based on the spline interpolation, as in [36]. However, this warping is continuous: every pixel in the input will map to the output. Hence this warping operation fails in the occluded region. As a result, Cole *et al*. [36] limit their scope to only synthesizing frontal-view faces by warping from normalized faces.

The CUDA implementation of our rendering layer is publicly available at `https://github.com/tranluan/Nonlinear_Face_3DMM`.

3.2.1.4 Occlusion-aware Rendering

Very often, in-the-wild faces are occluded by glasses, hair, hands, etc. Trying to reconstruct abnormal occluded regions could make the model learning more difficult or result in an model with external occlusion baked in. Hence, we propose to use a segmentation mask to exclude occluded regions in the rendering pipeline:

$$\hat{\mathbf{I}} \leftarrow \hat{\mathbf{I}} \odot \mathbf{M} + \mathbf{I} \odot (1 - \mathbf{M}). \tag{3.7}$$

As a result, these occluded regions won't affect our optimization process. The foreground mask $\mathbf{M}$ is estimated using the segmentation method given by Nirkin*et al*. [113]. Examples of segmentation masks and rendering results can be found in Fig. 3.6.

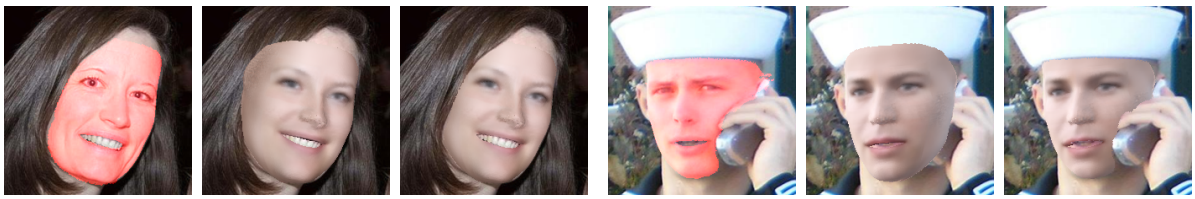


Figure 3.6: Rendering with segmentation masks. Left to right: segmentation results, naive rendering, occulusion-aware rendering.

3.2.1.5 Model Learning

The entire network is end-to-end trained to reconstruct the input images, with the loss function:

$$\mathcal{L} = \mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_{\text{lan}}\mathcal{L}_{\text{lan}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}, \tag{3.8}$$

where the reconstruction loss L_{rec} enforces the rendered image $\hat{\mathbf{I}}$ to be similar to the input $\mathbf{I}$, the landmark loss L_L enforces geometry constraint, and the regularization loss $\mathcal{L}_{\text{rec}}$ encourages plausible solutions.

Reconstruction Loss. The main objective of the network is to reconstruct the original face via disentangle representation. Hence, we enforce the reconstructed image to be similar to the original input image:

$$\mathcal{L}^{i}_{\text{rec}}(\hat{\mathbf{I}}, \mathbf{I}) = \frac{1}{|\mathcal{V}|}\sum_{q\in\mathcal{V}} ||\hat{\mathbf{I}}(q) - \mathbf{I}(q)||_2 \tag{3.9}$$

where $\mathcal{V}$ is the set of all pixels in the images covered by the estimated face mesh. There are different norms can be used to measure the closeness. To better handle outliers, we adopt the robust $l_{2,1}$, where the distance in the 3D RGB color space is based on l_2 and the summation over all pixels enforces sparsity based on l_1-norm [155, 156].

To improve from blurry reconstruction results of l_p losses, in our preliminary work [160], thanks for our rendering layer, we employ adversarial loss to enhance the image realism. However,

adversarial objective only encourage the reconstruction to be close to the real image distribution but not necessary the input image. Also, it's known to be not stable to optimize. Here, we propose to use a perceptual loss to enforce the closeness between images $\hat{\mathbf{I}}$ and $\mathbf{I}$, which overcomes both of adversarial loss's weaknesses. Besides encouraging the pixels of the output image $\hat{\mathbf{I}}$ to exactly match the pixels of the input $\mathbf{I}$, we encourage them to have similar feature representations as computed by the loss network φ.

$$\mathcal{L}^{f}_{\text{rec}}(\hat{\mathbf{I}},\mathbf{I}) = \frac{1}{|\mathcal{C}|}\sum_{j\in\mathcal{C}}\frac{1}{W_j H_j C_j}||\varphi_j(\hat{\mathbf{I}}) - \varphi_j(\mathbf{I})||_2^2. \tag{3.10}$$

We choose VGG-Face[118] as our φ to leverage its face-related features and also because of simplicity. The loss is summed over $\mathcal{C}$, a subset of layers of φ. Here $\varphi_j(\mathbf{I})$ is the activations of the j-th layer of φ when processing the image $\mathbf{I}$ with dimension $W_j \times H_j \times C_j$. This feature reconstruction loss is one of perceptual losses widely used in different image processing tasks [66].

The final reconstruction loss is a weighted sum of two terms:

$$\mathcal{L}_{\text{rec}}(\hat{\mathbf{I}},\mathbf{I}) = \mathcal{L}^{i}_{\text{rec}}(\hat{\mathbf{I}},\mathbf{I}) + \lambda_f \mathcal{L}^{f}_{\text{rec}}(\hat{\mathbf{I}},\mathbf{I}). \tag{3.11}$$

Sparse Landmark Alignment. To help achieving better model fitting, which in turn helps to improve the model learning itself, we employ the landmark alignment loss, measuring Euclidean distance between estimated and groundtruth landmarks, as an auxiliary task,

$$\mathcal{L}_{\text{lan}} = \left\| \mathbf{P} * \begin{bmatrix} \mathbf{S}(:,\mathbf{d}) \\ \mathbf{1} \end{bmatrix} - \mathbf{U} \right\|_2^2, \tag{3.12}$$

where $\mathbf{U} \in \mathbb{R}^{2\times 68}$ is the manually labeled 2D landmark locations, $\mathbf{d}$ is a constant 68-dim vector

storing the indexes of 68 3D vertices corresponding to the labeled 2D landmarks. Different from traditional face alignment work where the shape bases are fixed, our work jointly learns the bases functions (i.e., the shape decoder $\mathcal{D}_S$) as well. Minimizing the landmark loss while updating $\mathcal{D}_S$ only moves a tiny subsets of vertices. If the shape $\mathbf{S}$ is represented as a vector and $\mathcal{D}_S$ is a MLP consisting of fully connected layers, vertices are independent. Hence L_{L} only adjusts 68 vertices. In case $\mathbf{S}$ is represented in the UV space and $\mathcal{D}_S$ is a CNN, local neighbor region could also be modified. In both cases, updating $\mathcal{D}_S$ based on L_{L} only moves a subsets of vertices, which could lead to implausible shapes. Hence, when optimizing the landmark loss, we fix the decoder $\mathcal{D}_S$ and only update the encoder.

Also, note that different from some prior work [49], our network only requires ground-truth landmarks during training. It is able to predict landmarks via $\mathbf{P}$ and $\mathbf{S}$ during the test time.

Regularizations. To ensure plausible reconstruction, we add a few regularization terms:

$$\mathcal{L}_{\mathrm{reg}} = \mathcal{L}_{\mathrm{sym}}(\mathbf{A}) + \lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}(\mathbf{A}) + \lambda_{\mathrm{smo}}\mathcal{L}_{\mathrm{smo}}(\mathbf{S}). \tag{3.13}$$

Albedo Symmetry As the face is symmetry, we enforce the albedo symmetry constraint,

$$\mathcal{L}_{\mathrm{sym}}(\mathbf{A}) = \left\|\mathbf{A}^{\mathrm{uv}} - \mathrm{flip}(\mathbf{A}^{\mathrm{uv}})\right\|_1. \tag{3.14}$$

Employing on 2D albedo, this constraint can be easily implemented via a horizontal image flip operation $\mathrm{flip}()$.

Albedo Constancy Using symmetry constraint can help to correct the global shading. However, symmetrical details, i.e., dimples, can still be embedded in the albedo channel.

To further remove shading from the albedo channel, following Retinex theory [] which as-

sumes albedo to be piecewise constant, we enforce sparsity in two directions of its gradient, similar to [107, 144]:

$$\mathcal{L}_{\text{con}}(\mathbf{A}) = \sum_{\mathbf{v}_j^{\text{uv}} \in \mathcal{N}_i} \omega(\mathbf{v}_i^{\text{uv}}, \mathbf{v}_j^{\text{uv}}) \left\| \mathbf{A}^{\text{uv}}(\mathbf{v}_i^{\text{uv}}) - \mathbf{A}^{\text{uv}}(\mathbf{v}_j^{\text{uv}}) \right\|_2^p, \tag{3.15}$$

where $\mathcal{N}_i$ denotes a set of 4-pixel neighborhood of pixel $\mathbf{v}_i^{\text{uv}}$. With the assumption that pixels with the same chromaticity (i.e., $\mathbf{c}(x) = \mathbf{I}(x)/|\mathbf{I}(x)|$) are more likely to have the same albedo, we set the constant weight $\omega(\mathbf{v}_i^{\text{uv}}, \mathbf{v}_j^{\text{uv}}) = \exp\left(-\alpha \left\| \mathbf{c}(\mathbf{v}_i^{\text{uv}}) - \mathbf{c}(\mathbf{v}_j^{\text{uv}}) \right\|\right)$, where the color is referenced from the input image using the current estimated projection. Following [107], we set $\alpha = 15$ and $p = 0.8$ in our experiment.

Shape Smoothness For shape component, we impose the smoothness by adding the Laplacian regularization on the vertex locations for the set of all vertices.

$$\mathcal{L}_{\text{smo}}(\mathbf{S}) = \sum_{\mathbf{v}_i^{\text{uv}} \in \mathbf{S}^{\text{uv}}} \left\| \mathbf{S}^{\text{uv}}(\mathbf{v}_i^{\text{uv}}) - \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{v}_j^{\text{uv}} \in \mathcal{N}_i} \mathbf{S}^{\text{uv}}(\mathbf{v}_j^{\text{uv}}) \right\|_2. \tag{3.16}$$

Intermediate Semi-Supervised Training. Fully unsupervised training using only the reconstruction and adversarial loss on the rendered images could lead to a degenerate solution, since the initial estimation is far from ideal to render meaningful images. Therefore, we introduce intermediate loss functions to guide the training in the early iterations.

With the face profiling technique, Zhu *et al*. [193] expand the 300W dataset [134] into $122,450$ images with fitted 3DMM shapes $\widetilde{\mathbf{S}}$ and projection matrix $\widetilde{\mathbf{P}}$. Given $\widetilde{\mathbf{S}}$ and $\widetilde{\mathbf{P}}$, we create the pseudo groundtruth texture $\widetilde{\mathbf{T}}$ by referring every pixel in the UV space back to the input image, i.e., the backward of our rendering layer. With $\widetilde{\mathbf{P}}$, $\widetilde{\mathbf{S}}$, $\widetilde{\mathbf{T}}$, we define our intermediate loss by:

$$L_0 = L_{\text{S}} + \lambda_T L_{\text{T}} + \lambda_P L_{\text{P}} + \lambda_L L_{\text{L}} + \lambda_{\text{reg}} L_{\text{reg}}, \tag{3.17}$$

where:

$$L_{\mathrm{S}} = \left\| \mathbf{S} - \widetilde{\mathbf{S}} \right\|_2^2, \tag{3.18}$$

$$L_{\mathrm{T}} = \left\| \mathbf{T} - \widetilde{\mathbf{T}} \right\|_1, \tag{3.19}$$

$$L_{\mathrm{m}} = \left\| \mathbf{P} - \widetilde{\mathbf{P}} \right\|_2^2. \tag{3.20}$$

It's also possible to provide pseudo groundtruth to the SH coefficients $\mathbf{L}$ and followed by albedo $\mathbf{A}$ using least square optimization with a constant albedo assumption, as in [169, 144]. However, this estimation is not reliable for in-the-wild images with occlusion regions. Also empirically, with proposed regularizations, the model is able to explore plausible solutions for these components by itself. Hence, we decide to refrain from supervising $\mathbf{L}$ and $\mathbf{A}$ to simplify our pipeline.

Due to the pseudo groundtruth, using L_0 may run into the risk that our solution learns to mimic the linear model. Thus, we switch to the loss of Eqn. 3.8 after L_0 converges. Note that the estimated groundtruth of $\widetilde{\mathbf{P}}$, $\widetilde{\mathbf{S}}$, $\widetilde{\mathbf{T}}$ and the landmarks are the only supervision used in our training, for which our learning is considered as *weakly* supervised.

3.3 Experimental Results

The experiments study three aspects of the proposed nonlinear 3DMM, in terms of its expressiveness, representation power, and applications to facial analysis. Using facial mesh triangle definition by Basel Face Model (BFM) [121], we train our 3DMM using 300W-LP dataset [193], which contains $122,450$ in-the-wild face images, in a wide pose range from -90° to 90°. Images are loosely square cropped around the face and scale to 256×256. During training, images of size 224×224 are randomly cropped from these images to introduce translation variations.

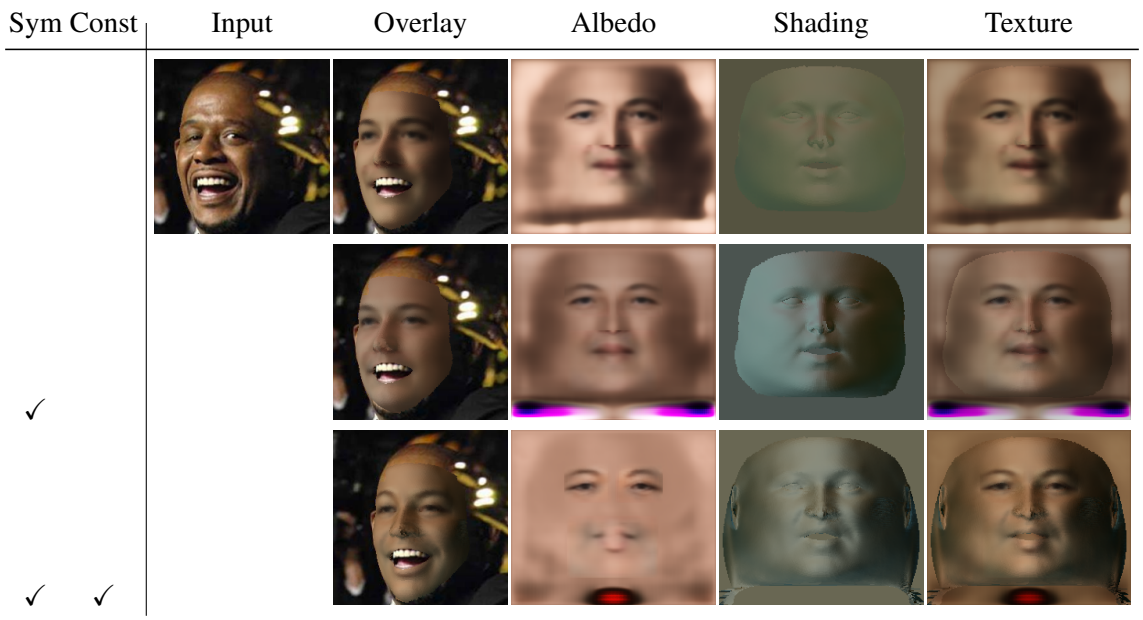


Figure 3.7: Effect of albedo regularizations: albedo symmetry (sym) and albedo constancy (const). When there is no regularization being used, shading is mostly baked into the albedo. Using the symmetry property helps to resolve the global lighting. Using constancy constraint futher removes shading from the albedo, which results in a better 3D shape.

The model is optimized using Adam optimizer with a learning rate of 0.001 in both training stages. We set the following parameters: $Q = 53,215$, $U = 192, V = 224$, $l_S = l_T = 160$. λ values are set to make losses to have similar magnitudes.

3.3.1 Ablation Study

3.3.1.1 Effect of Regularization

Albedo Regularization. In this work, to regularize albedo learning, we employ two constraints to efficiently remove shading from albedo namely albedo symmetry and constancy. To demonstrate the effect of these regularization terms, we compare our full model with its partial variants: one without any albedo requlrarization and one with the symmetry constraint only. Fig. 3.7 shows visual comparison of these models. Learning without any constraints results in the lighting is

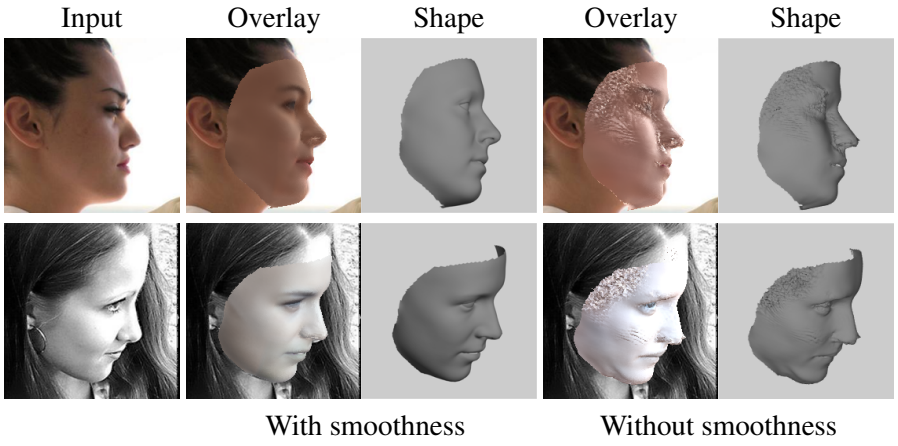


Figure 3.8: Effect of shape smoothness regularization.

totally explained by the albedo, meanwhile is the shading is almost constant (Fig. 3.7(a)). Using symmetry help to correct the global lighting. However, symmetric geometry details are still baked into the albedo (Fig. 3.7(b)). Enforcing albedo constancy helps to further remove shading from it (Fig. 3.7(c)). Combining these two regularizations helps to learn plausible albedo and lighting, which improves the shape estimation.

Shape Smoothness Regularization. We also evaluate the need in shape regularization. Fig. 3.8 shows visual comparisons between our model and its variant without the shape smoothness constraint. Without the smoothness term the learned shape becomes noisy especially on two sides of the face. The reason is that, the hair region is not completely excluded during training because of imprecise segmentation estimation.

3.3.1.2 Modeling Lighting and Shape Representation

In this work, we make two major algorithmic differences with our preliminary work [160]: incorporating lighting into the model and changing the shape representation.

Our previous work [160] models the texture directly, while this work disentangles the shading from the albedo. As argued, modeling the lighting should have a positive impact on shape learning.

Table 3.2: Face alignment performance on ALFW2000.

Method	Lighting	UV shape	NME
Our [160]			4.70
Our	✓		4.30
Our	✓	✓	**4.12**

Hence we compare our models with results from [160] in face alignment task.

Also, in our preliminary work [160], as well as in traditional 3DMM, shape is represented as a vector, where vertices are independent. Despite this shortage, this approach has been widely adopted due to its simplicity and sampling efficiency. In this work, we explore an alternative to this representation: represent the 3D shape as a position map in the 2D UV space. This representation has three channels: one for each spatial dimension. This representation maintains the spatial relation among facial mesh's vertices. Also, we can use CNN as the shape decoder replacing an expensive MLP. Here we also evaluate the performance gain by switching to this representation.

Tab. 3.2 reports the performance on the face alignment task of different variants. As a result, modeling lighting helps to reduce the error from 4.70 to 4.30. Using the 2D representation, with the convenience of using CNN, the error is further reduced to 4.12.

3.3.1.3 Comparison to Autoencoders

We compare our model-based approach with a convolutional autoencoder in Fig. 3.9. The autoencoder network has a similar depth and model size as ours. It gives blurry reconstruction results as the dataset contain large variations on face appearance, pose angle and even diversity background. Our model-based approach obtains sharper reconstruction results and provides semantic parameters allowing access to different components including 3D shape, albedo, lighting and projection matrix.

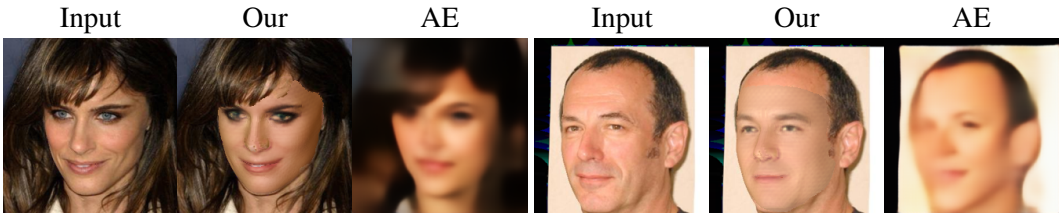


Figure 3.9: Comparison to convolutional autoencoders (AE). Our approach produces results of higher quality. Also it provides access to the 3D facial shape, albedo, lighting, and projection matrix.

3.3.2 Expressiveness

Exploring feature space. We feed the entire CelebA dataset [97] with ∼200k images to our network to obtain the empirical distribution of our shape and texture parameters. By varying the mean parameter along each dimension proportional to its standard deviation, we can get a sense how each element contribute to the final shape and texture. We sort elements in the shape parameter $\mathbf{f}_S$ based on their differences to the mean 3D shape. Fig. 3.10 shows four examples of shape changes, whose differences rank No.1, 40, 80, and 120 among 160 elements. Most of top changes are expression related. Similarly, in Fig. 3.11, we visualize different texture changes by adjusting only one element of $\mathbf{f}_A$ off the mean parameter $\bar{\mathbf{f}}_A$. The elements with the same 4 ranks as the shape counterpart are selected.

Attribute Embedding. To better understand different shape and albedo instances embedded in our two decoders, we dig into their attribute meaning. For a given attribute, e.g., male, we feed images with that attribute $\{\mathbf{I}_i\}_{i=1}^n$ into our encoder E to obtain two sets of parameters $\{\mathbf{f}_S^i\}_{i=1}^n$ and $\{\mathbf{f}_A^i\}_{i=1}^n$. These sets represent corresponding empirical distributions of the data in the low dimensional spaces. Computing the mean parameters $\bar{\mathbf{f}}_S, \bar{\mathbf{f}}_A$ and feed into their respective decoders, also using the mean lighting parameter, we can reconstruct the mean shape and texture with that attribute. Fig. 3.12 visualizes the reconstructed textured 3D mesh related to some attributes. Differences among attributes present in both shape and texture. Here we can observe the power of our

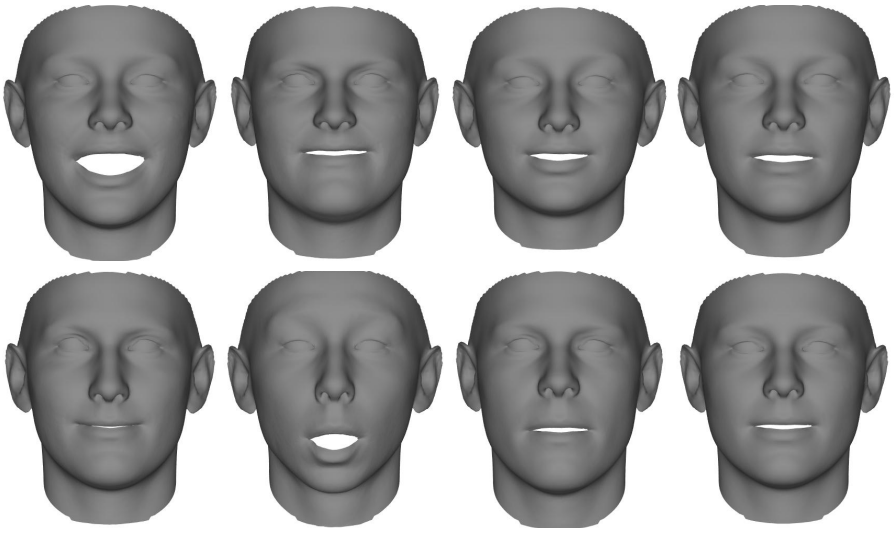


Figure 3.10: Each column shows shape changes when varying one element of $\mathbf{f}_S$, by 10 times standard deviations, in opposite directions. Ordered by the magnitude of shape changes.

Figure 3.11: Each column shows albedo changes when varying one element of $\mathbf{f}_A$ in opposite directions.

nonlinear 3DMM to model small details such as “bag under eyes", or “rosy cheeks", etc.

3.3.3 Representation Power

We compare the representation power of the proposed nonlinear 3DMM vs. traditional linear 3DMM.

Albedo. Given a face image, assuming we know the groundtruth shape and projection parameters, we can unwarp the texture into the UV space, as we generate “pseudo groundtruth" texture in

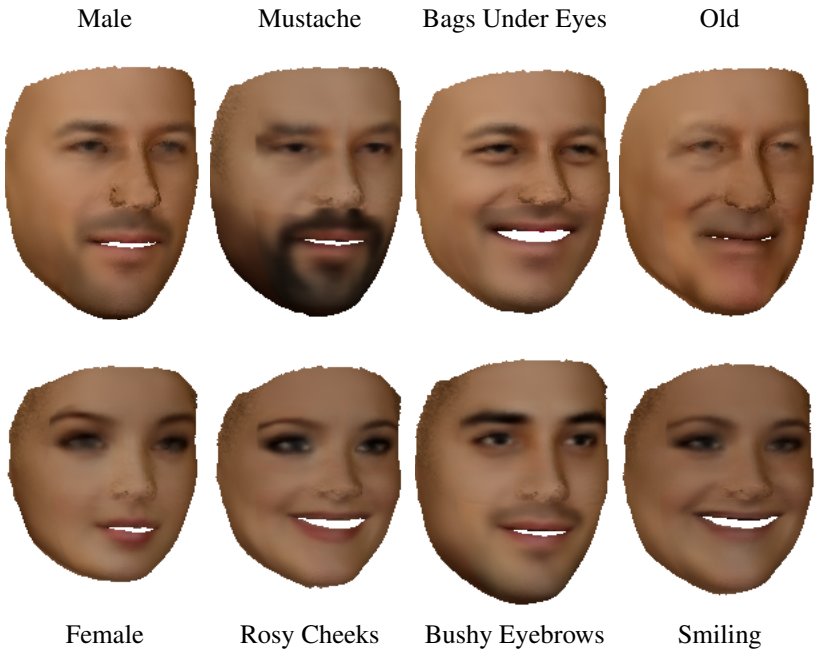


Figure 3.12: Nonlinear 3DMM generates shape and albedo embedded with different attributes.

the weakly supervision step. With the groundtruth texture, by using gradient descent, we can jointly estimate, a lighting parameter $\mathbf{L}$ and an albedo parameter $\mathbf{f}_A$ whose decoded texture matches with the groundtruth. Alternatively, we can minimize the reconstruction error in the image space, through the rendering layer with the groundtruth $\mathbf{S}$ and $\mathbf{P}$. Empirically, two methods give similar performances but we choose the first option as it involves only one warping step, instead of doing rendering in every optimization iteration. For the linear model, we use albedo bases of Basel Face Model (BFM) [121]. As in Fig. 3.13, our nonlinear texture is closer to the groundtruth than the linear model. This is expected since the linear model is trained with controlled images. Quantitatively, our nonlinear model has significantly lower averaged L_1 reconstruction error than the linear model (0.053 vs. 0.097, as in Tab. 3.3).

3D Shape. We also compare the power of nonlinear and linear 3DMMs in representing real-world 3D scans. We compare with BFM [121], the most commonly used 3DMM at present. We use ten 3D face scans provided by [121], which are not included in the training set of BFM. As these

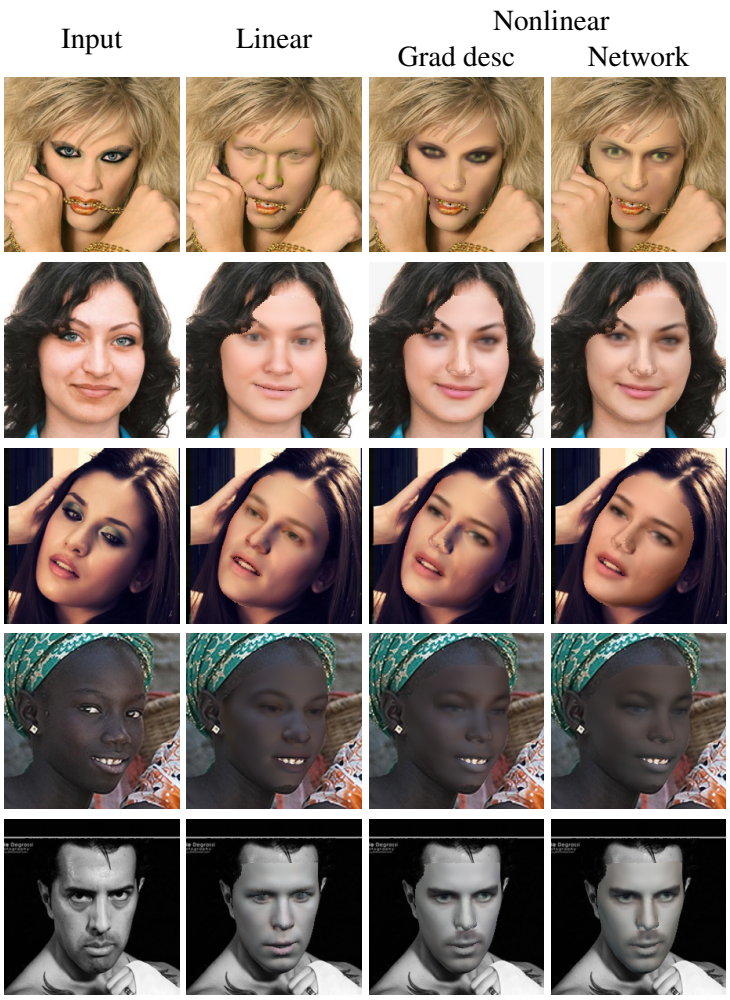


Figure 3.13: Texture representation power comparison. Our nonlinear model can better reconstruct the facial texture.

Table 3.3: Quantitative comparison of texture representation power (Average reconstruction error on non-occluded face portion.)

Method	Linear	Nonlinear w. Grad De.	Nonlinear w. Network
L_1	0.062	0.053	0.057

face meshes are already registered using the same triangle definition with BFM, no registration is necessary. Given the groundtruth shape, by using gradient descent, we can estimate a shape parameter whose decoded shape matches the groundtruth. We define matching criterion on both vertex distances and surface normal direction. This empirically improves fidelity of final results

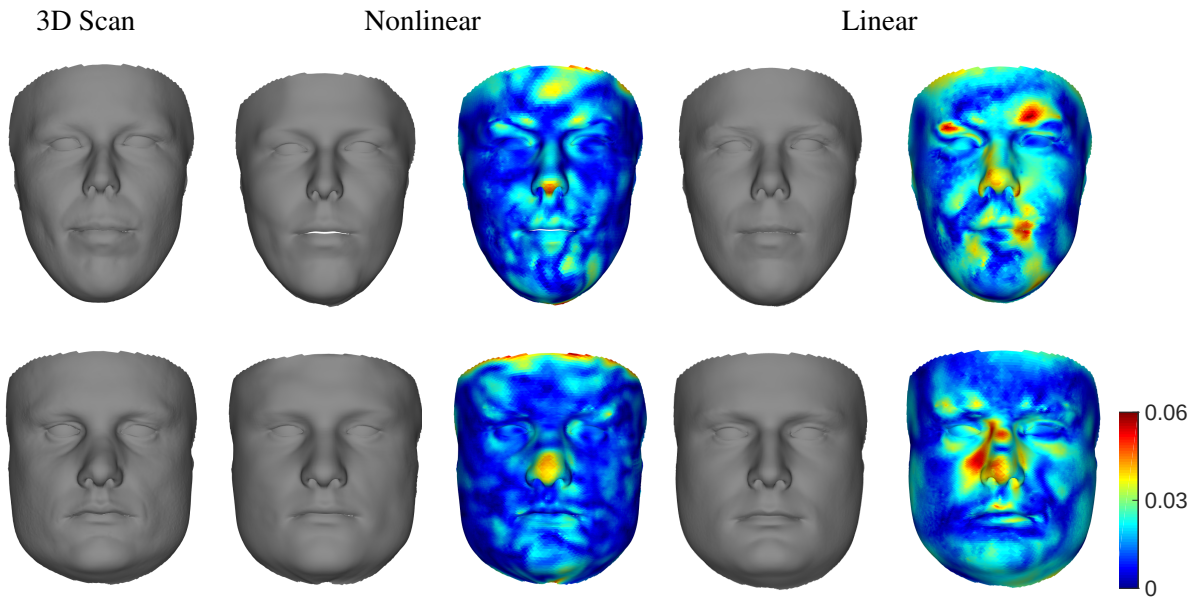


Figure 3.14: Shape representation power comparison ($l_S = 160$). The error map show the normalized per-vertex error.

Table 3.4: 3D scan reconstruction comparison (NME).

l_S	40	80	160
Linear	0.0321	0.0279	0.0241
Nonlinear[160]	0.0277	0.0236	0.0196
Nonlinear	0.0268	0.0214	**0.0146**

compared to only optimizing vertex distances. Also, to emphasize the compactness of nonlinear models, we train different models with different latent space sizes. Fig. 3.14 shows the visual quality of two models' reconstruction. Our reconstructions closely match the face shapes details.

To quantify the difference, we use NME, averaged per-vertex errors between the recovered and groundtruth shapes, normalized by inter-ocular distances. Our nonlinear model has a significantly smaller reconstruction error than the linear model, 0.0146 vs. 0.0241 (Tab. 3.4). Also, the nonlinear models are more compact. They can achieve similar performances as linear models whose latent space's sizes doubled.

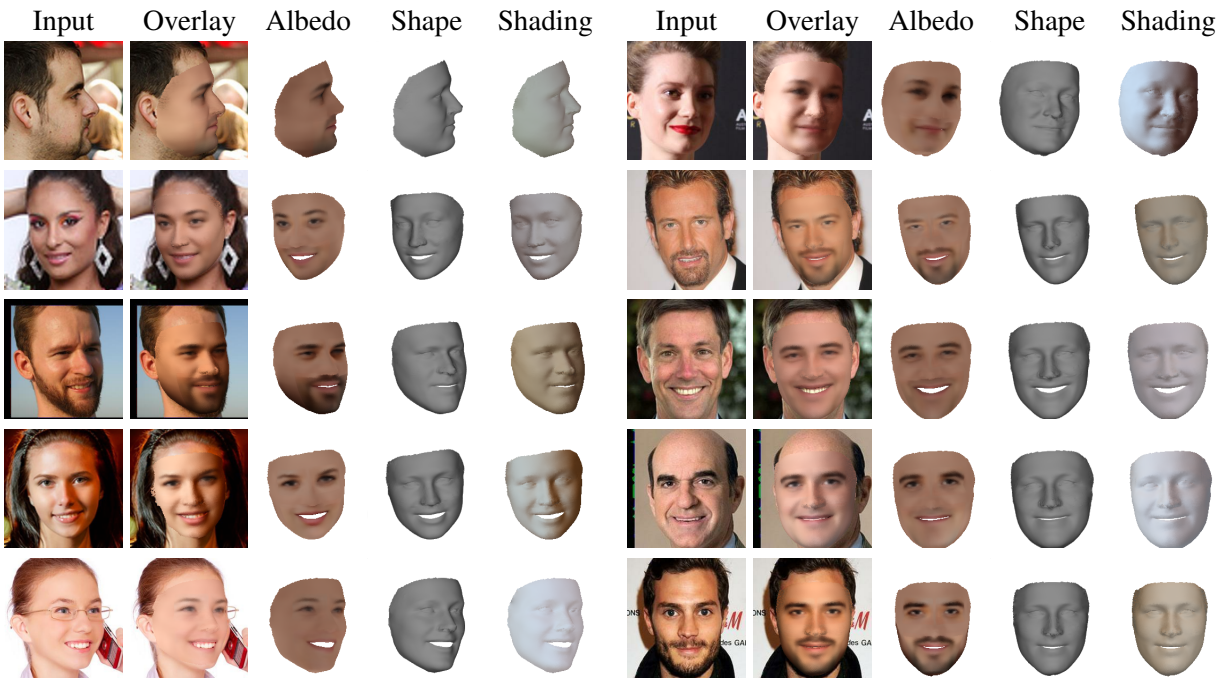


Figure 3.15: 3DMM fits to faces with diverse skin color, pose, expression, lighting, facial hair, and faithfully recovers these cues. Left half shows results from AFLW2000 dataset, right half shows results from CelebA.

3.3.4 Applications

Having shown the capability of our nonlinear 3DMM (i.e., two decoders), now we demonstrate the applications of our entire network, which has the additional encoder. Many applications of 3DMM are centered on its ability to fit to 2D face images. Similar to linear 3DMM, our nonlinear 3DMM can be utilized for model fitting, which decomposes a 2D face into its shape, albedo and lighting. Fig. 3.15 visualizes our 3DMM fitting results on AFLW2000 and CelebA dataset. Our encoder estimates the shape **S**, albedo **A** as well as lighting **L** and projection matrix **P**. We can recover personal facial characteristic in both shape and albedo. Our albedo can present facial hair, which is normally hard to be recovered by linear 3DMM.

Figure 3.16: Our face alignment results. Invisible landmarks are marked as red. We can well handle extreme pose, lighting and expression.

3.3.4.1 Face Alignment

Face alignment is a critical step for many facial analysis tasks such as face recognition [162, 163]. With enhancement in the modeling, we hope to improve this task (Fig. 3.16). We compare face alignment performance with state-of-the-art methods, 3DDFA [193], DeFA [96], 3D-FAN [22] and PRN [46], on AFLW2000 dataset on both 2D and 3D settings.

The accuracy is evaluated using Normalized Mean Error (NME) as the evaluation metric with bounding box size as the normalization factor [22]. For fair comparison with these methods in term of computational complexity, for this comparison we use ResNet18 [60] as our encoder. Here, 3DDFA and DeFA use the linear 3DMM model (BFM). Even though being trained with larger training corpus (DeFA) or having a cascade of CNNs iteratively refines the estimation (3DDFA), these methods are still significantly outperformed by our nonlinear model (Fig. 3.17). Meanwhile, 3D-FAN and PRN achieve competitive performances by by-passing the linear 3DMM model. 3D-FAN uses heat map representation. PRN uses the position map representation which shares a similar spirit to our UV representation. Not only outperforms these methods in term of regressing landmark locations (Fig. 3.17), our model also directly provides head pose information as well as the facial albedo and environment lighting condition.

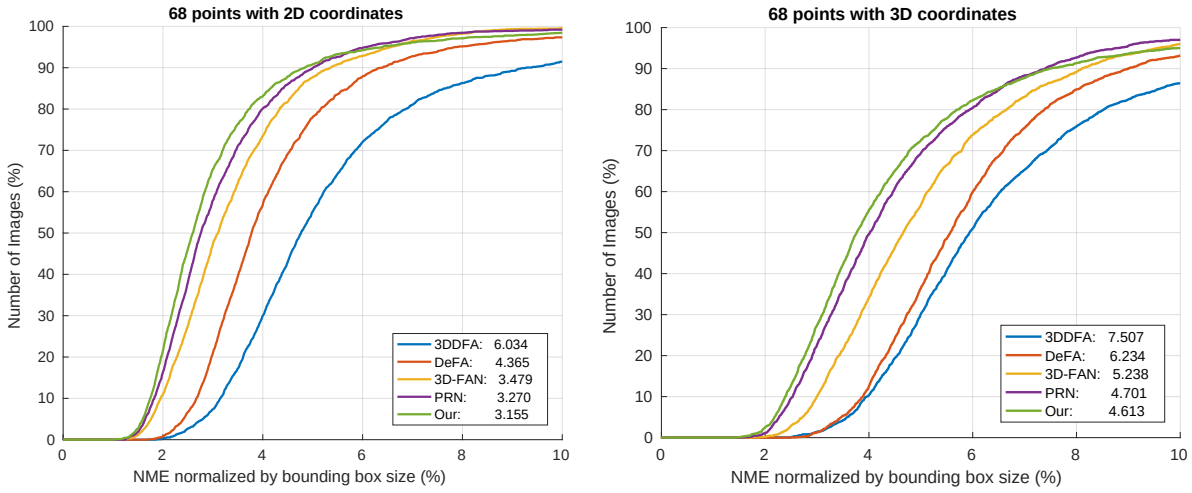


Figure 3.17: Face alignment Cumulative Errors Distribution (CED) curves on AFLW2000-3D on 2D (left) and 3D landmarks (right). NMEs are shown in legend boxes.

3.3.4.2 3D Face Reconstruction

We compare our approach to three recent representative face reconstruction work: 3DMM fitting networks learned in unsupervised (Tewari *et al*. [153, 152]) or supervised fashion (Sela *et al*. [139]) and also a non-3DMM approach (Jackson *et al*. [63]).

MoFA, the monocular reconstruction work by Tewari *et al*. [153], is relevant to us as they also learn to fit 3DMM in an unsupervised fashion. Even being trained on in-the-wild images, their method is still limited to the linear bases. Hence there reconstructions suffer the surface shrinkage when dealing with challenging texture, i.e., facial hair (Fig. 3.18). Our network faithfully models these in-the-wild texture, which leads to better 3D shape reconstruction.

Concurrently, Tewari *et al*. [152] try to improve the linear 3DMM representation power by learning a corrective space on top of a traditional linear model. Despite sharing similar spirit, our unified model exploits spatial relation between neighbor vertices and uses CNNs as shape/albedo decoders, which is more efficient than MLPs. As a result, our reconstructions more closely match the input images in both texture and shape (Fig. 3.19).

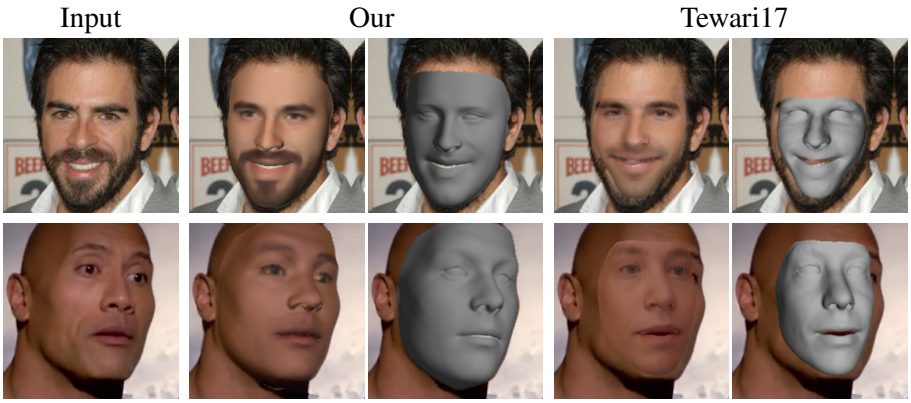


Figure 3.18: 3D reconstruction results comparison to Tewari *et al*. [153]. Their reconstructed shapes suffer from the surface shrinkage when dealing with challenging texture or shape outside the linear model subspace. They can't handle large pose variation well either. Meanwhile, our nonlinear model is more robust to these variations.

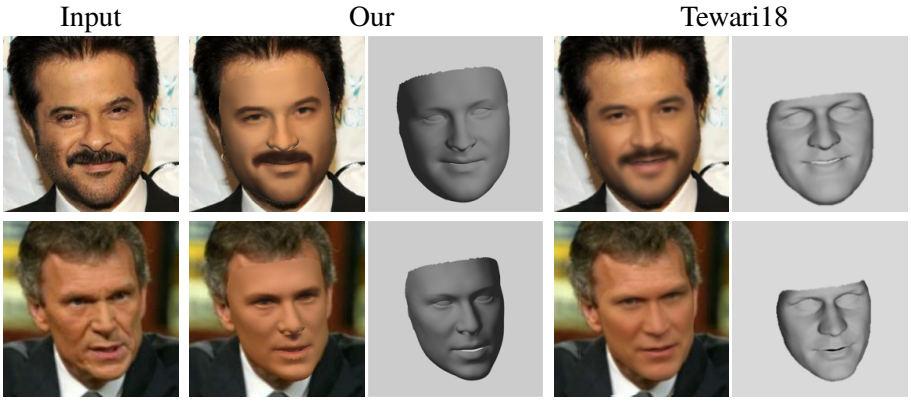


Figure 3.19: 3D reconstruction results comparison to Tewari *et al*. [152]. Our model better reconstruct the input image in both texture (facial hair direction on the first image) and shape (nasolabial folds in the second image).

The high-quality 3D reconstruction work by Richardson *et al*. [128, 129], Sela *et al*. [139] obtain impressive results on adding fine-level details to the face shape when images are within the span of the used synthetic training corpus or the employed 3DMM model. However, their performance significantly degrades when dealing with variations not in its training data span, e.g., facial hair. Our approach is not only robust to facial hair and make-up, but also automatically learns to reconstruct such variations based on the jointly learned model. We provide comparisons with them in Fig. 3.20, using the code provided by the author.

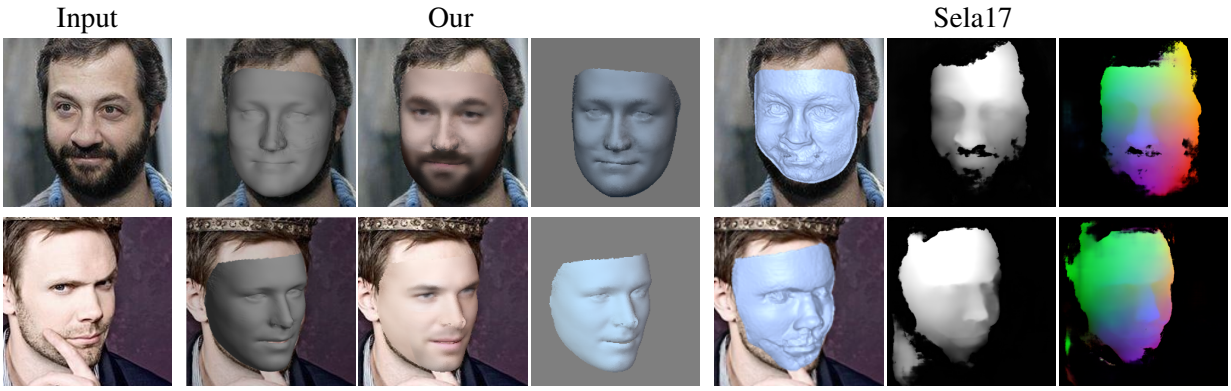


Figure 3.20: 3D reconstruction results comparison to Sela *et al*. [129]. Besides showing the shape, we also show their estimated depth and correspondence map. Facial hair or occlusion can cause serious problems in their output maps.

The current state-of-art method by Sela *et al*. [139] consisting of three steps: an image-to-image network estimating a depth map and a correspondence map, non-rigid registration and a fine detail reconstruction. Their image-to-image network is trained on synthetic data generated by the linear model. Besides domain gap between synthetic and real images, this network faces a more serious problem of lacking facial hair in the low-dimension texture subspace of the linear model. This network's output tends to ignore these unexplainable region (Fig. 3.20), which leads to failure in later steps. Our network is more robust in handing these in-the-wild variations. Furthermore, our approach is orthogonal to Sela *et al*. [139]'s fine detail reconstruction module or Richardson *et al*. [129]'s finenet. Employing these refinement on top of our fitting could lead to promising further improvement.

We also compare our approach with a non-3DMM apporach VRN by Jackson *et al*. [63]. To avoid using low-dimension subspace of the linear 3DMM, it directly regresses a 3D shape volumetric representation via an encoder-decoder network with skip connection. This potentially helps the network to explore a larger solution space than the linear model, however with a cost of losing correspondence between facial meshes. Fig. 3.21 shows 3D reconstruction visual comparison between VRN and ours. In general, VRN robustly handles in-the-wild texture variations. However,

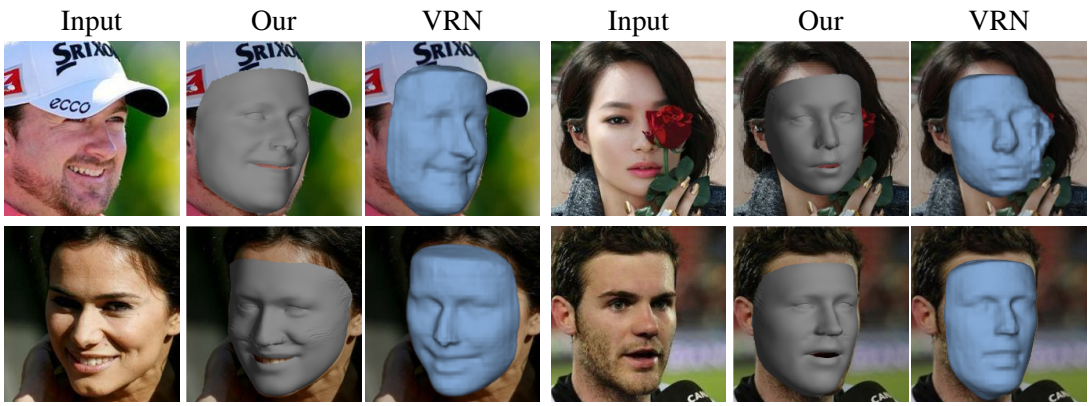


Figure 3.21: 3D reconstruction results comparison to VRN by Jackson *et al*. [63] on CelebA dataset. Volumetric shape representation results in non-smooth 3D shape and loses correspondence between reconstructed shapes.

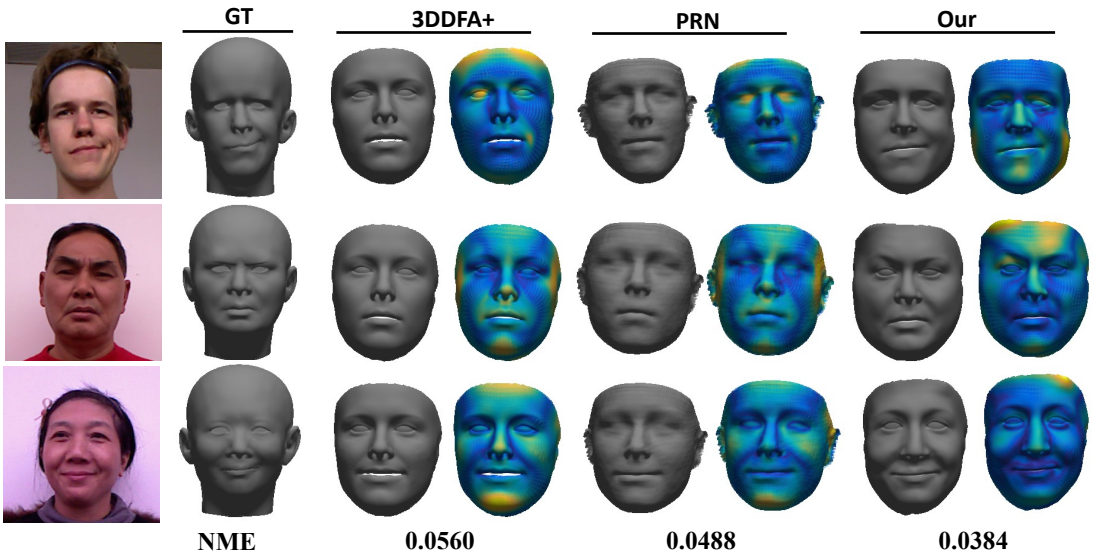
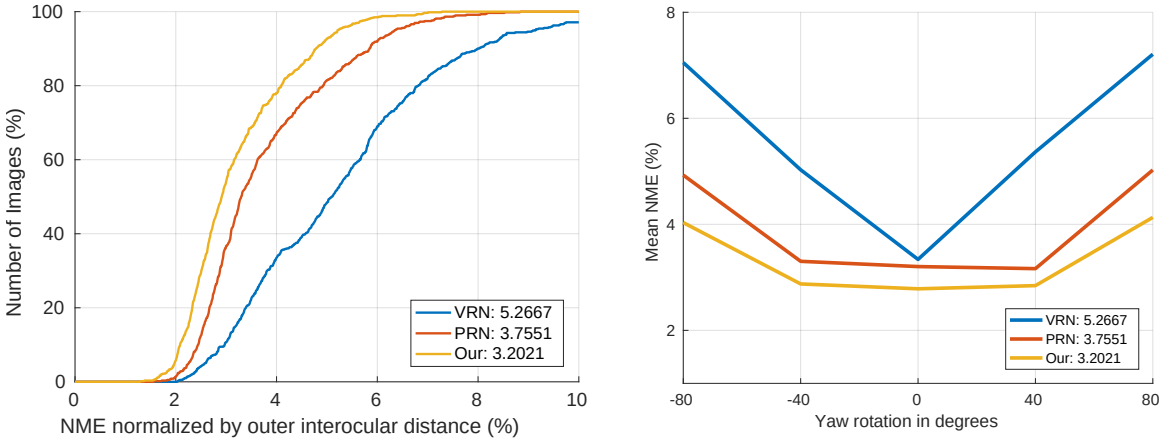


Figure 3.22: 3D reconstruction quantitative evaluation on FaceWarehouse. We obtain a lower error compared to PRN [46] and 3DDFA+ [195].

because of the volumetric shape representation, the surface is not smooth and is partially limited to present medium-level details as ours. Also, our model further provides projection matrix, lighting and albedo, which is applicable for more applications.

Quantitative Comparisons.

To quantitatively compare our method with prior works, we evaluate monocular 3D reconstruction performance on FaceWarehouse [24] and Florence dataset [9], in which groundtruth 3D shape



(a) CED Curves (b) Pose-specific NME

Figure 3.23: 3D face reconstruction results on the Florence dataset [9]. The NME of each method is showed in the legend

is available. Due to the diffrence in mesh topology, ICP [7] is used to establish correspondence between estimated shapes and ground truth point clouds. Similar to previous experiments, NME (averaged per-vertex errors normalized by inter-ocular distances) is used as the comparison metric.

FaceWarehouse. We compare our method with prior works with available pretrained models on all 19 expressions of 150 subjects of FaceWarehouse database [24]. Visual and quantitative comparisons are shown in Fig. 3.22. Our model can faithfully resemble the input expression and significantly surpass all other regression methods (PRN [46] and 3DDFA+ [195]) in term of dense face alignment.

Florence. Using the experimental setting proposed in [63], we also quantitatively compared our approach with state-of-the-art methods (*e.g.* VRN [63] and PRN [46]) on the Florence dataset [9]. Each subject is rendered with multiple poses: pitch rotations of $-15°$, $20°$ and $25°$ and raw rotations between $-80°$ and $80°$. Our model consistently outperforms other methods across different view angles (Fig. 3.23).

Table 3.5: Running time of various 3D face reconstruction methods.

Method	Encoder	Decoder	Post-processing	Rendering
Sela *et al*. [139]	$\sim$ 10 ms		$\sim$ 180s	-
VRN [63]	$\sim$ 10 ms		-	-
MoFA [153]	$\sim$ 4ms	Neglectable	-	-
Our	2.7ms	5.5 ms	-	140 ms

3.3.5 Runtime

In this section, we compare running time for multiple 3D reconstruction approaches. Since different methods implemented in different frameworks/languages; this comparison aims to only provide relative comparisons between them. Sela *et al*. [139] and VRN [63] both use an encoder-decoder network with skip connections with similar runtime. However, Sela *et al*. [139] requires an expensive nonrigid registration step as well as an refinement module. We get a comparable encoder running time with 3DMM regression network of MoFA [153]. However, since they directly use liner bases, the decoding step is trivial as a single multiplication; our model requires decoding features via two CNNs for shape and texture, respectively. We also note that the running time for the rendering layer is significantly higher than other components. Luckily, rendering to reconstruct input has no value and it is not required during testing.

3.4 Conclusions

Since its debut in 1999, 3DMM has became a cornerstone of facial analysis research with applications to many problems. Despite its impact, it has drawbacks in requiring training data of 3D scans, learning from controlled 2D images, and limited representation power due to linear bases for both shape and texture. These drawbacks could be formidable when fitting 3DMM to unconstrained faces, or learning 3DMM for generic objects such as shoes. This paper demonstrates that there

exists an alternative approach to 3DMM learning, where a nonlinear 3DMM can be learned from a large set of in-the-wild face images without collecting 3D face scans. Further, the model fitting algorithm can be learnt jointly with 3DMM, in an end-to-end fashion.

Our experiments cover a diverse aspects of our learnt model, some of which might need the subjective judgment of the readers. We hope that both the judgment and quantitative results could be viewed under the context that, unlike linear 3DMM, no genuine 3D scans are used in our learning. Finally, we believe that unsupervisedly or weak-supervisedly learning 3D models from large-scale in-the-wild 2D images is one promising research direction. This work is one step along this direction.

Chapter 4

Towards High-fidelity Nonlinear 3D Face Morphoable Model

4.1 Introduction

In chapter 3, we present our proposed framework using deep neural networks to present the 3DMM basis functions to increase model representation power and learning the model directly from unconstrained 2D images to better capture in-the-wild variations.

However, even with better representation powers, this model or related works [152] still rely on many constraints to regularize the model learning. Hence, their objective involves the conflicting requirements of a strong regularization for a global shape vs. a weak regularization for capturing higher level details. For example, in order to faithfully separate shading and albedo, albedo is usually assumed to be piecewise constant [82, 144], which prevents learning albedo with high level of details. In this chapter, beside learning the shape and the albedo, we propose to learn additional shape and albedo proxies, on which we can enforce regularizations. This also allows us to flexibly pair the true shape with strongly regularized albedo proxy to learn the detailed shape or

This chapter is adapted from the following publication:
[1] Luan Tran, Feng Liu, and Xiaoming Liu, "Towards High-fidelity Nonlinear 3D Face Morphable Model" in CVPR, 2019.

vice versa. As a result, each element can be learned with high-fidelity without sacrificing the other element's quality.

On a different note, many 3DMM models fail to represent small details because of their parameterization. Many global 3D face parameterization has been proposed to overcome the ambiguities associated with monocular face fitting/tracking such as noise or occlusion. However, because they are designed to model the whole face at once, it is difficult to use them to represent small details. Meanwhile, local-based models offer more flexibility than global approaches but with the cost of being less constrained to realistically represent human faces. We propose using dual-pathway networks to provide a better balance between global and local-based models. From the latent space, there is a global pathway focusing on the inference of global face structure and multiple local pathways generating details of different semantic facial parts. Their corresponding features are then fused together for successive process generation of the final shape and albedo. This network also helps to specialize filters in local pathways for each facial part which both improves the quality and saves computation power.

In this chapter, we improve the nonlinear 3D face morphable model in both learning objective and architecture:

◇ We solve the conflicting objective problem by learning additional shape and albedo proxies with proper regularization.

◇ The novel pairing scheme allows learning both detailed shape and albedo without sacrificing one's quality.

◇ The global local-based network architecture offers more balance between model robustness and flexibility.

◇ The proposed model allows, for the first time, high-fidelity 3D face reconstruction by solely optimize latent representations.

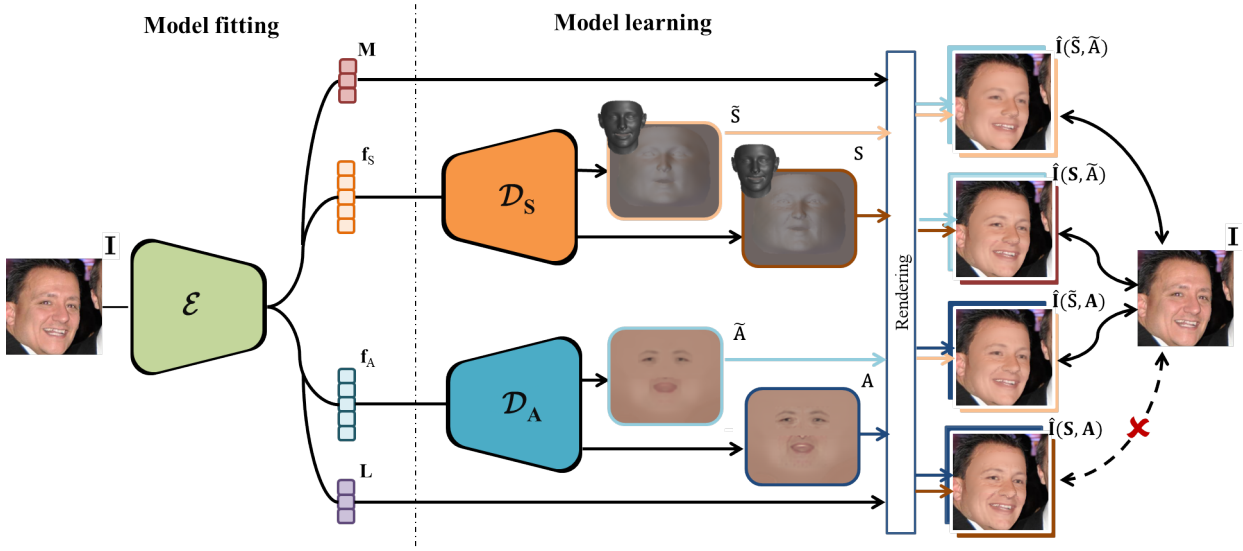


Figure 4.1: The proposed framework. Each shape or albedo decoder consist of two branches to reconstruct the true element and its proxy. Proxies free shape and albedo from strong regularizations, allow them to learn models with high level of details.

4.2 Proposed Method

4.2.1 Nonlinear 3DMM with Proxy and Residual

Recal from the last chapter, in the original nonlinear 3DMM, the overall objective can be summarized as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}}(\hat{\mathbf{I}}, \mathbf{I}) + \mathcal{L}_{\text{lan}} + \mathcal{L}_{\text{reg}}, \tag{4.1}$$

$$\text{with } \mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{sym}}(\mathbf{A}) + \lambda_{\text{con}}\mathcal{L}_{\text{con}}(\mathbf{A}) + \lambda_{\text{smo}}\mathcal{L}_{\text{smo}}(\mathbf{S}). \tag{4.2}$$

Proxy and Residual Learning. Strong regularization has been shown to be critical in ensuring the plausibility of the learned models [152, 161]. However, the strong regularization also prevents the model from recovering high-level of details in either shape or albedo. Hence, this prevents us from achieving the ultimate goal of learning a high-fidelity 3DMM model.

In this work, we propose to learn additional **proxy shape** ($\tilde{\mathbf{S}}$) and **proxy albedo** ($\tilde{\mathbf{A}}$), on which

we can apply the regularization. All presented regularizations will be moved to proxies now:

$$\mathcal{L}^{*}_{\text{reg}} = \mathcal{L}_{\text{sym}}(\tilde{\mathbf{A}}) + \lambda_{\text{con}}\mathcal{L}_{\text{con}}(\tilde{\mathbf{A}}) + \lambda_{\text{smo}}\mathcal{L}_{\text{smo}}(\tilde{\mathbf{S}}). \tag{4.3}$$

There will be no regularization applied directly to the actual shape $\mathbf{S}$ and albedo $\mathbf{A}$ other than a weak regularization encouraging each to be close to its proxy:

$$\mathcal{L}_{\text{res}} = \|\Delta\mathbf{S}\|_1 + \|\Delta\mathbf{A}\|_1 = \left\|\mathbf{S}-\tilde{\mathbf{S}}\right\|_1 + \left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_1. \tag{4.4}$$

By pairing two shapes $\mathbf{S},\tilde{\mathbf{S}}$ and two albedos $\mathbf{A},\tilde{\mathbf{A}}$, we can render four different output images (Fig. 4.1). Any of them can be used to compare with the original input image. We rewrite our reconstruction loss as:

$$\mathcal{L}^{*}_{\text{rec}} = \mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\tilde{\mathbf{S}},\tilde{\mathbf{A}}),\mathbf{I}) + \mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\tilde{\mathbf{S}},\mathbf{A}),\mathbf{I}) + \mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\mathbf{S},\tilde{\mathbf{A}}),\mathbf{I}). \tag{4.5}$$

Pairing strongly regularized proxies and weakly regularized components is a critical point in our approach. Using proxies allows us to learn high-fidelity shape and albedo without sacrificing quality of either component. This pairing is inspired by the observation that Shape from Shading techniques are able to recover detailed face mesh by assuming over regularized albedo or even using the mean albedo [129]. Here, $\mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\mathbf{S},\tilde{\mathbf{A}}),\mathbf{I})$ loss promote $\mathbf{S}$ to recover more details as $\tilde{\mathbf{A}}$ is constrained by piece-wise constant $\mathcal{L}_{\text{con}}(\tilde{\mathbf{A}})$ objective. Vice versa, $\mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\tilde{\mathbf{S}},\mathbf{A}),\mathbf{I})$ aims to learn better albedo. In order for these two losses to work as desired, proxies $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{A}}$ should perform well enough to approximate the input images by themselves. Without $\mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\tilde{\mathbf{S}},\tilde{\mathbf{A}}),\mathbf{I})$, a valid solution that minimizes $\mathcal{L}_{\text{rec}}(\hat{\mathbf{I}}(\mathbf{S},\tilde{\mathbf{A}}),\mathbf{I})$ is combination of a constant albedo proxy and noisy shape creating surface normal with dark shading in necessary regions, i.e., eyebrows.

Another notable design choice is that we intentionally left out the loss function on $\hat{\mathbf{I}}(\mathbf{S}, \mathbf{A})$, even though this theoretically is the most important objective. This is to avoid the case that the shape $\mathbf{S}$ learns an in-between solution that works well with both $\tilde{\mathbf{A}}, \mathbf{A}$ and vice versa.

Occlusion Imputation. With proposed objective function, our model is able to faithfully reconstruct input images. However, we empirically found that besides high-fidelity visible regions, the model tends to keep invisible region smooth. Since there is no supervision on those areas other than the residual magnitude loss pulling the shape and albedo closer to their proxies. To learn a more meaningful model, which is beneficial to other applications, i.e., face editing or face synthesis, we propose to use a soft symmetry loss [159] on occluded regions:

$$\mathcal{L}_{\text{res-sym}}(\mathbf{S}) = \left\| \mathbf{T} \circ (\Delta \mathbf{S}_z^{\text{uv}} - \text{flip}(\Delta \mathbf{S}_z^{\text{uv}})) \right\|_1, \tag{4.6}$$

where $\mathbf{T}$ is a mask in UV space indicating visibility of each pixel, approximated based on current surface normal direction. Even though the shape itself is not symmetric, i.e., face with asymmetric expression, we enforce symmetrical property on its depth residual.

4.2.2 Global Local Based Network Architecture

While global-based models are usually robust to noise and mismatches, they are usually over-constrained and do not provide sufficient flexibility to represent high-frequency deformations as local-based models. In order to take the best of both worlds, we propose to use dual-pathway networks for our shape and albedo decoders.

Here, we transfer the success of combining local and global models in image synthesis [110, 62] to 3D face modeling. The general architecture of a decoder is shown in Fig. 4.2. From the latent vector, there is a global pathway focusing on the inference of global structure and a local

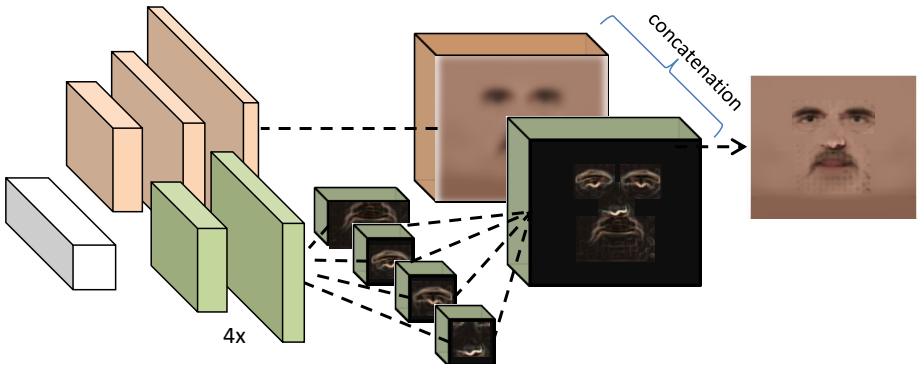


Figure 4.2: The proposed global local based network architecture.

pathway with four small sub-networks generating details of different facial parts, including eyes, nose and mouth. The global pathway is built from fractional strided convolution layers with five up-sampling steps. Meanwhile, each sub-network in local pathway have similar architecture but shallower with only three up-sampling steps. Using different small sub-networks for each facial part offers two benefits: i) with less up-sampling steps, the network is better able to represent high frequency details in early layers ii) each sub-network can learn part-specific filters which is more computationally efficient than applying across global face.

As shown in Fig. 4.2, to fuse two pathways' features, we firstly integrate four local pathways' outputs into one single feature tensor. Different from other works that synthesize face images with different yaw angles [162, 163, 73] with no fixed keypoints' locations, our 3DMM generates facial albedo as well as 3D shape in UV space with predefined topology. Merging these local feature tensors is efficiently done with zero padding operation. The max-pooling fusion strategy is also used to reduce the stitching artifacts on the overlapping areas. Then resultant feature is simply concatenated with the global pathway's feature, which has the same spatial resolution. Successive convolution layers integrate information from both pathways and generate the final albedo/shape (or their proxies).

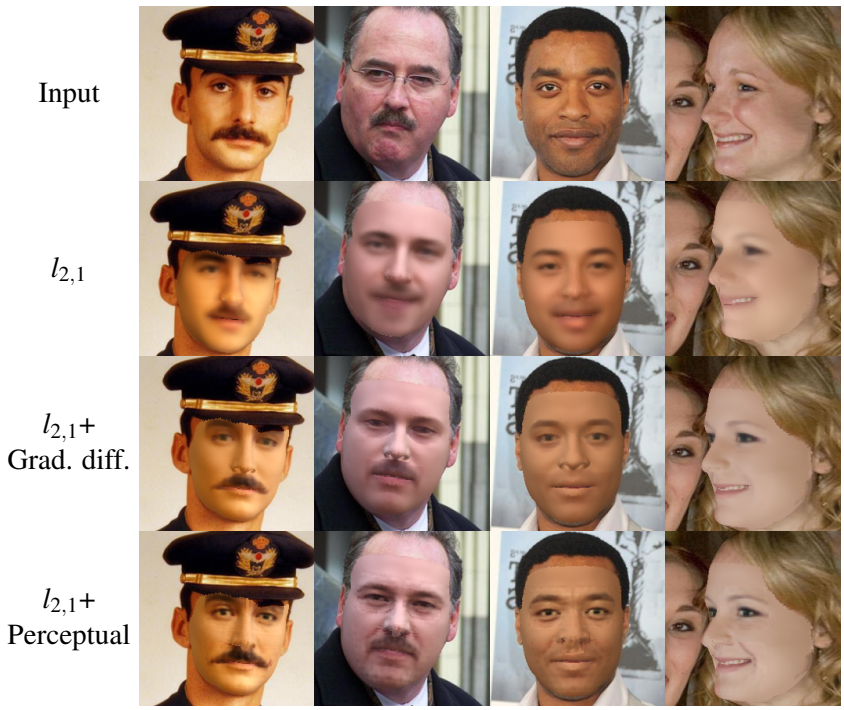


Figure 4.3: Reconstruction results with different loss functions.

4.3 Experimental Results

The experiments study different aspects of the proposed nonlinear 3DMM, in terms of its representation power, and applications to facial analysis. The model is trained followed the same setting as in chapter 3, including training dataset, mesh topology, optimizer parametters.

4.3.1 Ablation Study

Reconstruction Loss Functions. We study effects of different reconstruction losses on quality of the reconstructed images (Fig. 4.3). As expected, the model trained with $l_{2,1}$ loss only results in blurry reconstruction, similar to other l_p loss. To make the reconstruction to be more realistic, we explore other options such as gradient difference [104] or perceptual loss [66]. While adding the gradient difference loss creates more details in the reconstruction, combining perceptual loss with

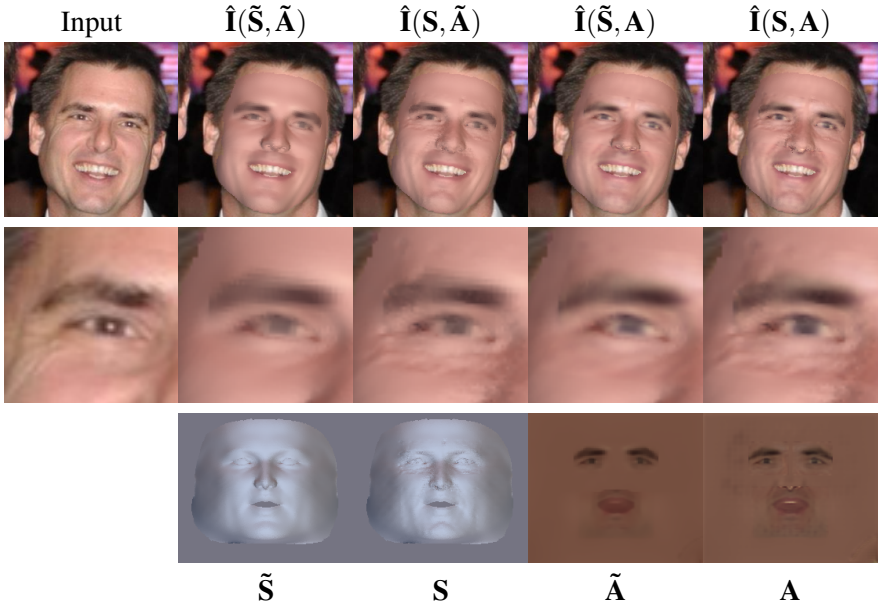
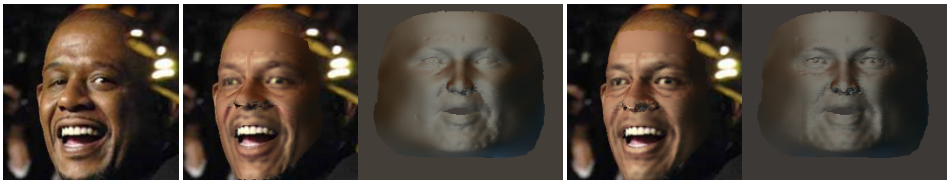


Figure 4.4: Image reconstruction with our 3DMM model using the proxy and the true shape and albedo. Our shape and albedo can faithfully recover details of the face. Note: for the shape, we show the shading in UV space – a better visuallization than the raw $\mathbf{S}^{\text{UV}}$.

$l_{2,1}$ gives best results with high level of details and realism. For the rest of the paper we will refer to the model trained using this combination.

Understanding image pairing. Fig. 4.4 shows fitting results of our model on a 2D face image. By using the proxy or the final components (shape or albedo) we can render four different reconstructed images with different quality and characteristics. The image generated by two proxies $\tilde{\mathbf{S}},\tilde{\mathbf{A}}$ is quite blurry but is still be able to capture major variations in the input face. By pairing $\mathbf{S}$ and the proxy $\tilde{\mathbf{A}}$, S is enforced to capture high level of details to bring the image closer to the input. Similarly, A is also encouraged to capture more details by pairing with the proxy $\tilde{\mathbf{S}}$. The final image $\hat{\mathbf{I}}(\mathbf{S},\mathbf{A})$ inherently achieves high level of details and realism even without direct optimization.

Residual Soft Symmetry Loss. We study effects of the residual soft symmetry loss on recovering details on occluded face region. As shown in Fig. 4.5, without $\mathcal{L}_{\text{res-sym}}$, the learned model can result in an unnatural shape, in which one side of the face is over-smooth, on occluded regions, while the



Without $\mathcal{L}_{\text{res-sym}}$ With $\mathcal{L}_{\text{res-sym}}$

Figure 4.5: Affect of soft symmetry loss on our shape model.

Table 4.1: Quantitative comparison of texture representation power (Average reconstruction error on non-occluded face portion.)

Method	Reconstruction error ($l_{2,1}$)
Linear [193]	0.1287
Nonlinear [161]	0.0427
Nonlinear + GL (Ours)	0.0386
Nonlinear + GL + Proxy (Ours)	**0.0363**

other side still has high level of details. Our model learned with $\mathcal{L}_{\text{res-sym}}$ can consistently create details across the face, even in occluded areas.

4.3.2 Representation Power

We compare the representation power of the proposed nonlinear 3DMM with Basel Face Model [121], the most commonly used linear 3DMM. We also make comparisons with the recently proposed nonlinear 3DMM [160].

Texture. We evaluate our model's power to represent in-the-wild facial texture. Given a face image, also with the groundtruth shape and projection matrix, we can jointly estimate an albedo parameter $\mathbf{f}_A$ and a lighting parameter $\mathbf{L}$ whose decoded texture can reconstruct the original image. To accomplish this, we use SGD on $\mathbf{f}_A$ and $\mathbf{L}$ with the initial parameters estimated by our encoder $\mathcal{E}$. For the linear model, Zhu *et al*. [193] fitting results of Basel albedo using Phong illumination model [122] is used. As in Fig. 4.6, nonlinear model significantly outperforms the Basel

Figure 4.6: Texture representation power comparison. Our nonlinear model can better reconstruct the facial texture.

Face model. Despite, being close to the original image, Tran *et al*. [161] model reconstruction results are still blurry. Using global local-based network architecture ("+GL") with the same loss functions helps to bring the image closer to the input. However, these models are still constrained by regularizations on the albedo. By learning using proxy technique ("+Proxy"), our model can learn more realistic albedo with more high frequency details on the face. This conclusion is further supported with quantitative comparison in Tab. 4.1. We report the averaged $l_{2,1}$ reconstruction error over the face portion of each image. Our model achieves the lowest averaged reconstruction error among four models, 0.0363, which is a 15% error reduction of the recent nonlinear 3DMM work [161].

Shape. Similarly, we also compare models' power to represent real-world 3D scans. Using ten 3D face meshes provided by [121], which share the same triangle topology with us, we can

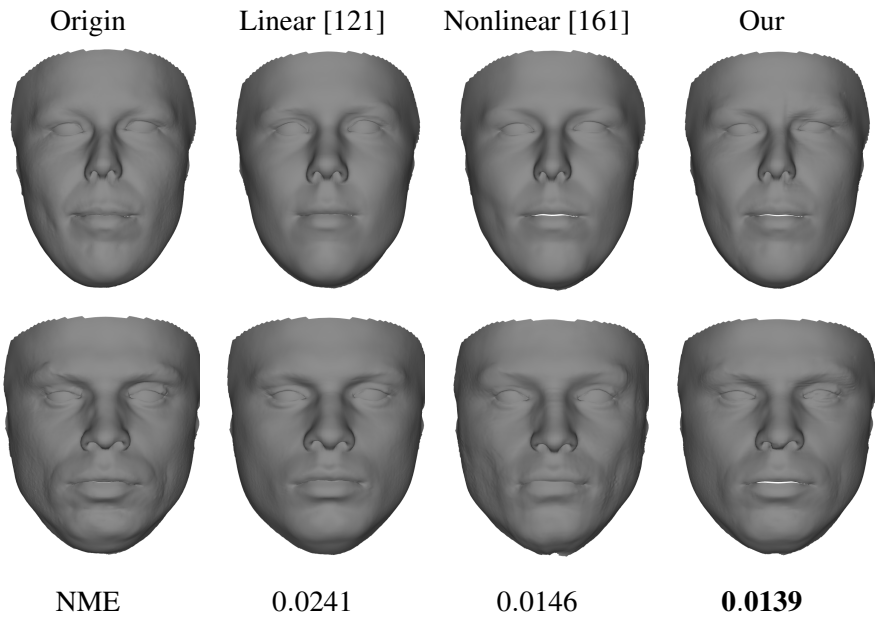


Figure 4.7: Shape representation power comparison. Given a 3D shape, we optimize the feature $\mathbf{f}_S$ to approximate the original one.

optimize the shape parameter to generate, through the decoder, shapes matching the groundtruth scans. The matching criterion is defined based on both vertex distances (Euclidean) and surface normal direction (cosine distance), which empirically improves fidelity of reconstructed meshes compared to optimizing vertex distances only. Fig. 4.7 shows the visual comparisons between different reconstructed meshes. Our reconstructions closely match the face shapes details. To quantify the difference, we use NME — averaged per-vertex Euclidean distances between the recovered and groundtruth meshes, normalized by inter-ocular distances. The proposed model has a significantly smaller reconstruction error than the linear model, and is also smaller than the nonlinear model by Tran *et al*. [161] (0.0139 vs. 0.0146 [161], and 0.0241 [121])

4.3.3 Identity-Preserving

We explore the effect of our proposed 3DMM on preserving identity when reconstructing face images. Using DR-GAN [163], a pretrained face recognition network, we can compute the cosine

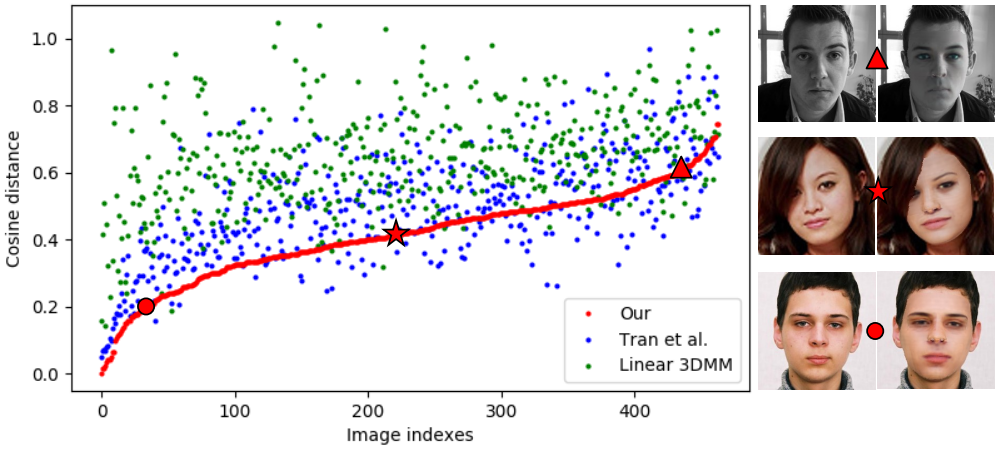


Figure 4.8: The distance between the input images and their reconstruction from three models. For better visualization, images are sorted based on their distance to our model's reconstructions.

distance between the input and its reconstruction from different models. Fig. 4.8 shows the plot of these score distributions. At each horizontal mark, there are exactly three points presenting distances between an image with its reconstructions from three models. Images are sorted based on the distance to our reconstruction. For the majority of the cases (77.2%), our reconstruction has the smallest difference to the input in the identity space.

4.3.4 3D Reconstruction

Using our model $\mathcal{D}_S, \mathcal{D}_A$, together with the model fitting CNN $\mathcal{E}$, we can decompose a 2D photograph into different components: 3D shape, albedo and lighting (Fig. 4.9). Here we compare our 3D reconstruction results with different lines of works: linear 3DMM fitting [153], nonlinear 3DMM fitting [152, 161] and approaches beyond 3DMM [63, 139].

For linear 3DMM model, the representative work, MoFA by Tewari *et al*. [153, 151], learns to regress 3DMM parameters in an unsupervised fashion. Even being trained on in-the-wild images, it is still limited to the linear subspace, with limited power to recovering in-the-wild texture. This results in the surface shrinkage when dealing with challenging texture, i.e., facial hair as discussed

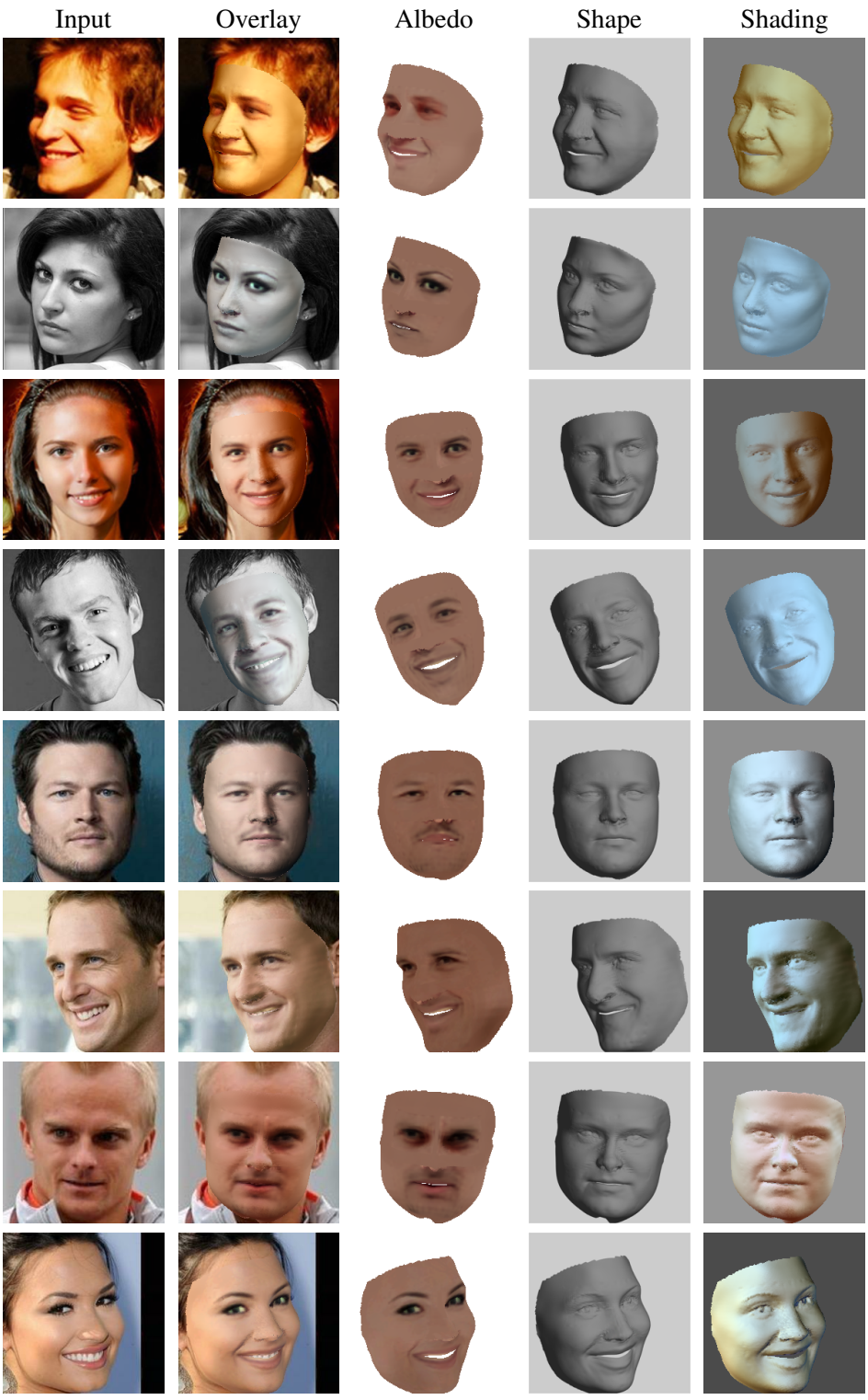
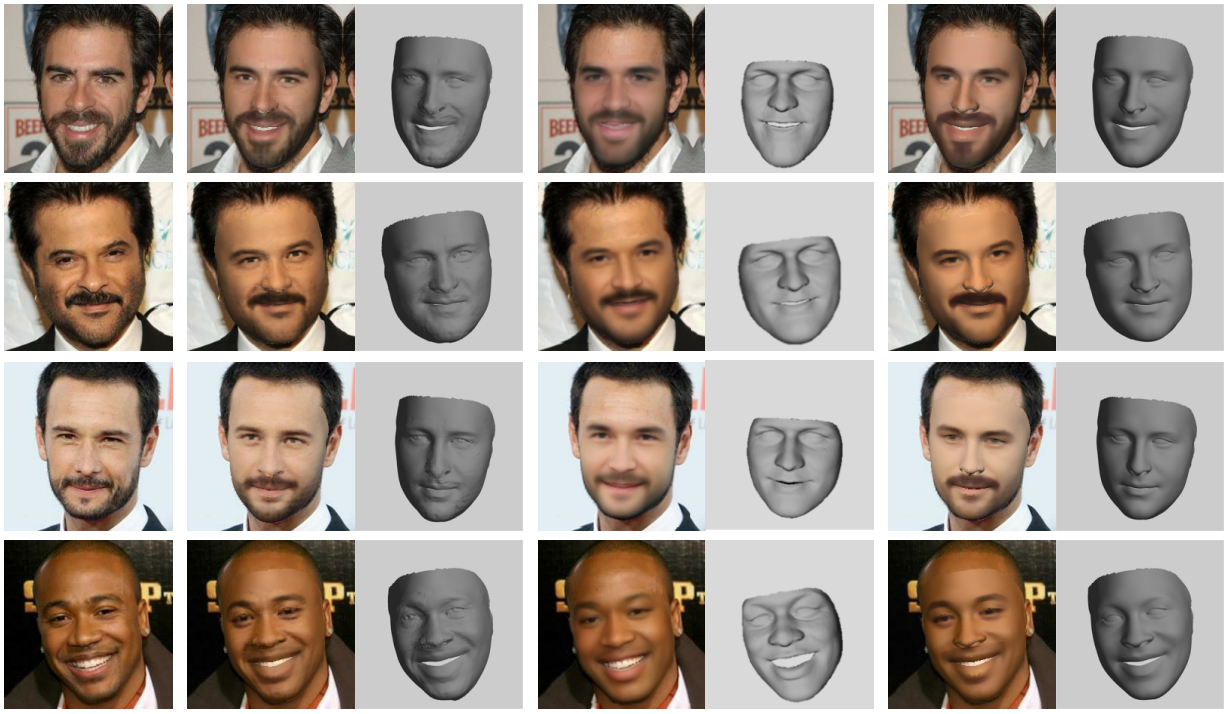


Figure 4.9: 3DMM fits to faces with diverse skin color, pose, expression, lighting, and faithfully recovers these cues.

Input Our Tewari17

Figure 4.10: 3D reconstruction comparison to Tewari *et al*. [153].



Input Our Tewari18 Tran18a

Figure 4.11: 3D reconstruction comparisons to nonlinear 3DMM approaches by Tewari *et al*. [152] or Tran and Liu [161]. Our model can reconstruct face images with higher level of details. Please zoom-in for more details. Best view electronically.

in [152, 160, 161]. Besides, even with regular skin texture their reconstruction is still blurry and has less details compared to ours (Fig. 4.10).

The most related work to our proposed model is Tewari *et al*. [152], Tran and Liu [161], in

Figure 4.12: 3D reconstruction comparisons to Sela *et al*. [139] or Tran *et al*. [159], which go beyond latent space representations.

which 3DMM bases are embedded in neural networks. With more representation power, these models can recover details which the traditional 3DMM usually can't, i.e. make-up, facial hair. However, the model learning process is attached with strong regularization which limits their ability to recover high frequency details of the face. Our propose model enhances the learning process in both learning objective and network architecture to allow higher-fidelity reconstruc-

tions (Fig. 4.11).

To improve 3D reconstruction quality, many approaches also try to move beyond the 3DMM such as Richardson *et al*. [129], Sela *et al*. [139] or Tran *et al*. [159]. The current state-of-the-art 3D monocular face reconstruction method by Sela *et al*. [139] using a fine detail reconstruction step to help reconstructing high fidelity meshes. However, their first depth map regression step is trained on synthetic data generated by the linear 3DMM. Besides domain gap between synthetic and real, it faces a more serious problem of lacking facial hair in the low-dimension texture. Hence, this network's output tends to ignore these unexplainable regions, which leads to failure in later steps. Our network is more robust in handling these in-the-wild variations (Fig. 4.12). The approach of Tran *et al*. [159] share a similar objective with us to be both robust and maintain high level of details in 3D reconstruction. However, they use an over-constrained foundation which loses personal characteristics of the each face mesh. As a result, the 3D shapes look similar across different subjects (Fig. 4.12).

4.3.5 Face editing

Decomposing face image into individual components give us ability to edit the face by manipulating any component. Here we show three examples of face editing using our model.

Relighting. First we show an application to replacing the lighting of a target face image using lighting from a source face (Fig. 4.13). After estimating the lighting parameters $\mathbf{L}_{\text{source}}$ of the source image, we render the transfer shading using the target shape $\mathbf{S}_{\text{target}}$ and the source lighting $\mathbf{L}_{\text{source}}$. This transfer shading can be used to replace the original source shading. Alternatively, value of $\mathbf{L}_{\text{source}}$ can be arbitrarily chosen based on the SH lighting model, without the need of source images. Also, here we use the original texture instead of the output of our decoder to

Figure 4.13: Lighting transfer results. We transfer the lighting of source images (first row) to target images (first column). We have similar performance compare to the state-of-the-art method of Shu *et al*. [143] despite being orders of magnitude faster (150 ms vs. 3 min per image).

maintain image details.

Attribute Manipulation. Given faces fitted by 3DMM model, we can edit images by naive modifying one or more elements in the albedo or shape representation. More interestingly, we can even manipulate the semantic attribute, such as growing beard, smiling, etc. The approach is similar to learning attribute embedding in Sec. 3.3.2. Assuming, we would like to edit appearance

Figure 4.14: Growing mustache editing results. The first collumn shows original images, the following collumns show edited images with increasing magnitudes. Comparing to Shu *et al*. [144] results (last row), our edited images are more realistic and identity preserved.

only. For a given attribute, e.g., beard, we feed two sets of images with and without that attribute $\{\mathbf{I}_i^p\}_{i=1}^n$ and $\{\mathbf{I}_i^n\}_{i=1}^n$ into our encoder to obtain two average parameters $\mathbf{f}_A^p$ and $\mathbf{f}_A^n$. Their difference $\Delta\mathbf{f}_A = \mathbf{f}_A^p - \mathbf{f}_A^n$ is the direction to move from the distribution of negative images to positive ones.

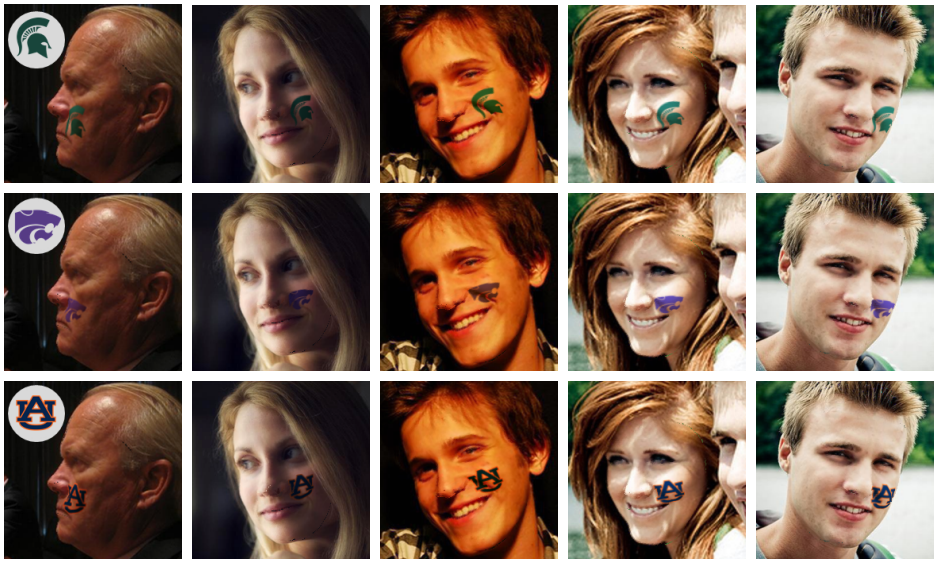


Figure 4.15: Adding stickers to faces. The sticker is naturally added into faces following the surface normal or lighting.

By adding $\Delta\mathbf{f}_A$ with different magnitudes, we can generate modified images with different degree of changes. To achieve high-quality editing with identity-preserved, the final editing result is obtained by adding the residual, the different between the modified image and our reconstruction, to the original input image. This is a critical difference to Shu *et al*. [144] to improve results quality (Fig. 4.14).

Adding Sticker. With more precise 3D face mesh reconstruction, the quality of successive tasks is also improved. Here, we show an application of our model on face editing: adding stickers or tattoos onto faces. Using the estimated shape as well as the projection matrix, we can unwrap the facial texture into the UV space. Thanks to the lighting decomposition, we can also remove the shading from the texture to get the detailed albedo. From here we can directly edit the albedo by adding sticker, tattoo or make-up. Finally, the edited images can be rendered using the modified albedo together with other original elements. Fig. 4.15 shows our editing results by adding stickers into different people's face.

4.4 Conclusions

In realization that the strong regularization and global-based modeling are the roadblocks to achieve high-fidelity 3DMM model, this chapter presents a novel approach to improve the nonlinear 3DMM modeling in both learning objective and network architecture. Hopefully, with insights and findings discussed, this can be a step toward unlocking the possibility to build a model which can capture mid and high-level details in the face. Through which, high-fidelity 3D face reconstruction can be achieved solely by doing model fitting.

Chapter 5

Intrinsic 3D Decomposition, Segmentation, and Modeling Generic Objects

5.1 Introduction

Understanding 3D structure is one of computer vision's fundamental problems. A human has no difficulty understanding the 3D structure of an object upon seeing its 2D image. Even without geometric cues (motion or stereopsis), our visual system can still infer detailed surfaces or plausibly hidden parts. Meanwhile, such a 3D inferring task remains extremely challenging for computer vision systems.

In recent years, with advancements in deep learning, many have shown human-level performance on 2D image understanding, such as object detection [59], recognition [61, 163], segmentation [57, 21]. One of the main reasons for this success is the abundance of annotated data. For majority of 2D understanding tasks, nowadays, there usually be many databases with sufficiently annotated images. Hence, the decent performance can be obtained using end-to-end supervised learning. However, extending this success to supervised learning for 3D inference is far behind

This chapter is adapted from the following work:
[1] Feng Liu, Luan Tran and Xiaoming Liu, "Intrinsic 3D Decomposition and Modelingfor Generic Objects via Colored Occupancy Field" under submission. (Luan Tran and Feng Liu make equal contribution to this work).

due to limited availability of 3D labels.

With the introduction of large 3D Computer Aided Design (CAD) databases like ObjectNet3D [176], ShapeNet [26], majority of recent work on 3D monocular object reconstruction [56, 54, 34] and intrinsic image decomposition [64, 142] rely entirely on synthetic images generated from the CAD models. However, using synthetic data alone has a major drawback. First of all, creating CAD models is not scalable. Making a single 3D object instance is labor extensive and requires expertise in computer graphics. Hence, it's not feasible to build models for *all* available objects. Secondly, there still be an obvious gap between synthetic rendering images and real images even with advanced rendering techniques in computer graphic. Therefore, these methods have limited ability in reconstruction from real-world images.

Meanwhile, there is a large collection of 2D images for any object categories. If those images can be effectively used in either 3D object modeling or learning to fit the model, it could have a great impact on the 3D object reconstruction. Essentially, the reason that real-world 2D images have not been effectively used in generic object 3D reconstruction is the lack of corresponding ground truth 3D shapes for these images, and thus no supervised learning.

Early attempts [89, 164] on learning 3D shape model from 2D photographs in an unsupervised fashion are still limited on exploiting 2D images. Given an input image, they mainly try to learn 3D model to reconstruct 2D silhouette of the object. To learn a better model, multiple views of the same object with ground-truth pose or keypoints annotations are needed. More importantly, they ignore additional monocular cues, *e.g.*, shading, that obtain rich 3D information. One common issue among prior work is the lack of modeling for albedo, one key element in image formulation. As a result, *analysis-by-synthesis* approaches is not applicable to 3D modeling of generic objects.

To address these issues, we propose a novel paradigm to jointly learn a completed and segmented 3D model, consisting of both 3D shape and albedo, as well as a model fitting module to

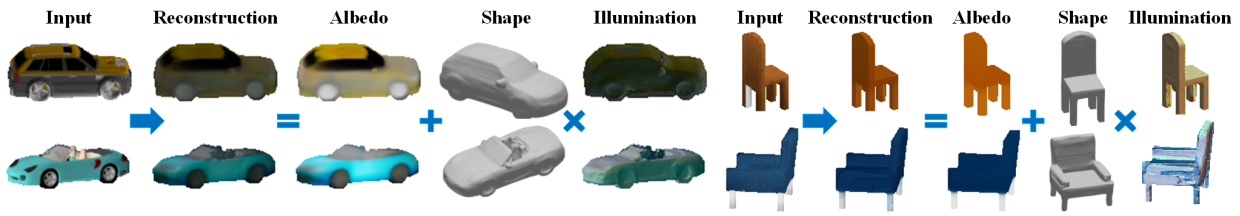


Figure 5.1: This work decomposes a 2D image of genetic objects into albedo, 3D shape, illumination, and camera projection.

estimate the shape, albedo, lighting and camera matrix from 2D images, as in Fig. 5.1. Different from prior 3D reconstruction work, this is the first work modeling both shape and albedo of a generic object, in a *semi-supervised* manner. Modeling albedo, together with estimating the environment lighting condition, enables us to fully exploit the shading cues from 2D images to estimate the 3D shape.

Specifically, considering large intra-class variations in mesh topology, we propose to use *colored occupancy field* to completely represent a 3D object. For every spatial point, the colored occupancy field provides the probability whether it is inside the object and also the RGB value of its albedo. The surface of the object is implicitly represented as the iso-surface at a certain threshold of the occupancy probability. Colored occupancy field theoretically can represent a shape at an arbitrarily high resolution, which only depends on the sampling density of spatial points. Moreover, also due to the lack of consistency in meshes' topology, the dense correspondence between 3D shapes is missing. We propose to jointly model the object part segmentation which exploits its implicit correlation with shape and albedo, and also creates explicit constraints for our model fitting learning.

In summary, the contributions of this chapter include:

◇ We build the first 3D model, that fully models segmented 3D shape, albedo for generic objects using *colored occupancy field* as a representation.

◇ Modeling intrinsic components allows us to not only better exploit visual cues, but also, for the first time, use real images for model training in an unsupervised manner.

◇ Incorporating unsupervised part segmentation enables better constraints to fine-tune the shape and pose estimation.

◇ We demonstrate superior performance on 3D reconstruction of generic objects from a single 2D image.

5.1.1 3D Shape and Albedo Representation

Shape Implicit Field. In contrast to 2D domain, the community has not yet agreed on a 3D representation that is both memory efficient and inferable from data. Recently, a lot of attention is focus on implicit representation, where each shape can be represented by a function $o : \mathbb{R}^3 \rightarrow [0, 1]$. This function takes a spatial location $\mathbf{x} \in \mathbb{R}^3$ as an input and outputs its probability of occupancy [34, 108, 117]. With this implicit representation, the shape can be viewed at an arbitrary high resolution. Another appealing property of this representation is that the surface normal can be analytically computed using the spatial derivative $\frac{\delta \mathcal{D}_S(\mathbf{f}_S, \mathbf{x})}{\delta \mathbf{x}}$ via back-propagation through the network. This is helpful for successive analysis tasks such as rendering.

As in [34, 108, 117], leveraging deep neural networks, a family or an instance of shape functions can be represented using a decoder network $\mathcal{D}_S$ and each shape $\mathbf{S}$ is encoded by a latent representation $\mathbf{f}_S \in \mathbb{R}^{d_S}$ (Fig 5.2.a):

$$\mathcal{D}_S : \mathbb{R}^3 \times \mathbb{R}^{d_S} \rightarrow [0, 1]. \tag{5.1}$$

The shape decoder's architecture follows BAE-NET [33]. BAE-NET is a joint shape co-segmentation and reconstruction network, which takes shape latent representation $\mathbf{f}_S$ and a spatial

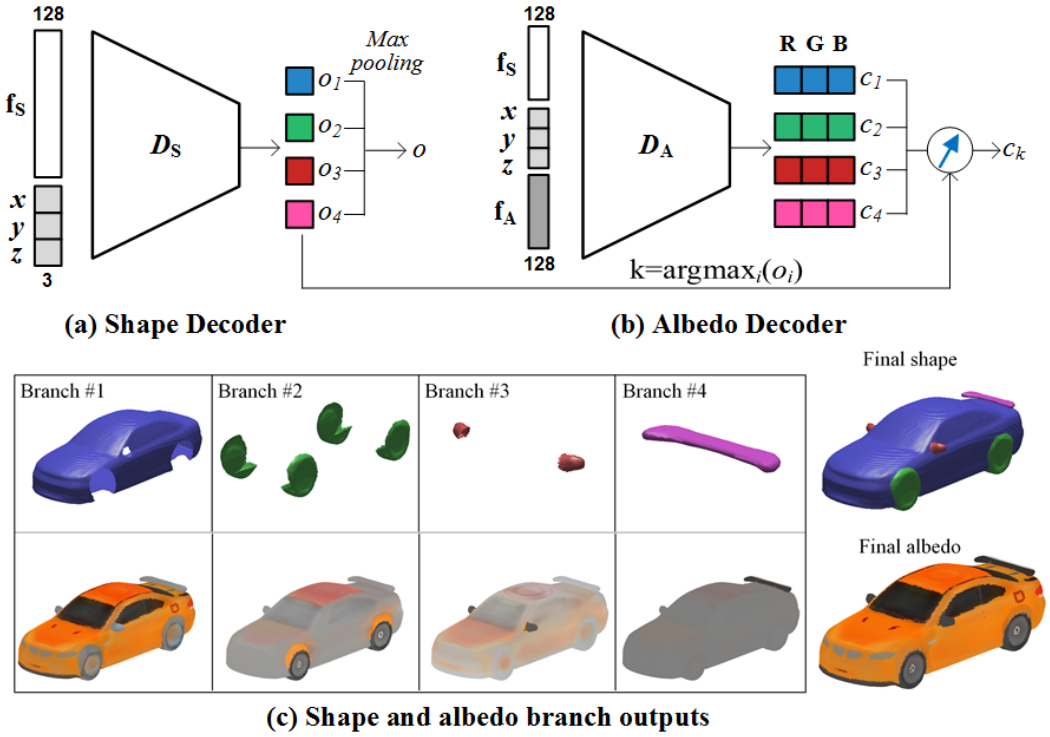


Figure 5.2: Shape and albedo decoder networks. Shape decoder $\mathcal{D}_S$ takes a shape latent representation $\mathbf{f}_S$ and a spatial point $\mathbf{x}$=(x,y,z) and produces the implicit field for each branch. The final output layer groups the branch outputs, via max pooling, to form the spatial probability of occupancy. Albedo decoder $\mathcal{D}_A$ receives both latent representations $\mathbf{f}_S, \mathbf{f}_A$ and estimates the albedo colors of 4 branches, one of which is selected by the shape branch/segmentation and returned as the final albedo color of $\mathbf{x}$.

point $\mathbf{x}$ as inputs. It is composed of 3 fully connected layers each followed by a LeakyReLU, except the final output (*Sigmoid*). The final layer gives the implicit field for four branches (o_1, o_2, o_3, o_4). Finally, a max pooling operator on branch outputs results in the final implicit field o. BAE-NET is much shallower and thinner compared to IM-NET [34], since it cares more about the quality of segmentation rather than reconstruction. We propose to integrate the shape into albedo learning, which is shown to benefit both segmentation and reconstruction.

Albedo Implicit Field. For a completed model, each vertex on the shape surface is assigned a RGB albedo color. Extending the idea of the occupancy field to albedo, we propose to represent the albedo as a *colored field*. The albedo decoder $\mathcal{D}_A$ returns an RGB color for any spatial location $\mathbf{x} \in \mathbb{R}^3$. One approach for the colored field is naïvely using a single albedo latent representation $\mathbf{f}_A$

to represent a colored shape, *i.e.*, $\mathcal{D}_A(\mathbf{f}_A, \mathbf{x})$. However, it puts a redundant burden to $\mathbf{f}_A$ to encode the object geometry, *e.g.*, the position of the tire, and body of a car. Hence, we propose to take the shape latent vector $\mathbf{f}_S$ as an additional input to the albedo decoder $\mathcal{D}_A(\mathbf{f}_A, \mathbf{f}_S, \mathbf{x})$ (Fig 5.2.b):

$$\mathcal{D}_A : \mathbb{R}^3 \times \mathbb{R}^{d_T} \times \mathbb{R}^{d_S} \rightarrow \mathbb{R}^3. \tag{5.2}$$

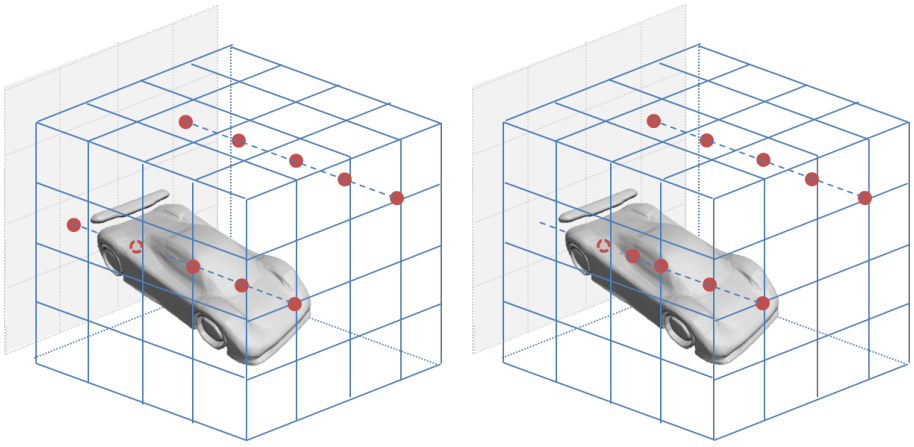
For simplicity we will omit $\mathbf{f}_S, \mathbf{f}_A$ in $\mathcal{D}_*$ in later sections.

The albedo decoder has a similar architecture as the shape decoder, with a few differences. The input to the network has an additional vector, albedo representation $\mathbf{f}_A$. The output is applied *Tanh* activation. Also, the third layer gives the color field for four branches (c_1, c_2, c_3, c_4) and each with 3 channels. At every spatial location, the final color is c_k, where $k = \arg\max_i(o_i)$ (Fig. 5.2). One key motion for integrating shape segmentation into albedo decoder is that, different parts of an object often differ in *both* shape and texture. The four albedo branches essentially represent the *dominant* albedo colors of the object, whose learning will in turn encourage the shape decoder to segment parts that differ not only in shape, but also in dominant albedo.

5.1.2 Physis-Based Rendering

To render an object image from shape, albedo, represented by latent vectors $\mathbf{f}_S$, $\mathbf{f}_A$, as well as lighting $\mathbf{L}$ and projection matrix $\mathbf{P}$, we first find a set of $W \times H$ surface points corresponding to each pixel. Then the RGB color of each pixel is computed via a lighting model using lighting parameters $\mathbf{L}$ and decoder outputs.

Camera model. We assume a full perspective camera model. Any spatial points $\mathbf{x}$ in the 3D world space can be projected in 2D by a multiplication between a projection matrix $\mathbf{P}$ and its homogeneous coordinates representation,



(a) Linear Search (b) Linear-Binary Search

Figure 5.3: Ray tracing for surface points detection. In Linear search, candidates (red points) are uniformly distributed in the grid. In Linear-Binary search, after the first point inside the object found, Binary search will be used between the last outside point and current inside point for all remaining iterations.

$$\mathbf{u} = \mathbf{P}\,[\mathbf{x}, 1]^T\,, \tag{5.3}$$

where $\mathbf{P}$ is a 3×4 full perspective projection matrix.

Essentially, $\mathbf{P}$ can be extended to its 4×4 version with zero translation in z-direction, With an abuse in annotation in homogeneous coordinates, relation between 3D points $\mathbf{x}$ and its camera space projection $\mathbf{u}$ can be written as:

$$\mathbf{u} = \mathbf{P}\mathbf{x}, \quad \text{and} \quad \mathbf{x} = \mathbf{P}^{-1}\mathbf{u}. \tag{5.4}$$

Surface point detection. To render a 2D image, for each ray from the camera to the pixel $j = (u, v)$, we select one "surface point". Here, a surface point is defined as the first interior point ($\mathcal{D}_S(\mathbf{x}) > \tau$) or the outerior point with largest $\mathcal{D}_S(\mathbf{x})$ in case the ray doesn't hit the object.

For efficient network training, instead of finding exact surface points, we approximate them using Linear search or Linear-Binary search (Fig. 5.3).

Intuitively, with the distance margin error of ε, in Linear search, from an initial location in

the object boundary, we evaluate $\mathcal{D}_S(\mathbf{x})$ for all spatial point candidates $\mathbf{x}$ with step size of ε. In Linear-Binary search, after the first interior point is found, as $\mathcal{D}_S(\mathbf{x})$ is a continuous function, a Binary search can be used to better approximate the surface point.

For better parallelization, the number of points evaluated on each ray is the same. In this case, Linear-Binary search doesn't result in speed up but leads to better approximation of surface points, hence better render quality.

Image formation. We assume distant low-frequency illumination and a purely Lambertian surface reflectance. Hence the incoming radiance can be approximated via Spherical Harmonics (SH) basis functions $\mathbf{H}_b : \mathbb{R}^3 \to \mathbb{R}$, and controlled by coefficients $\mathbf{L}$. At the pixel j with corresponding surface point $\mathbf{x}_j$, the image color value is computed as a product of albedo $\mathbf{A}$ and shading $\mathbf{C}$:

$$\mathbf{I}_j = \mathbf{A}_j.\mathbf{C}_j = \mathbf{A}_j. \sum_{b=1}^{B^2} \gamma_b \mathbf{H}_b(\mathbf{n}_j) \tag{5.5}$$

$$= \mathcal{D}_A(\mathbf{x}_j). \sum_{b=1}^{B^2} \gamma_b \mathbf{H}_b \left(\sigma \left(\frac{\delta \mathcal{D}_S(\mathbf{x}_j)}{\delta \mathbf{x}_j} \right) \right), \tag{5.6}$$

where $\mathbf{n}_j = \sigma \left(\frac{\delta \mathcal{D}_S(\mathbf{x}_j)}{\delta \mathbf{x}_j} \right)$ is the L_2-normalized surface normal at $\mathbf{x}_j$, and $\sigma()$ is a vector normalization function. We use $B = 3$ SH bands, which leads to $B^2 = 9$ coefficients in $\mathbf{L}$ for each of three color channels.

5.1.3 Model Learning

Our model is designed to learn from real-world 2D images. However, in addition we also need to learn shape prior from 3D CAD models, due to inherent ambiguity in inverse problems. We first describe learning from 2D images, and then learning from CAD models.

5.1.3.1 Unsupervised Joint Modeling and Fitting

Given a set of 2D images, without corresponding ground truth 3D shape, we define the loss function as:

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \lambda_{\text{sil}}\mathcal{L}_{\text{sil}} + \lambda_{\text{fea-const}}\mathcal{L}_{\text{fea-const}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}, \tag{5.7}$$

where $\mathcal{L}_{\text{img}}$ is the photometric loss, $\mathcal{L}_{\text{sil}}$ enforces consistence between predicted silhouette and ground truth silhouette, and $\mathcal{L}_{\text{fea-const}}$ is the local feature consistency loss, $\mathcal{L}_{\text{reg}}$ consists of different regularization terms.

Silhouette Consistency Loss. Given the object's silhouette mask $\mathbf{M}$ for each image, obtained by an off-the-shell segmentation method [21], the silhouette consistency loss is:

$$\mathcal{L}_{\text{sil}} = \frac{1}{W \times H} \sum_{j=1}^{W \times H} \mathcal{L}\left(\mathcal{D}_S(\mathbf{f}_S, \mathbf{x}_j), o_j\right) \tag{5.8}$$

$$= \frac{1}{W \times H} \sum_{j=1}^{W \times H} \mathcal{L}\left(\mathcal{D}_S(\mathcal{E}_S, \mathcal{E}_P^{-1}\mathbf{u}_j), o_j\right). \tag{5.9}$$

With the occupancy field, the target value o_j is defined as $o_j = 0.5$ if $\mathbf{M}_j = 1$, otherwise $o_j = 0$.

Here, we also analyze how our silhouette loss differs to prior work. If 3D shape is represented as a mesh, there is no gradient when comparing two binary masks, unless the predicted silhouette is expensively approximated as in Soft rasterizer [89]. If the shape is represented by a voxel, the loss can provide gradient to adjust voxel occupancy predictions, but not the object orientation [164]. Our loss can update both occupancy field, camera projection estimation (Eqn. 5.9).

Photometric Loss. To enforce similarity between our reconstruction and input, we use a L_1 loss on the foreground:

$$\mathcal{L}_{\text{img}} = \frac{1}{|\mathbf{M}|} \left\| (\hat{\mathbf{I}} - \mathbf{I}) \odot \mathbf{M} \right\|_1. \tag{5.10}$$

To our best knowledge, this is the first work on generic 3D object modeling that can fully exploit the RGB color information to supervise the shape learning rather than just silhouette guidance [89]. This is only possible due to two specific designs of our approach. 1) We learn the completed model including albedo. 2) The shape implicit representation (contrast to voxel) provides accurate, efficient surface normal computation, allows shading decomposition.

Local Feature Consistency Loss. We propose a novel local feature consistency loss based on the 3D segmentation provided by the shape decoder. We first select q boundary points $\mathbf{U}_{3D} \in \mathbb{R}^{q\times 3}$ from all pairs of neighboring segments based on the shape decoder branches. Then these 3D points are projected to 2D locations $\mathbf{U}_{2D} \in \mathbb{R}^{q\times 2}$ on the image plane using the estimated projection matrix $\mathbf{P}$. Similar to [178], we retrieve features on each feature map using the location $\mathbf{U}_{2D}$ and form the local image features $\mathbf{F} \in \mathbb{R}^{q\times 256}$, where 256 is the feature dimension. Finally, we perform PCA to obtain the engenvector associated with the largest eigenvalue ($\mathbf{v} \in \mathbb{R}^{1\times 256}$), which describes the largest variation among the visual features of q points. Despite the different colors of two images of the same object category, we assume that this major variation is similar. Thus, we define the local feature consistency loss as:

$$\mathcal{L}_{\text{fea-const}} = \frac{1}{|B|} \sum_{(i,j)\in B} \left\| \mathbf{v}_i - \mathbf{v}_j \right\|_1 , \tag{5.11}$$

where B is the training batch.

Regularization. We define two regularization terms to constrain the learning.

Albedo local constancy: following Retinex theory [82] which assumes albedo to be piecewise constant, we enforce the gradient sparsity in two directions, similar to [144]:

$$\mathcal{L}_{\text{alb-const}} = \sum_{t\in\mathcal{N}_j} \omega(j,t) \left\| \mathbf{A}_j - \mathbf{A}_t \right\|_2^p , \tag{5.12}$$

where $\mathcal{N}_j$ represents pixel j's set of 4 neighbor pixels. With the assumption that pixels with the same chromaticity (i.e., $\mathbf{c}_j = \mathbf{I}_j/|\mathbf{I}_j|$) are more likely to have the same albedo, we set the constant weight $\omega(i,t) = \exp\left(-\alpha \left\|\mathbf{c}_j - \mathbf{c}_t\right\|\right)$, where the color is referenced from the input image. In our experiment we set $\alpha = 15$ and $p = 0.8$ as in [107].

Batch-wise White Shading: Due to ambiguity in the magnitude of lighting, and therefore the intensity of shading, it is necessary to incorporate constraints on the shading magnitude to prevent the network from generating arbitrary bright/dark shading. To handle these ambiguities, we use a Batch-wise White Shading [144] constraint on shading:

$$\mathcal{L}_{\text{bws}} = \left\| \frac{1}{m} \sum_{j=1}^{m} \mathbf{C}_j^{s(r)} - c \right\|_1, \tag{5.13}$$

where $\mathbf{C}_j^{s(r)}$ is a red channel diffuse shading of pixel j, m is the number of foreground pixels in a training batch. c is a constant for the target average shading, which is set to 1. The same constraint is applied for blue and green channels.

5.1.3.2 Supervised Prior Learning with Synthetic Image

The CAD model helps to learn the shape prior and provide supervision in training.

Learning Shape and Albedo Decoder. To learn the shape and albedo model (decoders), we adopt widely used techniques which is training encoder-decoder networks [34, 50]. Here the input to the encoder is a *colored* voxel, and the encoder $\mathcal{E}'$ is 3D CNN. Voxel is picked over 2D images as it contains all shape information which better eliminates ambiguity for the encoding process.

Given a dataset of N models, each of which can be represented as a colored 3D occupancy voxel $\mathbf{V}$. Equivalently, each model can also be represented with K spatial points $\mathbf{x} \in \mathbb{R}^3$ and its occupancy label $o \in [0,1]$ and albedo c. This model learning objective is written as:

$$\underset{\mathcal{D}_S,\mathcal{D}_A,\mathcal{E}'}{\arg\min} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \left(\mathcal{L}(\mathcal{D}_S(\mathcal{E}'_S(\mathbf{V}_i), \mathbf{x}_j), o_j) + \mathcal{L}(\mathcal{D}_A(\mathcal{E}'_S(\mathbf{V}_i), \mathcal{E}'_A(\mathbf{V}_i), \mathbf{x}_j), c_j) \right) \right). \tag{5.14}$$

The loss $\mathcal{L}$ (softmax cross-entropy or L_p) penalizes deviation of the network prediction from the actual value o_j, c_j.

We also adopt progressive training techniques [34], to train our model on gradually increasing resolution data. Since the model structure doesn't change when switching training data of different resolutions, thus higher-resolution models can be trained with pre-trained weights on low-resolution data. Progressive training stabilizes and significantly speeds up the training.

Learning Image Encoder. For each CAD model, we render multiple images of the same object with different poses and lighting conditions. Here each training sample is a triplet of voxel, 2D image and its corresponding ground truth projection matrix $(\mathbf{V}, \mathbf{I}, \widetilde{\mathbf{P}})$. They can be used as an additional supervision for our encoder and decoders.

$$\mathcal{L}_{\mathrm{S}} = \left\| \mathcal{E}_S(\mathbf{I}) - \mathcal{E}'_S(\mathbf{V}) \right\|_2^2, \tag{5.15}$$

$$\mathcal{L}_{\mathrm{A}} = \left\| \mathcal{E}_A(\mathbf{I}) - \mathcal{E}'_A(\mathbf{V}) \right\|_2^2, \tag{5.16}$$

$$\mathcal{L}_{\mathrm{P}} = \left\| \mathcal{E}_P(\mathbf{I}) - \widetilde{\mathbf{P}} \right\|_2^2, \tag{5.17}$$

The ground truth latent representations are obtained from the ground truth voxel ($\mathcal{E}'(\mathbf{V})$).

5.1.4 Implementation Details

5.1.4.1 Model training

The full model is trained in three stages. First, the shape and albedo decoder is trained with colored voxel data. Then the encoder is trained with 2D synthetic images as inputs. Both supervised and

unsupervised losses are used in this stage. Finally, the model fitting module (encoder and albedo decoder) can be finetuned using real images with unsupervised losses. We empirically found that, the real images training has incremental benefit on finetuning the shape decoder. But it significant improves the generalization ability of our encoder on fitting model to real images. Hence, we decide to fix the weight of the shape decoder after the first stage. The encoder is a modified ResNet-18, while decoders are 3 layers MLPs [33]. Weights are initialized from a normal distribution with a standard deviation of 0.02. Adam optimizer is used with a learning rate of 0.0001 in all stages.

5.1.5 Network Structure

Colored Voxel Encoder. To learn the shape and albedo models (prior) simultaneously, our voxel encoder requires colored voxels as input. We obtain color voxelization for the ShapeNet 3D mesh models by the work [29]. Fig. 5.4 shows two examples of color voxelization. The voxel encoder architecture (Table 5.1) is 3D CNN, which is adopted from [34, 33].

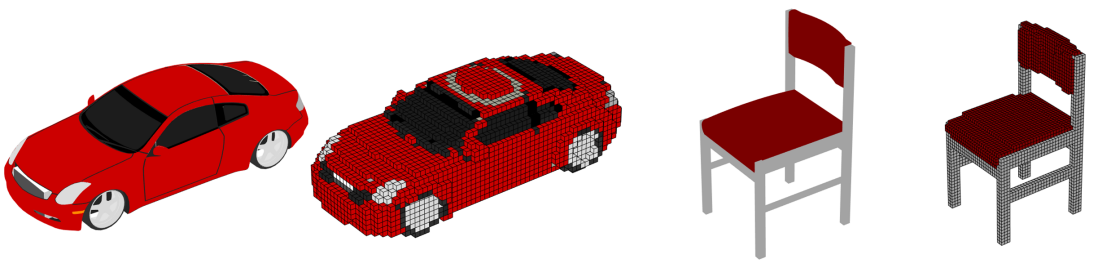


Figure 5.4: Color voxelization of ShapeNet models. Original 3D mesh (left) and 64^3 colored voxel (right).

Shape and Albedo Decoders. The shape decoder architecture is followed the work of [33] (unsupervised case). The network takes shape latent representation $\mathbf{f}_S$ and a spatial point (x, y, z) as inputs. It is composed of 3 fully connected layers each of which are applied with Leaky ReLU, except the final output is applied *Sigmoid* activation (Fig. 5.5). The albedo decoder architecture

Table 5.1: Colored voxel encoder network structure.

Layer	Kernel size	Stride	Activation function	Output size (d1,d2,d3,C)
input	-	-	-	$(64,64,64,3)$
conv3d	$(4,4,4)$	$(2,2,2)$	LReLU	$(32,32,32,32)$
conv3d	$(4,4,4)$	$(2,2,2)$	LReLU	$(16,16,16,64)$
conv3d	$(4,4,4)$	$(2,2,2)$	LReLU	$(8,8,8,128)$
conv3d	$(4,4,4)$	$(2,2,2)$	LReLU	$(4,4,4,256)$
conv3d	$(4,4,4)$	$(1,1,1)$	Sigmoid	$(1,1,1,256)$
$\mathbf{f}_A$	-	-	-	128
$\mathbf{f}_S$	-	-	-	128

is similar, with only two differences. The input to the network has an additional vector, albedo latent representation $\mathbf{f}_A$. The output is applied *Tanh* activation. Fig. 5.6 depicts the albedo decoder architecture.

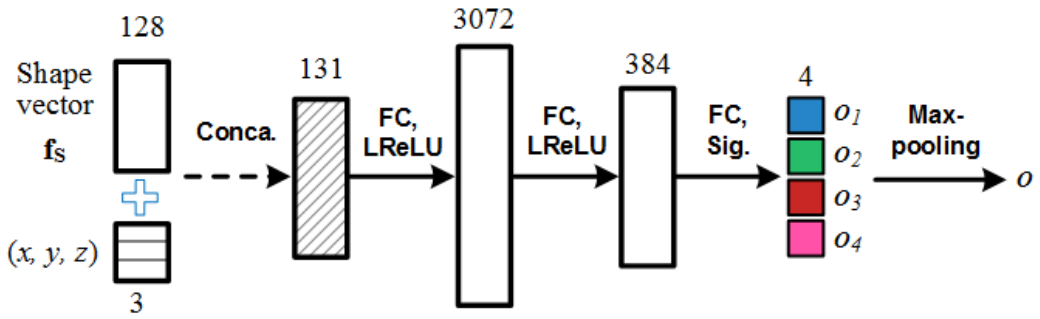


Figure 5.5: The shape decoder network is composed of 3 fully connected layers, denotes as "FC". The shape latent vector (128-dim) is concatenated, denoted "+", with the xyz query, making a 131-dim vector, and is provided as input to the first layer. The Leaky ReLU activation is applied to the fist 2 FC layers while the final value is obtained with *Sigmoid* activation denoted as "Sig.".

Local Feature Extraction. We first select q boundary points $\mathbf{U}_{3D} \in \mathbb{R}^{q\times 3}$ from all pairs of neighboring segments based on the shape decoder branches. Then these 3D points are projected to 2D locations $\mathbf{U}_{2D} \in \mathbb{R}^{q\times 2}$ on the image plane using the estimated projection matrix $\mathbf{P}$. Fig. 5.7 shows one example of the selected visible points. We set $q = 50$ in our experiment.

The image encoder is a modified ResNet-18. Table 5.2 illustrates the detail network architecture. Given the $3D$ points $\mathbf{U}_{3D}$, we identity the projected location $\mathbf{U}_{2D}$ on the feature map layers of the encoder. Here, we concatenate features from the outputs of *conv*1, *conv*2 and *conv*3 (see

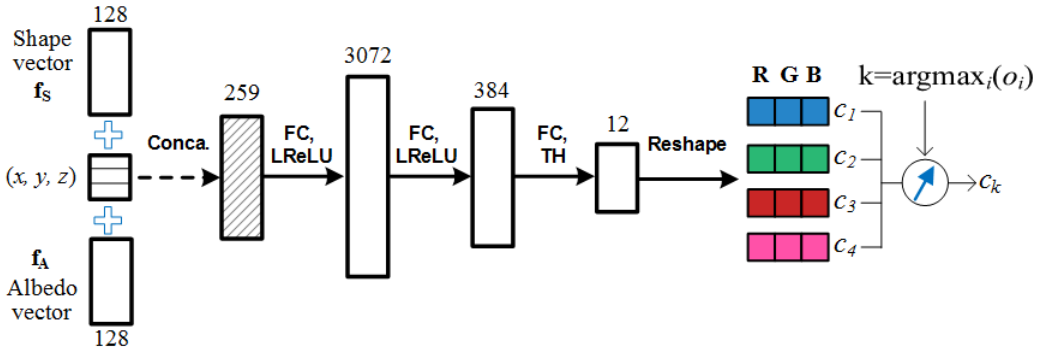
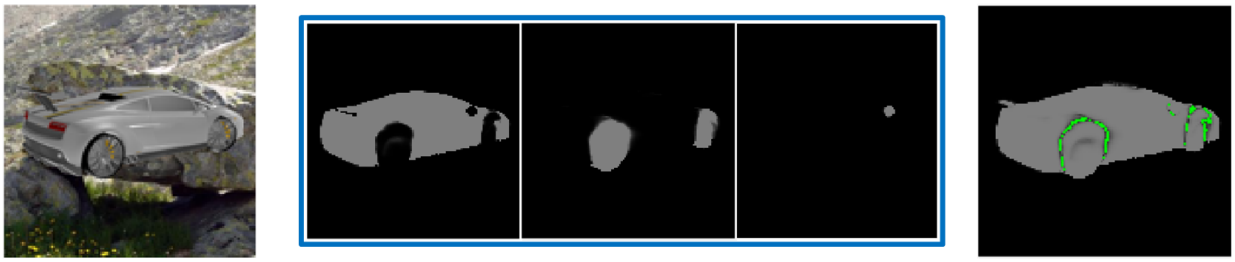


Figure 5.6: The albedo decoder network is also composed of 3 fully connected layers. Specifically, it takes the point coordinate (x, y, z), along with shape and albedo feature vectors, and outputs the RGB color value. 'TH' denotes *Tanh* activation.



Input image | Shape parts on image plane | Selected points

Figure 5.7: One example of boundary points selection for local feature extraction.

Table 5.2) to get the local features $\mathbf{F} \in \mathbb{R}^{q \times 256}$ (size: 64+64+128) of the point. Here, we reshape the feature maps to the original image size with bilinear interpolation.

To better illustrate the efficiency of the proposed local feature consistence constraint for pose and shape estimation, we fist select the boundary points for 20 pairs of input images based on the ground-truth camera parameter and shape parts information. Then we disturb the selected points with additive zero-mean Gaussian noise. Fig. 5.8 presents the average local feature distance under noise of different standard deviations. As shown by the results, the local feature distance is sensitive to the noise in selected points, which means the local feature consistence loss enables the framework to generate better camera and shape parameter so that the corresponding semantic points can be obtained.

Table 5.2: Image encoder network structure (slightly modified from ResNet-18).

Layer	Kernel size	Stride	Activation function	Input size	Output size
input	-	-	-	-	$(128, 128, 3)$
conv1	$(7, 7)$	$(2, 2)$	Max-pooling, BN, LReLU	$(128, 128, 3)$	$(32, 32, 64)$
conv2 (ResNet block)	$(3, 3)$	-	-	$(32, 32, 64)$	$(32, 32, 64)$
conv3 (ResNet block)	$(3, 3)$	-	-	$(32, 32, 64)$	$(16, 16, 128)$
conv4 (ResNet block)	$(3, 3)$	-	-	$(16, 16, 128)$	$(8, 8, 256)$
conv5 (ResNet block)	$(3, 3)$	-	-	$(8, 8, 256)$	$(4, 4, 512)$
average pool	$(4, 4)$	-	-	$(4, 4, 512)$	$(1, 1, 512)$
FC_l	-	-	-	512	27
FC_p	-	-	-	512	12
FC_{shape}	-	-	Sigmoid	512	128
FC_{albedo}	-	-	Sigmoid	512	128

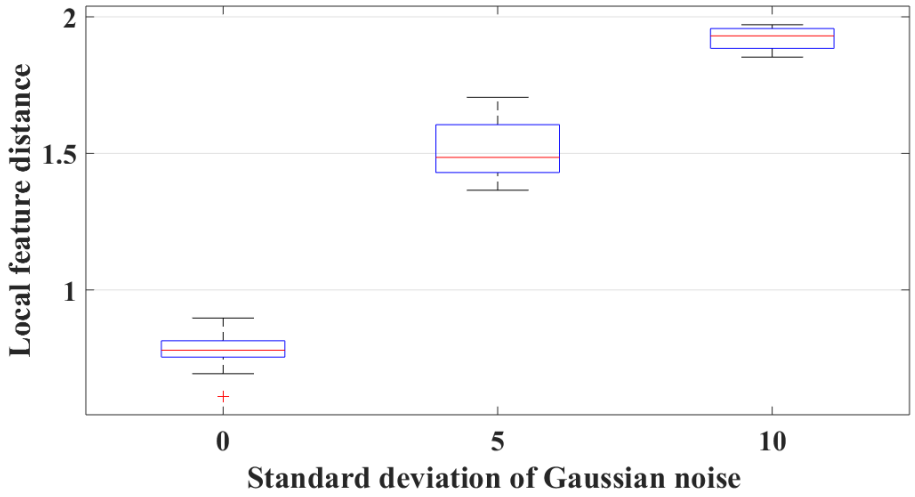


Figure 5.8: Local feature distance under noise of different standard deviations.

5.2 Experimental Results

We study four aspects of proposed methods, in terms of ablation study, unsupervised segmentation, single-view 3D reconstruction on synthetic, and real-world images.

5.2.1 Experiment Setup

Data. For evaluation of 3D shape reconstruction, we use the ShapeNet Core v1 dataset [26]. It is composed of CAD models of objects in various categories. Following the settings of [56], we use the same training/testing split. While using the same test set, we render training data ourselves,

Figure 5.9: 3D reconstruction using models learned with (third row) and without real image (second row). Higher quality reconstruction is observed in the bottom.

adding lighting and real-world pose variations (pose distribution from Pascal 3D+ [177] training data). This helps us to leverage the shading cue to better learn the model as well as model fitting to real-world images.

We use images from Pascal 3D+ database [177], into our unsupervised model training step. Pascal 3D+ augments 12 rigid categories of Pascal VOC 2012 [44] with 3D annotations. We select the same 5 categories (plane, car, chair, couch and table) with our synthetic data. The training subset of from Pascal 3D+ images we considered after filtering occluded instances, which would affect the image decomposition training process.

Metrics. We adopt the standard 3D reconstruction metric: IoU and Chamfer Distance (CD) [108] for evaluation. To compare with methods that output point clouds, we first use marching cubes to obtain meshes from 256^3-voxelized models. For IoU, larger is better. For CD, smaller is better.

Table 5.3: Effect of loss terms on pose and reconstruction estimation.

	Azimuth angle error	Reconstruction error (CD)
w/o $\mathcal{L}_{\text{sil}}$	18.51°	0.136
w/o $\mathcal{L}_{\text{fea-const}}$	15.02°	0.124
w/o $\mathcal{L}_{\text{reg}}$	13.01°	0.131
Full model	12.20°	0.116

5.2.2 Ablation Study

Effect of Unsupervised Training. By modeling the completed shape and estimating image formation parameters, our method can leverage in-the-wild images without annotations of its ground truth shape via unsupervised losses. Here we demonstrate the benefits of adding real images into training to improve our model fitting ability on real images. Fig. 5.9 shows visual reconstructions on images from Pix3D and Pascal 3D+ datasets of our model at different stage of training: a model trained with synthetic data only and a model trained with additional real images.

Effect of Loss Terms. We compare our full model with its partial variants, without silhouette consistency loss, local feature consistency loss, or albedo regularization loss. We conduct experiments on Pascal 3D+ database (car category) and evaluate the pose estimation and reconstruction. Table 5.3 shows quantitative comparison of these four models. As the silhouette provides strong constraints on global shape and pose, without silhouette loss, the performance on both metrics are severely impaired. The regularization helps to disentangle shading from albedo, which leads to better surface normal, thus better shape and pose fitting. The local feature consistency loss helps to fine-tune the model fitting, which improves the final pose and shape estimation. These results demonstrate that all the loss components presented in this work contribute to the final performance.

Table 5.4: Segmentation and shape representation comparisons (IoU/CD) on ShapeNet part [181]. IoU is utilized to measure for segmentation against ground-truth parts. CD is used for shape representation evaluation. Chair* is training on chair+table joint set.

Shape (#parts)	airplane(3)	chair(3)	chair*(4)	table(2)
BAE-Net	80.4/0.19	86.6/0.27	83.7/–	87.0/0.30
Proposed	83.0/0.16	87.4/0.23	84.1/0.28	88.2/0.25

5.2.3 Unsupervised Segmentation

As modeling shape, albedo and co-segmentation are closely-related tasks [188], jointly modeling them allows us to exploit their correlation. Following the same training and testing setting with [33], we evaluate our model's co-segmentation and shape representation power on the category of airplane, chair and table. As in Table 5.4, our model achieves a higher segmentation accuracy, comparing with BAE-NET [33]. Further, we compare the power of two methods in representing 3D shapes. By feeding a ground-truth voxel shape from the testing set to the voxel encoder and shape decoder, we can estimate the shape parameter whose decoded shape matches the ground-truth CAD model. The lower CD, as well as higher IoU, in Table 5.4 show that the novel design of our shape and albedo decoders improves both the segmentation and reconstruction.

We show additional upsupervised segmentation results of our 5 categories on ShapeNet Part dataset in Fig. 5.10. We assign a color for the output of each branch of our shape decoder and reasonable parts are obtained. Since our segmentation is unsupervised and the model for each category is trained separately, our results are not guaranteed to produce the same part counts for all categories. Fig. 5.11 shows the estimations of albedo colors of 4 branches. The four albedo branches do represent the dominant albedo colors of the objects.

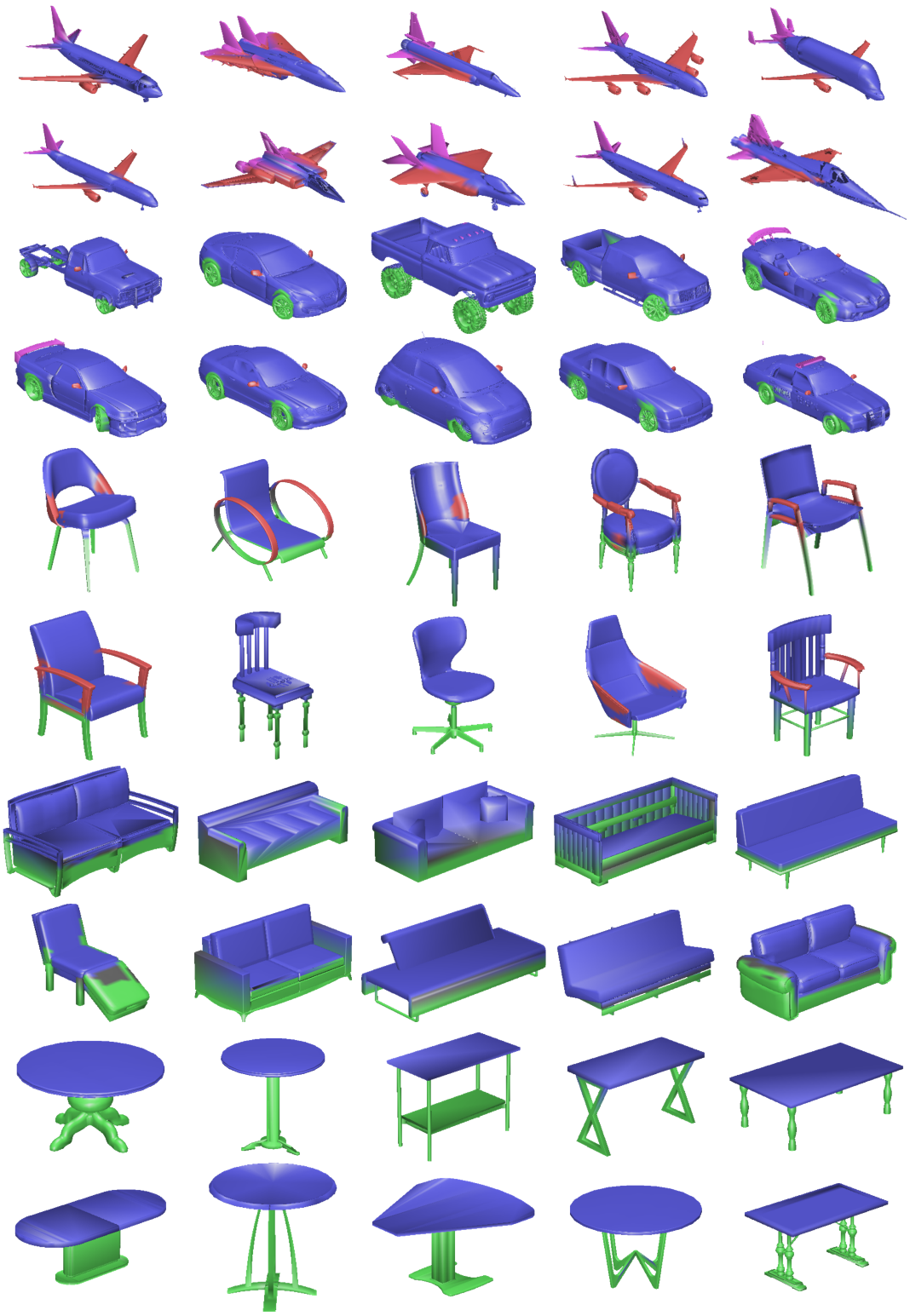


Figure 5.10: Unsupervised segmentation results on ShapeNet Part dataset. We render the original meshes with different colors representing different parts.

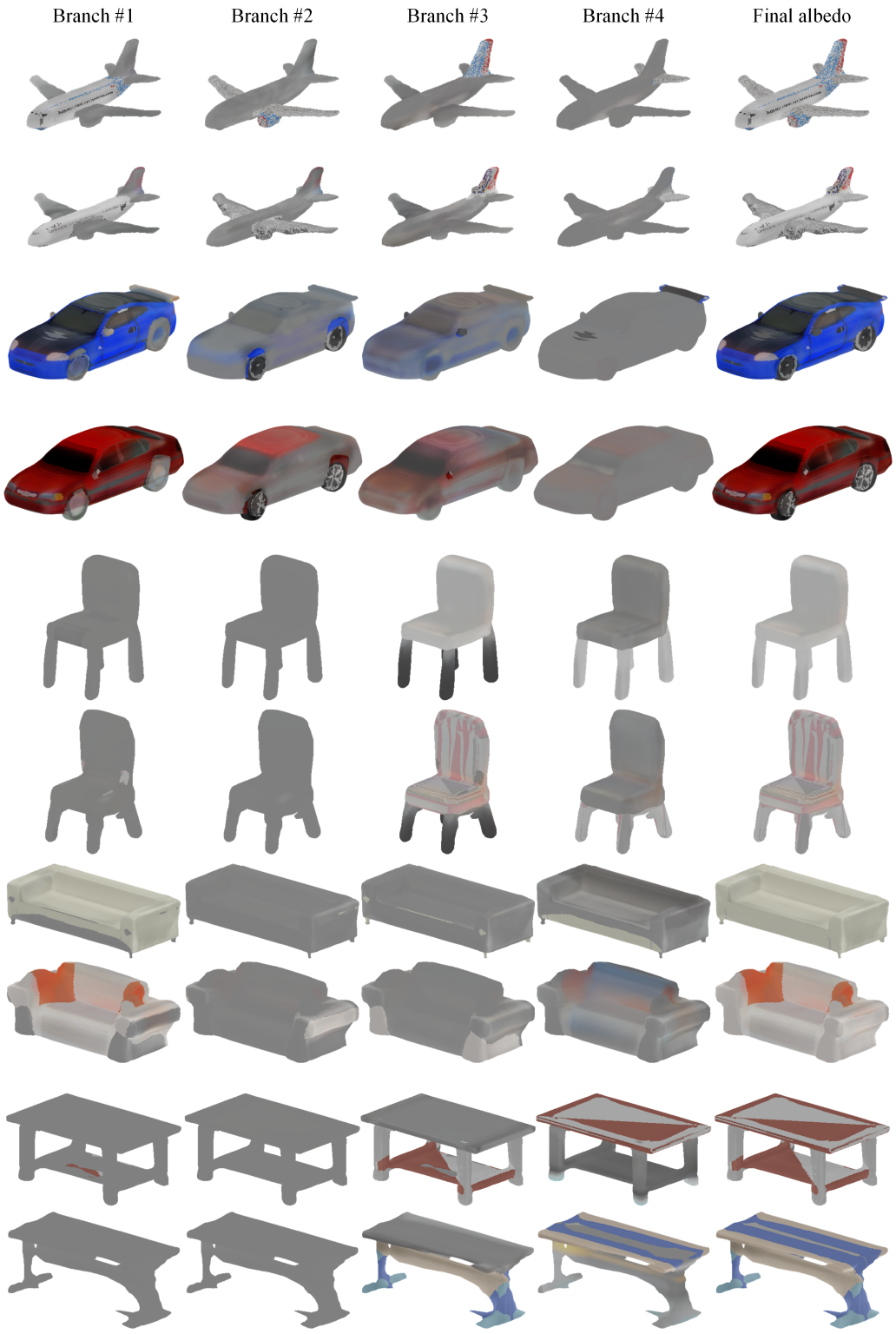


Figure 5.11: Visualization of albedo branch outputs for our 5 categories. We render the albedo with reconstructed mesh.

Table 5.5: Quantitative comparison of single-view 3D reconstruction on synthetic images of ShapeNet.

Category	Chamfer Distance						IoU					
	3D-R2N2	PSG	Pix2Mesh	AtlasNet	IM-SVR	**Proposed**	3D-R2N2	PSG	Pix2Mesh	AtlasNet	IM-SVR	**Proposed**
airplane	0.227	0.137	0.187	**0.104**	0.137	0.110	0.426	–	0.515	0.392	0.554	**0.577**
car	0.213	0.169	0.180	0.141	0.123	**0.092**	0.661	–	0.501	0.220	0.745	**0.773**
chair	0.270	0.247	0.265	0.209	0.199	**0.155**	0.439	–	0.402	0.257	0.522	**0.546**
couch	0.229	0.224	0.212	**0.177**	0.181	0.178	0.626	–	0.600	0.279	0.641	**0.651**
table	0.239	0.222	0.218	0.190	0.173	**0.164**	0.420	–	0.312	0.233	0.450	**0.479**
Mean	0.278	0.188	0.216	0.175	0.187	**0.165**	0.493	–	0.473	0.300	0.546	**0.567**

5.2.4 3D Image Decomposition

We further provide several 3D image decomposition results on real-world images on Fig. 5.12. Since our network produces a full 3D shape, we can change the reconstruction or any single component to a different viewpoint.

5.2.5 Single-view 3D Reconstruction

5.2.5.1 Reconstruction on synthetic images

Monocular 3D reconstruction performance is first evaluated on synthetic images. We compare our model against multiple state-of-the-art baselines that leverage various 3D representations: 3D-R2N2 [35] (voxel), Point Set Generation (PSG) [45] (point cloud), Pixel2Mesh [147], AtlasNet [54] (mesh), and IM-SVR [34] (implicit field). For our model, we employ both supervised and unsupervised losses.

In general, our model is able to predict 3D shapes that closely resemble the ground truth shapes (Fig. 5.13.a). Our approach outperforms the other methods in most categories and achieves the best mean score (both IoU and CD (Tab. 5.5)). While using the same shape representation as us, IM-SVR [34] only learns to reconstruct the 3D shape by minimizing the latent representation different with ground-truth latent vectors. By modeling albedo, our model is beneficial from learn-

Figure 5.12: 3D image decomposition on real-world images. Our work decomposes a 2D image of generic objects into albedo, completed 3D shape and illumination.

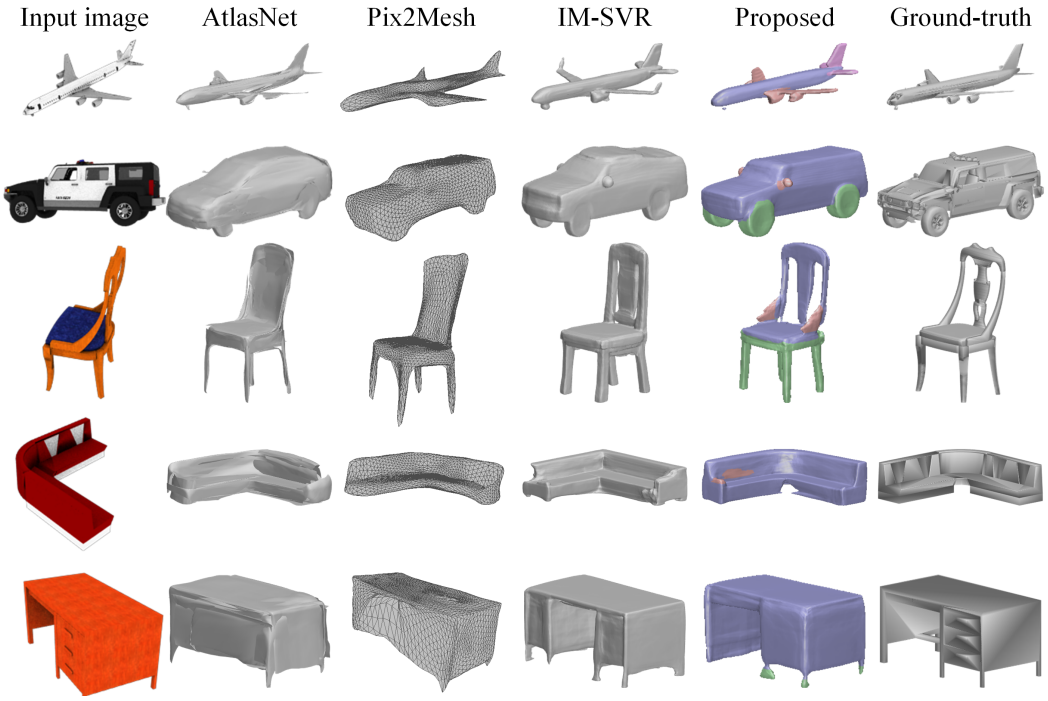


Figure 5.13: Qualitative comparison for single-view 3D reconstruction on ShapeNet, Pascal 3D+, and Pix3D datasets.

ing with both supervised and unsupervised (photometric, silhouette) losses. This results in better performance in both quantitative and qualitative comparisons.

5.2.5.2 Reconstruction on real images

We also evaluate our approach in reconstruction on two real image databases, Pascal 3D+ [177] and Pix3D [147]. Our model is finetuned on real images from Pascal 3D+ ***train*** subset *without* access to ground truth 3D shapes. Since most of reconstruction methods only can infer shapes for synthetic. Here, we compare proposed method with the state-of-the-art methods which can work for real world images, including 3D-R2N2 [35], differentiable ray consistency (DRC) [164], ShapeHD [174] and DAREC [123]. Again, our work is the first one that can fully leverage real images to learn model fitting in a unsupervised fashion. For Pascal 3D+ evaluation, we use the ***val*** subset of the 5 categories. For Pix3D, we use 3 categories (chair, couch and table) which are

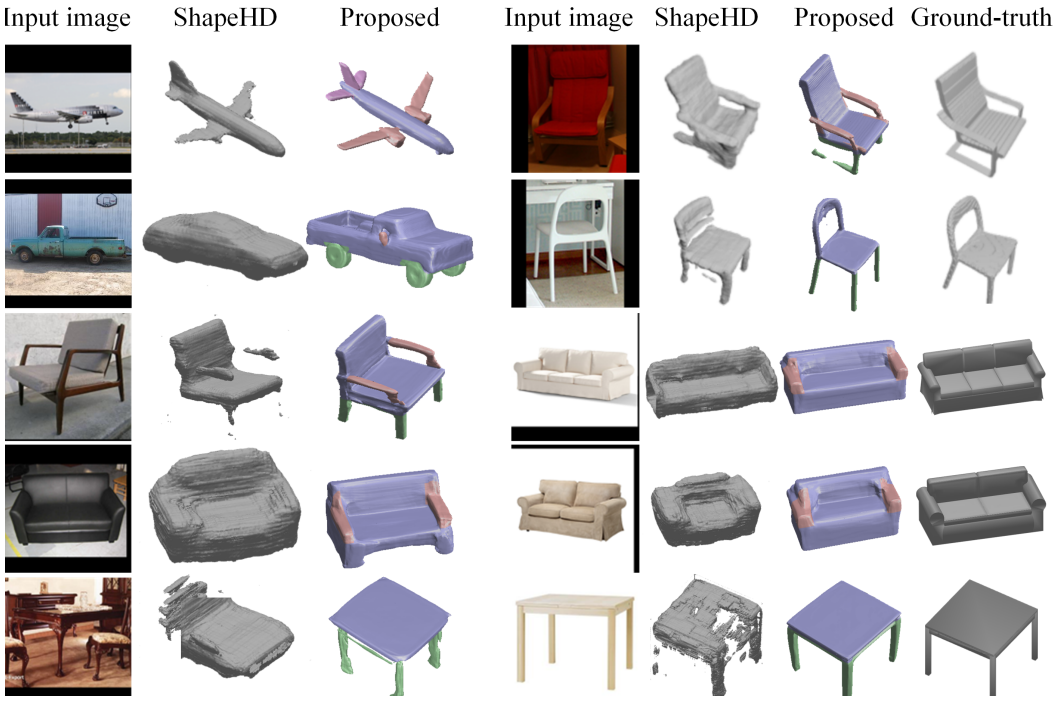


Figure 5.14: Qualitative comparison for single-view 3D reconstruction on real images from Pascal 3D+ (left) and Pix3D (right).

overlapped with our 5 real categories.

As shown in Fig. 5.14, our model infers reasonable shapes even in challenging conditions. Quantitatively, Table 5.6 suggests that the proposed method performs significantly better than other methods in Pascal 3D+ database. As Pascal 3D+ only has 10 CAD models for each object category as ground truth 3D shapes, the ground truth labels and the scores can be inaccurate, failing to reflect the shape details. We therefore conduct an experiment on more precise 3D annotation database Pix3D. As shown in Table 5.7, our model also has significantly lowest Chamfer Distance and best quality as in Fig. 5.14 comparing to baselines.

To provide more comprehensive comparisons on the 3D reconstruction quality. We provide more reconstruction results on Pascal3D+ [177] (Fig. 5.15) and Pix3D [147] dataset (Fig. 5.16. Comparisons are made with ShapeHD [174], AtlasNet [54] using pre-trained models provided by the authors.

Table 5.6: Real image 3D reconstruction on PASCAL 3D+ with CD.

Category	3D-R2N2	DRC	ShapeHD	DAREC	Proposed
plane	0.305	0.112	**0.094**	0.108	0.102
car	0.305	**0.099**	0.129	0.101	0.113
chair	0.238	0.158	0.137	0.135	**0.119**
couch	0.347	0.169	0.176	-	**0.148**
table	0.321	0.162	0.153	-	**0.127**
Mean	0.303	0.140	0.138	-	**0.122**

Table 5.7: Real image 3D reconstruction on Pix3D+ with CD.

Category	3D-R2N2	DRC	ShapeHD	DAREC	Proposed
chair	0.239	0.160	0.123	0.112	**0.091**
couch	0.307	0.178	0.137	-	**0.114**
table	0.289	0.163	0.133	-	**0.127**
Mean	0.278	0.167	0.131	-	**0.110**

5.3 Conclusions

With the objective of 3D modeling from real-world 2D images, this chapter presents a semi-supervised learning approach that jointly learns the fitting algorithm and the models. Since our approach offers completed albedo and 3D shape models, as well as intrinsic decomposition from images, we are able to effectively leverage real images in the training. As a result, we observe substantial improvement on the quality of 3D reconstruction from a single image. In essential, our proposed method is applicable to 3D modeling and reconstruction for any object category if both i) an in-the-wild 2D image collection and ii) CAD models of the object are available. We are interested in applying this method to a wide variety of object categories and building a “zoo" of 3D models.

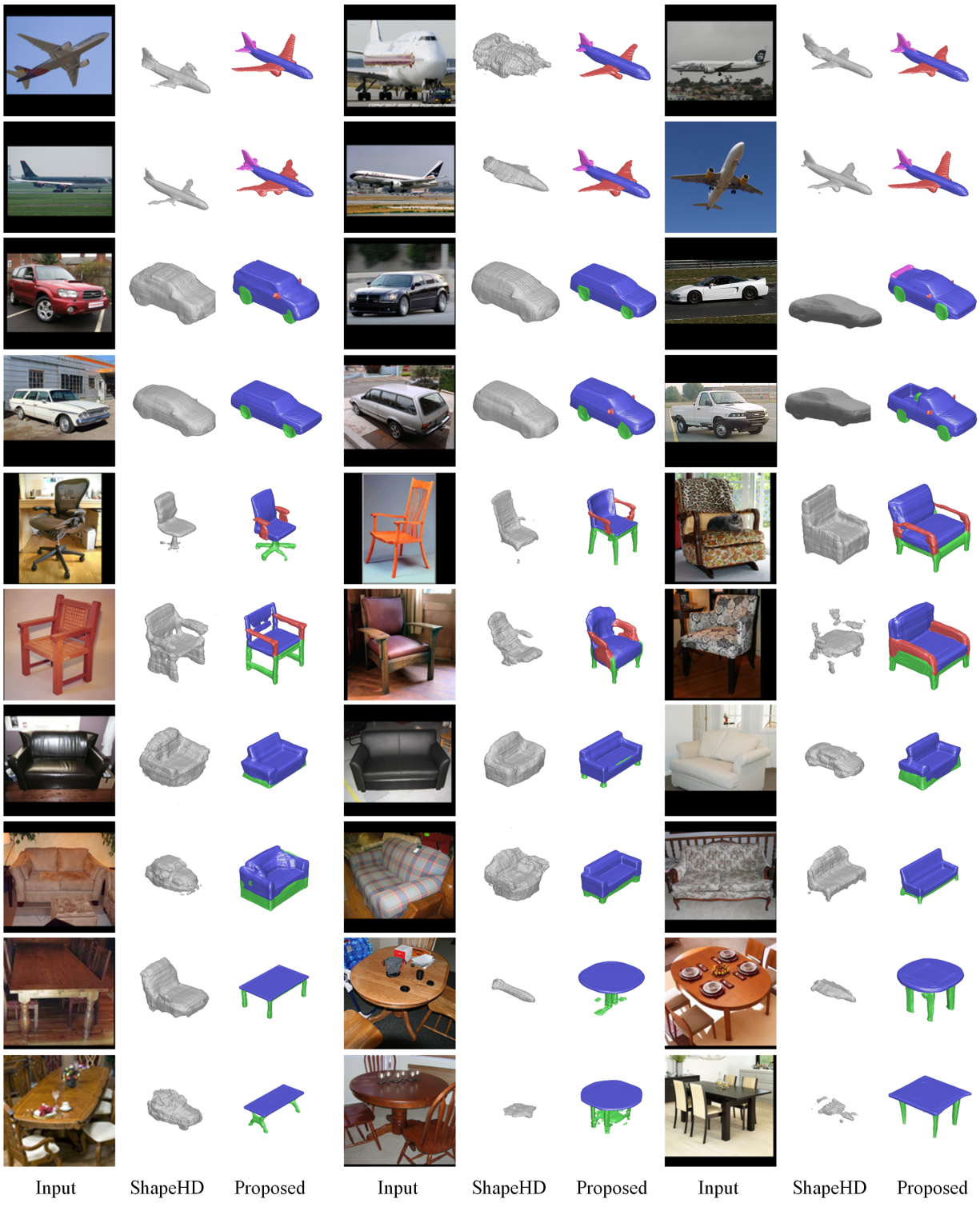


Figure 5.15: Additional 3D reconstruction results on Pascal3D+ [177] dataset.

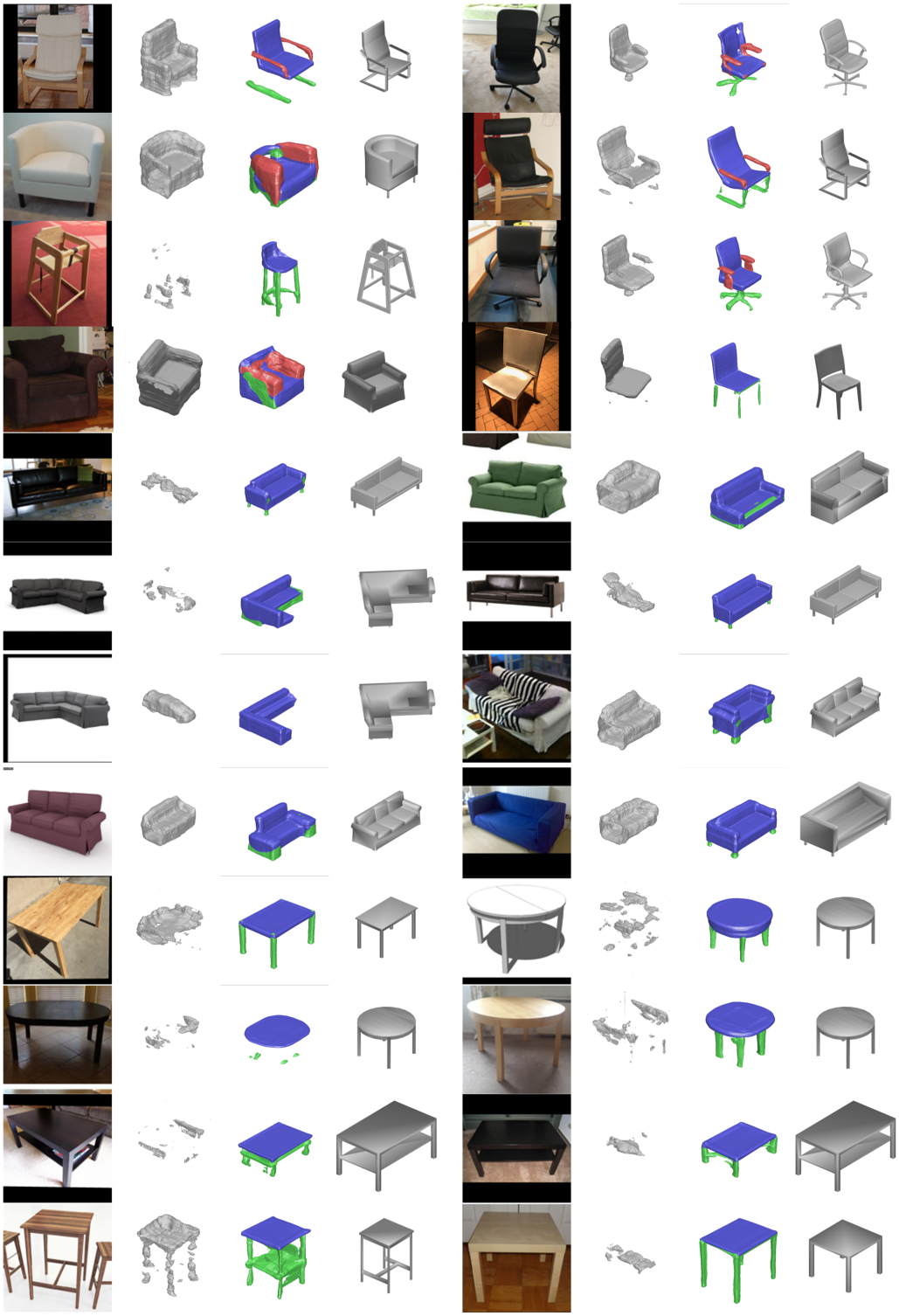


Figure 5.16: Additional 3D reconstruction results on Pix3D [147]. For each input image, we show reconstructions by ShapeHD [174], and ground truth. Our reconstructions resemble the ground truth.

Chapter 6

Conclusions and Future Work

Reconstructing faces or generic objects from a single photo graph is extremely challenging due to the ambiguity in the image formation process. Reconstruction quality is highly depend on expressiveness of the underlying used model. Given limited in annotated $3D$ data, throughout this thesis, I have presented an approach to learn and improve 3D models representation power as well as fitting ability by using large collection of 2D in-the-wild images. Even achieving the state-of-the-art performance, the current model still has limitations.

Lighting model

The Lambertian lighting model, which is used in this thesis, is known to be a poor approximation for the complex reflectance properties of facial skin or generic objects. When humans sweat, the skin clearly exhibits specular reflections, particularly on the nose and forehead. The specular reflection is even more obvious on other objects like cars. A more complex lighting assumption is necessary to accurately handle these scenarios.

I believe better modeling the lighting is critical for unsupervised/ weakly supervised approach as using a approximation of a real rendering process prevents the model from learning the true shape or albedo as these truthful elements could lead to a higher loss value under a poor approximation of lighting model. In computer graphics, extremely complex, physically-valid lighting models have been developed specifically for materials of relevance to face, for example for skin [79] and hair [100]. However, these methods have proven to be too complex and too computational-

expensive to integrate into 3DMM fitting pipelines.

Feedback Mechanism

Different from classification tasks where small changes in predicted probability can be tolerated as long as the classification (class rankings) results aren't changed; our models do regression on pose, shape/albedo parametters. High precision estimation is usually required. Currently, across all chapters, we use a single encoder to estimate parametters from the input image. With multiple down-sampling operations in the network structure, maintainig infomation of the face, inclusing precise landmark locations, small facial structure could be challenging. As a results, the estimated shape, pose could be off from the groud-truth value.

Besides, in our tasks, visualizing our current estimations, in the form of reconstructed images, gives us a luxury of comparing our estimation to the original input. Study the disperency between the reconstruction and input image could be a form of feedback signal that we can use to further refine the current estimation. Hence, one interesting idea that we could explore is to learn a second encoder that take both original input and our rendered image as inputs and try to produce parametter residuals to refine our inital predicted parametters.

APPENDIX

Representation Learning GAN for pose-invariant face recognition (DR-GAN)

While other chapters in this thesis looking at image formation/synthesis in a model-driven approach, there are other approach that can learn to manipulate images without using any 3D models. In this appendix, I would like to introduce one of our work in that direction with an application on face synthesis and face recognition.

A1 Introduction

Face recognition is one of the most widely studied topics in computer vision due to its wide application in law enforcement, biometrics, marketing, and etc. Recently, great progress has been achieved in face recognition with deep learning-based methods [149, 119, 138]. For example, surpassing human performance is reported by Schroff et al. [138] on Labeled Faces in the Wild (LFW) database. However, one of the shortcomings of the LFW database is that it does not offer a high degree of pose variation — the variance that has been shown to be a major challenge in face recognition. Up to now, the key ability of Pose-Invariant Face Recognition (PIFR) desired by real-world applications is far from solved [92, 93, 25, 4, 41]. A recent study [140] observes a significant drop, over 10%, in performance of most algorithms from frontal-frontal to frontal-profile face verification, while human performance only degrades slightly. This indicates that the pose variation remains to be a significant challenge in face recognition and warrants future study.

In PIFR, the facial appearance change caused by pose variation often significantly surpasses the intrinsic appearance differences between individuals. To overcome these challenges, a wide

This chapter is adapted from following publications:
[1] Luan Tran, Xi Yin, and Xiaoming Liu, "Disentangled Representation Learning GAN for Pose-Invariant Face Recognition," in CVPR,2017.
[2] Luan Tran, Xi Yin, and Xiaoming Liu, "Representation Learning by Rotating your Faces" in TPAMI, 2019.

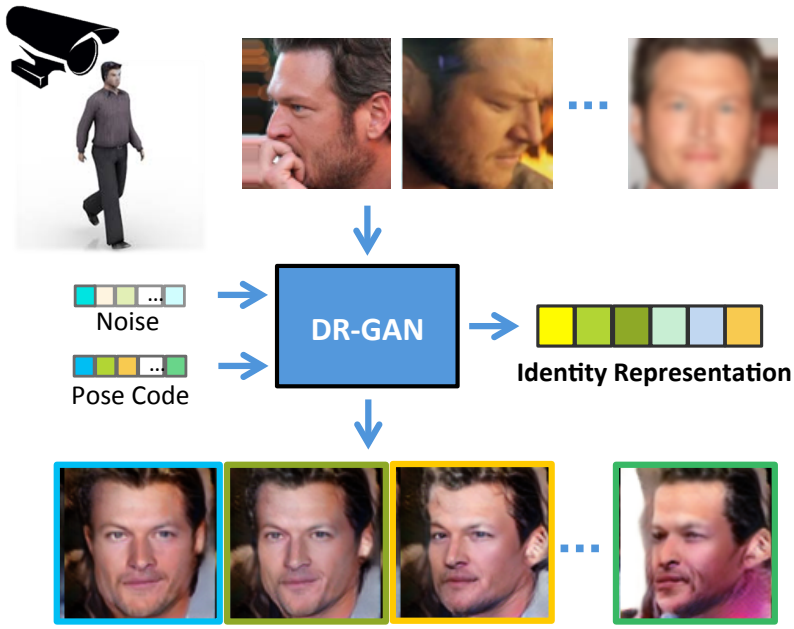


Figure A1: Given one or multiple in-the-wild face images as the input, DR-GAN can produce a unified identity representation, by virtually rotating the face to arbitrary poses. The learnt representation is both *discriminative* and *generative*, i.e., the representation is able to demonstrate superior PIFR performance, and synthesize identity-preserved faces at target poses specified by the pose code.

variety of approaches have been proposed, which can be grouped into two categories. First, some work employ *face frontalization* on the input image to synthesize a frontal-view face, where traditional face recognition algorithms are applicable [58, 194], or an identity representation can be obtained via modeling the face frontalization/rotation process [72, 197, 182]. The ability to generate a realistic identity-preserved frontal face is also beneficial for law enforcement practitioners to identify suspects. Second, other work focus on *learning discriminative representations* directly from the non-frontal faces through either one joint model [119, 138] or multiple pose-specific models [101, 40]. In contrast, we propose a novel framework to take the best of both worlds — *simultaneously learn pose-invariant identity representation* and *synthesize faces with arbitrary poses*, where face rotation is both a facilitator and a by-product for representation learning.

As shown in Fig. A1, we propose Disentangled Representation learning-Generative Adversarial Network (DR-GAN) for PIFR. Generative Adversarial Networks (GANs) [51] can generate

samples following a data distribution through a two-player game between a generator G and a discriminator D. Despite many recent promising developments [109, 39, 124, 30, 11], image synthesis remains to be the main objective of GAN. To the best of our knowledge, this is the first work that utilizes the generator in GAN for representation learning. To achieve this, we conduct G with an encoder-decoder structure (Fig. A2 (d)) to learn a disentangled representation for PIFR. The input to the encoder G_{enc} is a face image of any pose, the output of the decoder G_{dec} is a synthetic face at a target pose, and the learnt representation bridges G_{enc} and G_{dec}. While G serves as a face rotator, D is trained to not only distinguish real vs. synthetic (or fake) images, but also predict the identity and pose of a face. With the additional classifications, D strives for the rotated face to have the same identity as the input real face, which has two effects on G: 1) The rotated face looks more like the input subject in terms of identity. 2) The learnt representation is more *inclusive* or *generative* for synthesizing an identity-preserved face.

In conventional GANs, G takes a random noise vector to synthesize an image. In contrast, our G takes a face image, a pose code $\mathbf{c}$, and a random noise vector $\mathbf{z}$ as the input, with the objective of generating a face of the same identity with the target pose that can fool D. Specifically, G_{enc} learns a mapping from the input image to a feature representation. The representation is then concatenated with the pose code and the noise vector to feed to G_{dec} for face rotation. The noise models facial appearance variations other than identity or pose. Note that it is a crucial architecture design to concatenate one representation with *varying* randomly generated pose codes and noise vectors. This enables DR-GAN to learn a *disentangled* identity representation that is *exclusive* or *invariant* to pose and other variations, which is the holy grail for PIFR when achievable.

Most existing face recognition algorithms only takes one image for testing. In practice, there are many scenarios when an image collection of the same individual is available [75]. In this case, prior work fuse results either in the feature level [27] or the distance-metric level [167, 103].

Differently, our fusion is conducted within a unified framework. Given multiple images as the input, G_{enc} operates on each image, and produces an identity representation and a coefficient, which is an indicator of the quality of that input image. Using the dynamically learned coefficients, the representations of all input images are linearly combined as one representation. During testing, G_{enc} takes any number of images and generates a single identity representation, which is used by G_{dec} for face synthesis along with the pose code.

Our generator is essential to both representation learning and image synthesis. We propose two techniques to further improve G_{enc} and G_{dec} respectively. First, we have observed that our G_{enc} can always outperform D in representation learning for PIFR. Therefore, we propose to replace the identity classification part of D with the latest G_{enc} during training so that a superior D can push G_{enc} to further improve itself. Second, since our G_{dec} learns a mapping from the feature space to the image space, we propose to improve the learning of G_{dec} by regularizing the average representation of two representations from different subjects to be a valid face, assuming a convex space of face identities. These two techniques are shown to be effective in improving the generalization ability of DR-GAN.

In summary, this paper makes the following contributions.

- We propose DR-GAN via an encoder-decoder structured generator that can frontalize or rotate a face with an arbitrary pose, even the extreme profile.

- Our learnt representation is explicitly disentangled from the pose variation via the pose code in the generator and the pose estimation in the discriminator. Similar disentanglement is conducted for other variations, e.g., illumination.

- We propose a novel scheme to adaptively fuse multiple faces to a single representation based on the learnt coefficients, which empirically shows to be a good indicator of the face image

quality.

- We achieve state-of-the-art face frontalization and face recognition performance on multiple benchmark datasets, including Multi-PIE [52], CFP [140], and IJB-A [75].

A2 Prior Work

Generative Adversarial Network (GAN). Goodfellow *et al*. [51] introduce GAN to learn generative models via an adversarial process. With a minimax two-player game, the generator and discriminator can both improve themselves. GAN has been used for image synthesis [39, 127], image super resolution [187], and etc. More recent work focus on incorporating constraints to $\mathbf{z}$ or leveraging side information for better synthesis. E.g., Mirza and Osindero [109] feed class labels to both G and D to generate images conditioned on class labels. In [136] and [114], GAN is generalized to learn a discriminative classifier where D is trained to not only distinguish between real vs. fake, but also classify the images. In InfoGAN [30], G applies information regularization to the optimization by using the additional latent code. In contrast, this paper proposes a novel DR-GAN aiming for face *representation learning*, which is achieved via modeling the face rotation process. In Sec. A3.4, we will provide in-depth discussion on our difference to most relevant work in GANs.

One crucial issue with GANs is the difficulty for quantitative evaluation. Previous work either perform human study to evaluate the quality of synthetic images [39] or use the features in the discriminator for image classification [124]. In contrast, we innovatively construct the generator for representation learning, which can be quantitatively evaluated for PIFR.

Face Frontalization. Generating a frontal face from a profile face is very challenging due to self-occlusion. Prior methods in face frontalization can be classified into three categories: 3D-

based methods [194, 58, 84], statistical methods [135], and deep learning methods [197, 179, 182, 72, 191]. E.g., Hassner *et al*. [58] use a mean 3D face model to generate a frontal face for any subject. A personalized face model could be used but accurate 3D face reconstruction remains a challenge [133, 87, 160, 161]. In [135], a statistical model is used for joint frontalization and landmark localization by solving a constrained low-rank minimization problem. For deep learning methods, Kan *et al*. [72] propose SPAE to progressively rotate a non-frontal face to a frontal one via auto-encoders. Yang *et al*. [179] apply the recurrent action unit to a group of hidden units to incrementally rotate faces in fixed yaw angles.

All prior work frontalize only near frontal in-the-wild faces [58, 194] or large-pose controlled faces [182, 197]. In contrast, we can synthesize arbitrary-pose faces from a large-pose in-the-wild face. We use the *adversarial loss* to improve the quality of the synthetic images and identity classification in the discriminator to preserve identity.

Representation Learning. Designing the appropriate objectives for learning a good representation is an open question [10]. The work in [99] is among the first to use an encoder-decoder structure for representation learning, which, however, is not explicitly disentangled. DR-GAN is similar to DC-IGN [80] — a variational autoencoder-based method to disentangled representation learning. However, DC-IGN achieves disentanglement by providing batch training samples with one attribute being fixed, which may not be applicable to unstructured in-the-wild data.

Prior work also explore joint representation learning and face rotation for PIFR where [197, 182] are most relevant to our work. In [197], Multi-View Perceptron [197] is used to untangle the identity and view representations by processing them with different neurons and maximizing the data log-likelihood. Yim *et al*. [182] use a multi-task CNN to rotate a face with any pose and illumination to a target pose, and the *L2* loss-based reconstruction of the input is the second task. Both work focus on image synthesis and the identity representation is a by-product during the

network learning. In contrast, DR-GAN focuses on representation learning, of which face rotation is both a facilitator and a by-product. We differ to [197, 182] in four aspects. First, we explicitly disentangle the identity representation from pose variations by pose codes. Second, we employ the adversarial loss for high-quality synthesis, which drives better representation learning. Third, none of them applies to in-the-wild faces as we do. Finally, our ability to learn the representation from multiple unconstrained images has not been observed in prior work.

Face Image Quality Estimation. Low image quality is known to be a challenge for vision tasks [95, 32]. Image quality estimation is important for biometric recognition systems [12, 53, 157]. Numerous methods have been proposed to measure the image quality of different biometric modalities including face [1, 3, 116], iris [31, 78], fingerprint [148, 150], and gait [111, 105]. In the scenario of face recognition, an effective algorithm for face image quality estimation can help to either (i) reduce the number of poor images acquired during enrollment, or (ii) improve feature fusion during testing. Both cases can improve the face recognition performance. Abaza *et al*. [1] evaluate multiple quality factors such as contrast, brightness, sharpness, focus and illumination as a face image quality index for face recognition. However, they did not consider pose variance, which is a major challenge in face recognition. Ozay *et al*. [116] employ a Bayesian network to model the relationships between predefined quality related image features and face recognition, which is show to boost the performance significantly. The authors in [171] propose a patch-based face image quality estimation method, which takes into account of geometric alignment, pose, sharpness, and shadows.

In this work, we employ quality estimation in a unified GAN framework that considers all factors of image quality presented in the dataset, with *no* direct supervision. For each input image, DR-GAN can generate a coefficient that indicates the quality of the input image. The representations from multiple images of the same subject are fused based on the learnt coefficients to

generate one unified representation. We will show that the learnt coefficients are correlated to the image quality, i.e., a measurement of how good it can be used for face recognition.

A3 The Proposed DR-GAN Model

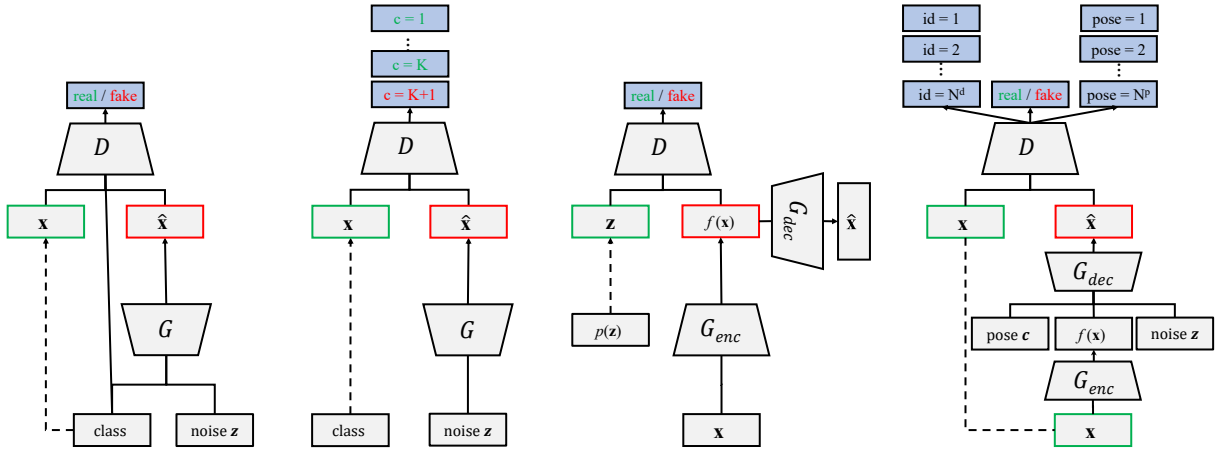
Our proposed DR-GAN has two variations: the basic model can take one image per subject for training, termed *single-image DR-GAN*, and the extended model can leverage multiple images per subject for both training and testing, termed *multi-image DR-GAN*. We start by introducing the original GAN, followed by two DR-GAN variations, and the proposed techniques to improve the generalization of our generator. Finally, we will compare our DR-GAN with previous GAN variations in detail.

A3.1 Generative Adversarial Network

Generative Adversarial Network consists of a generator G and a discriminator D that compete in a two-player minimax game. The discriminator D tries to distinguish between a real image $\mathbf{x}$ and a synthetic image $G(\mathbf{z})$. The generator G tries to synthesize realistic-looking images from a random noise vector $\mathbf{z}$ that can fool D, i.e., $G(\mathbf{z})$ being classified as a real image. Concretely, D and G play the game with the following loss function:

$$\min_G \max_D \mathcal{L}_{gan} = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \quad (1)$$

It is proved in [51] that this minimax game has a global optimum when the distribution p_g of the synthetic samples and the distribution p_d of the real samples are the same. Under mild conditions



(a) Conditional GAN (b) Semi-Supervised GAN (c) Adversarial Autoencoder (d) DR-GAN

Figure A2: Comparison of previous GAN architectures and our proposed DR-GAN.

(e.g., G and D have enough capacity), p_g converges to p_d. In the beginning of training, the samples generated from G are extremely poor and are rejected by D with high confidences. In practice, it is better for G to maximize $\log(D(G(\mathbf{z})))$ instead of minimizing $\log(1-D(G(\mathbf{z})))$ [51]. This objective results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning. As a result, G and D are trained to alternatively optimize the following objectives:

$$\max_D \mathcal{L}_{gan}^D = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1-D(G(\mathbf{z})))], \quad (2)$$

$$\max_G \mathcal{L}_{gan}^G = \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(D(G(\mathbf{z}))]. \quad (3)$$

A3.2 Single-Image DR-GAN

Our single-image DR-GAN has two distinctive novelties compared to prior GANs. First, it learns an identity representation for a face image by using an encoder-decoder structured generator, where the representation is the encoder's output and the decoder's input. Since the representation is the

input to the decoder to synthesize various faces of the same subject, i.e., virtually rotating his/her face, it is a *generative* representation.

Second, the appearance of a face is determined by not only the identity, but also the numerous distractive variations, such as pose, illumination, expression. Thus, the identity representation learned by the encoder would inevitably include the distractive side variations. E.g., the encoder would generate *different* identity representations for two faces of the same subject with 0° and 90° yaw angles. To remedy this, in addition to the class labels similar to semi-supervised GAN [136], we employ side information such as pose and illumination to explicitly disentangle these variations, which in turn helps to learn a *discriminative* representation.

A3.2.1 Problem Formulation

Given a face image $\mathbf{x}$ with label $\mathbf{y} = \{y^d, y^p\}$, where y^d represents the label for identity and y^p for pose, the objectives of our learning problem are twofold: 1) to learn a pose-invariant identity representation for PIFR, and 2) to synthesize a face image $\hat{\mathbf{x}}$ with the *same* identity y^d but at a *different* pose specified by a pose code $\mathbf{c}$. Our approach is to train a DR-GAN conditioned on the original image $\mathbf{x}$ and the pose code $\mathbf{c}$ with its architecture illustrated in Fig. A2 (d).

Different from the discriminator in conventional GAN, our D is a multi-task CNN consisting of three components: $D = [D^r, D^d, D^p]$. $D^r \in \mathbb{R}^1$ is for real/fake image classification. $D^d \in \mathbb{R}^{N^d}$ is for identity classification with N^d as the total number of subjects in the training set. $D^p \in \mathbb{R}^{N^p}$ is for pose classification with N^p as the total number of discrete poses. Given a face image $\mathbf{x}$, D aims to classify it as the real image class, and estimate its identity and pose; while given a synthetic face image from the generator $\hat{\mathbf{x}} = G(\mathbf{x}, \mathbf{c}, \mathbf{z})$, D attempts to classify $\hat{\mathbf{x}}$ as fake, using the following

objectives:

$$\mathcal{L}_{gan}^{D} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y})}[\log D^r(\mathbf{x})] + \mathbb{E}_{\substack{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y}),\\ \mathbf{z}\sim p_z(\mathbf{z}),\mathbf{c}\sim p_c(\mathbf{c})}}[\log(1 - D^r(G(\mathbf{x},\mathbf{c},\mathbf{z})))], \tag{4}$$

$$\mathcal{L}_{id}^{D} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y})}[\log D_{y^d}^{d}(\mathbf{x})], \tag{5}$$

$$\mathcal{L}_{pos}^{D} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y})}[\log D_{y^p}^{p}(\mathbf{x})], \tag{6}$$

where D_i^d and D_i^p are the ith element in D^d and D^p. For clarity, we will eliminate all subscripts for expected value notations, as all random variables are sampled from their respected distributions $(\mathbf{x},\mathbf{y} \sim p_d(\mathbf{x},\mathbf{y}), \mathbf{z} \sim p_z(\mathbf{z}), \mathbf{c} \sim p_c(\mathbf{c}))$. The final objective for training D is the weighted average of all objectives:

$$\max_D \mathcal{L}^D = \lambda_g \mathcal{L}_{gan}^{D} + \lambda_d \mathcal{L}_{id}^{D} + \lambda_p \mathcal{L}_{pos}^{D}, \tag{7}$$

where we set $\lambda_g = \lambda_d = \lambda_p = 1$.

Meanwhile, G consists of an encoder G_{enc} and a decoder G_{dec}. G_{enc} aims to learn an identity representation $f(\mathbf{x}) = G_{enc}(\mathbf{x})$ from a face image $\mathbf{x}$. G_{dec} aims to synthesize a face image $\hat{\mathbf{x}} = G_{dec}(f(\mathbf{x}),\mathbf{c},\mathbf{z})$ with identity y^d and a target pose specified by $\mathbf{c}$, and $\mathbf{z} \in \mathbb{R}^{N^z}$ is the noise modeling other variations besides identity or pose. The pose code $\mathbf{c} \in \mathbb{R}^{N^p}$ is a one-hot vector with the target pose y^t being 1. The goal of G is to fool D to classify $\hat{\mathbf{x}}$ to the identity of input $\mathbf{x}$ and the target pose with the following objectives:

$$\mathcal{L}_{gan}^{G} = \mathbb{E}[\log D^r(G(\mathbf{x},\mathbf{c},\mathbf{z}))], \tag{8}$$

$$\mathcal{L}_{id}^{G} = \mathbb{E}[\log D_{y^d}^{d}(G(\mathbf{x},\mathbf{c},\mathbf{z}))], \tag{9}$$

$$\mathcal{L}_{pos}^{G} = \mathbb{E}[\log D_{y^t}^{p}(G(\mathbf{x},\mathbf{c},\mathbf{z}))]. \tag{10}$$

Similarly, the final objective for training the discriminator G is the weighted average of each objective:

$$\max_G \mathcal{L}^G = \mu_g \mathcal{L}^G_{gan} + \mu_d \mathcal{L}^G_{id} + \mu_p \mathcal{L}^G_{pos}, \tag{11}$$

where we set $\mu_g = \mu_d = \mu_p = 1$.

G and D improves each other during the alternative training process. With D being more powerful in distinguishing real vs. fake images and classifying poses, G strives for synthesizing an identity-preserved face with the target pose to compete with D. We benefit from this process in three aspects. First, the learnt representation $f(\mathbf{x})$ will preserve more discriminative identity information. Second, the pose classification in D guides the pose of the rotated face to be more accurate. Third, with a separate pose code as input to G_{dec}, G_{enc} is trained to disentangle the pose variation from $f(\mathbf{x})$, i.e., $f(\mathbf{x})$ should encode as *much* identity information as possible, but as *little* pose information as possible. Therefore, $f(\mathbf{x})$ is not only generative for image synthesis, but also discriminative for PIFR.

A3.2.2 Network Structure

The network structure of single-image DR-GAN is adopted from CASIA-Net [180] with batch normalization (BN) for G_{enc} and D. Besides, since the stability of the GAN game suffers if sparse gradient layers (MaxPool, ReLU) are used, we replace them with strided convolution and exponential linear unit (ELU) respectively. D is trained to optimize Eqn. 7 by adding a fully connected layer with the softmax loss for real vs. fake, identity, and pose classifications respectively. G includes G_{enc} and G_{dec} that are bridged by the to-be-learned identity representation $f(\mathbf{x}) \in \mathbb{R}^{N^f}$, which is the AvgPool output in our G_{enc}. $f(\mathbf{x})$ is concatenated with a pose code $\mathbf{c}$ and a random noise $\mathbf{z}$. A series of fractionally-strided convolutions (FConv) [124] transforms the $(N^f + N^p + N^z)$-dim

concatenated vector into a synthetic image $\hat{\mathbf{x}} = G(\mathbf{x}, \mathbf{c}, \mathbf{z})$, which is the same size as $\mathbf{x}$. G is trained to maximize Eqn. 11 when a synthetic face $\hat{\mathbf{x}}$ is fed to D and the gradient is back-propagated to update G.

Previous work in face rotation use $L2$ loss [197, 182] to enforce the synthetic face to be similar to the ground truth face at the target pose. This line of work requires the training data to include face image pairs of the same identity at different poses, which is achievable for controlled datasets such as Multi-PIE, but hard to fulfill for in-the-wild datasets. On contrary, DR-GAN does not require image pairs since there is no direct supervision on the synthetic images. This enables us to utilize extensive real-world unstructured datasets for model training. To initialize the training, given a training image, we randomly sample the pose code with equal probability for each pose view. Such a random sampling is conducted at *each* epoch during the training, for the purpose of assigning *multiple* pose codes to one training image. For the noise vector, we also randomly sample each dimension independently from the uniform distribution in the range of $[-1, 1]$.

A3.3 Multi-Image DR-GAN

Our single-image DR-GAN extracts an identity representation and performs face rotation by processing one single image. Yet, we often have multiple images per subject in training and sometimes in testing. To leverage them, we propose multi-image DR-GAN that can benefit both the training and testing stages. For training, it can learn a better identity representation from multiple images that are complementary to each other. For testing, it can enable template-to-template matching, which addresses a crucial need in real-world surveillance applications.

The multi-image DR-GAN has the same D as single-image DR-GAN, but a different G as shown in Fig. A3. Given n images $\{\mathbf{x}_i\}_{i=1}^{n}$ of the same identity y^d at various poses as input, besides extracting the feature representation $f(\mathbf{x}_i)$, G_{enc} also estimates a confident coefficient ω_i

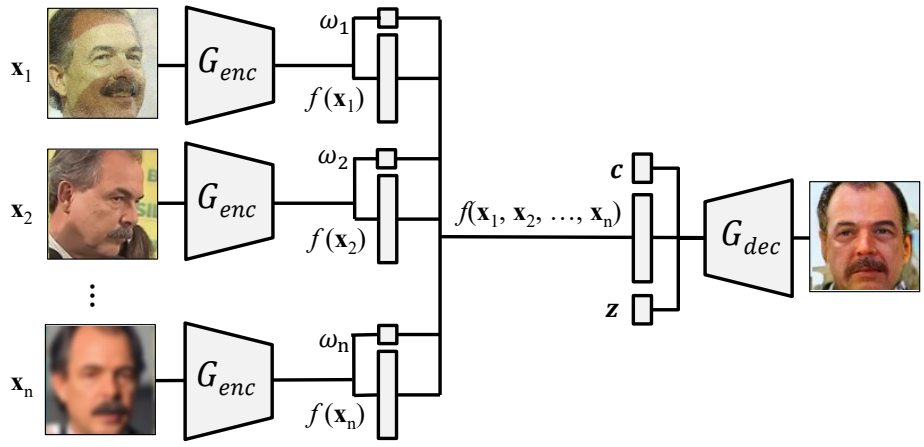


Figure A3: Generator in mlti-image DR-GAN. From an image set of a subject, we can fuse the features to a single representation via dynamically learnt coefficients and synthesize images in any pose.

for each image, which predicts the quality of the learnt representation. The fused representation of n images is the weighted average of all representations,

$$f(\mathbf{x}_1, ..., \mathbf{x}_n) = \frac{\sum_{i=1}^{n} \omega_i f(\mathbf{x}_i)}{\sum_{i=1}^{n} \omega_i}. \tag{12}$$

This fused representation is then concatenated with $\mathbf{c}$ and $\mathbf{z}$ and fed to G_{dec} to generate a new image, which is expected to have the same identity as all input images and a target pose y^t specified by the pose code. Thus, each sub-objective for learning G has $(n+1)$ terms:

$$\mathcal{L}_{gan}^{G} = \sum_{i=1}^{n} \Big[\mathbb{E}[\log(D^r(G(\mathbf{x}_i, \mathbf{c}, \mathbf{z})))]\Big] + \mathbb{E}[\log(D^r(G(\mathbf{x_1}, ..., \mathbf{x}_n, \mathbf{c}, \mathbf{z})))]. \tag{13}$$

The similar extension applied for $\mathcal{L}_{id}^{G}$ and $\mathcal{L}_{pos}^{G}$. The coefficient ω_i in Eqn. 12 is learned so that an image with a higher quality contributes more to the fused representation. The quality is an indicator of the PIFR performance of the image, rather than the low-level image quality. Face quality prediction is a classic topic where many prior work attempt to estimate the former from the latter [116, 171]. Our coefficient learning is essentially the quality prediction, from novel

perspectives in contrast to prior work. That is, without explicit supervision, it is driven by D through the decoded image $G_{dec}(f(\mathbf{x}_1, ..., \mathbf{x}_n), \mathbf{c}, \mathbf{z})$, and learned in the context of, as a byproduct of, representation learning. Note that, jointly training multiple images per subject results in *one*, but not multiple, generator, i.e., all G_{enc} in Fig. A3 share the same parameters. This makes it flexible to take an *arbitrary number* of images during testing for representation learning and face rotation.

For the network structure, multi-image DR-GAN only makes minor modification from the single-image counterpart. Specifically, at the end of G_{enc}, we add one more convolutional filter to the layer before AvgPool to estimate the coefficient ω. We apply *Sigmoid* activation to constrain ω in the range of $[0, 1]$. During training, despite unnecessary, we keep the number of input images per subject n the same for the sake of convenience in image sampling and network training. To mimic the variation in the number of input images, we use a simple but effective trick: applying Dropout on the coefficients ω: each ω is set to 0 with a probability of 0.5. Hence, during training, the network takes any number of inputs varying from 1 to n.

DR-GAN can be used in PIFR, image quality prediction, and face rotation. While the network in Fig. A2 (d) is used for training, our network for testing is much simplified. First, for PIFR, only G_{enc} is used to extract the representation from one or multiple images. Second, for quality prediction, only G_{enc} is used to compute ω from one image. Thirdly, both G_{enc} and G_{dec} are used for face rotation by specifying a target pose and a noise vector.

A3.4 Comparison to Prior GANs

We compare DR-GAN with most relevant GAN variants (Fig. A2).

Conditional GAN. Conditional GAN [109, 81] extends GAN by feeding the labels to both G

and D to generate images conditioned on labels, either class labels, modality information, or even partial data for inpainting. It has been used to generate MNIST digits conditioned on the class label and to learn multi-modal models. In conditional GAN, D is trained to classify a real image with mismatched conditions to a fake class. In DR-GAN, D classifies a real image to the corresponding class based on the labels.

Auxiliary Classifier GAN. Odena *et al*. [115] extends conditional GAN to add an additional classifier to D to classify real images into N^c classes. DR-GAN shares a similar loss for D but with a distinguish purpose. The auxiliary classifier in Odena*et al*. [115] is used to help improving the stability and quality of GAN training. Meanwhile, we employ two additional classifiers to guide the representation learning in the encoder-decoder structure G.

Adversarial Autoencoder (AAE). In AAE [98], G is the encoder of an autoencoder. AAE has two objectives in order to turn an autoencoder into a generative model: the autoencoder reconstructs the input image, and the latent vector generated by the encoder matches an arbitrary prior distribution by training D. DR-GAN differs to AAE in two aspects. First, the autoencoder in [98] is trained to learn a latent representation similar to an imposed prior distribution, while our encoder-decoder learns discriminative identity representations. Second, D in AAE is trained to distinguish real/fake distributions while our D is trained to classify real/fake images, the identity and pose of the images.

A4 Experiments

DR-GAN can be used for face recognition by using the learnt representation from G_{enc}, and face rotation by specifying different pose codes and noise vectors with G. We evaluate DR-GAN quantitatively for PIFR and qualitatively for face rotation. We further conduct experiments to analyze

the training strategy, disentangle representation, and image coefficients. Our experiments are conducted for both controlled and in-the-wild databases.

A4.1 Experimental Settings

Databases. Multi-PIE [52] is the largest database for evaluating face recognition under pose, illumination, and expression variations in controlled setting. For fair comparison, we follow the setting in [197]: using 337 subjects with neutral expression, 9 poses within $\pm 60^{\circ}$, and 20 illuminations. The first 200 subjects are used for training and the rest 137 subjects for testing. In the testing set, one image per subject with frontal view and neutral illumination forms the gallery set and the others are the probe set. For Multi-PIE experiments, we add an additional illumination code similar to the pose code to disentangle the illumination variation. Therefore, we have $N^d = 200$, $N^p = 9$, $N^{il} = 20$. Further, to demonstrate our ability in synthesizing large-pose faces, we train a second model with training faces up to 90° (i.e., $N^p = 13$).

For the in-the-wild setting, we train on CASIA-WebFace [180] and AFLW [76], and test on CFP [140] and IJB-A [75]. CASIA-WebFace includes $494,414$ near-frontal faces of $10,575$ subjects. We add the AFLW ($25,993$ images) to the training set to supply more pose variation. Since there is no identity information in this dataset, those images only used to compute GAN, pose related losses. CFP consists of 500 subjects each with 10 frontal and 4 profile images. The evaluation protocol includes frontal-frontal (FF) and frontal-profile (FP) face verification, each having 10 folders with 350 same-person pairs and 350 different-person pairs. As another large-pose database, IJB-A has $5,396$ images and $20,412$ video frames of 500 subjects. It defines template-to-template face recognition where each template has one or multiple images. We remove 27 overlap subjects between CASIA-Webface and IJB-A from the training. We have $N^d = 10,548$, $N^p = 13$. We set

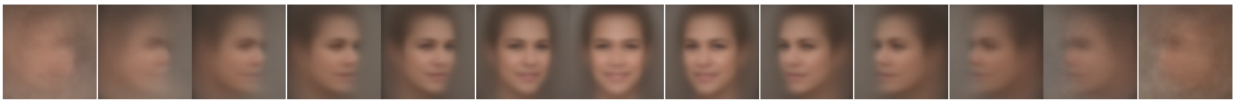


Figure A4: The mean faces of 13 pose groups in CASIA-Webface. The blurriness shows the challenges of pose estimation for large poses.

$N^f = 320$, $N^z = 50$ for both settings.

Implementation Details. Following [180], we align all face images to a canonical view of size 110×110. We randomly sample 96×96 regions from the aligned 110×110 face images for data augmentation. Image intensities are linearly scaled to the range of $[-1, 1]$. To provide pose labels y^p for CASIA-WebFace, we apply 3D face alignment [71, 70] to classify each face to one of 13 poses. The mean face image of each pose group is shown in Fig. A4. The mean faces of profile faces are less sharp than those of the near-frontal pose groups, which indicates the pose estimation error caused by the face alignment algorithm.

Our implementation is extensively modified from a publicly available implementation of DCGAN. We follow the optimization strategy in [124]. The batch size is set to be 64. All weights are initialized from a zero-centered normal distribution with a standard deviation of 0.02. Adam optimizer [74] is used with a learning rate of 0.0002 and momentum 0.5.

Evaluation. The proposed DR-GAN aims for both face representation learning and face image synthesis. The cosine distance between two representations is used for face recognition. We also evaluate the performance of face recognition w.r.t. different numbers of images in both training and testing. For image synthesis, we show qualitative results by comparing different losses and interpolation of the learnt representations. We also evaluate the various effects of different components in our method.

Table A1: DR-GAN and its partial variants performance comparison.

	Verification		Identification	
Method	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
DR-GAN $-D^r$	80.0 ± 2.2	55.5 ± 3.5	88.7 ± 0.8	95.0 ± 0.8
DR-GAN $-D^p$	78.0 ± 2.0	53.9 ± 6.8	87.5 ± 0.8	94.5 ± 0.7
DR-GAN	81.2 ± 2.7	56.2 ± 9.1	89.0 ± 1.4	95.1 ± 0.9

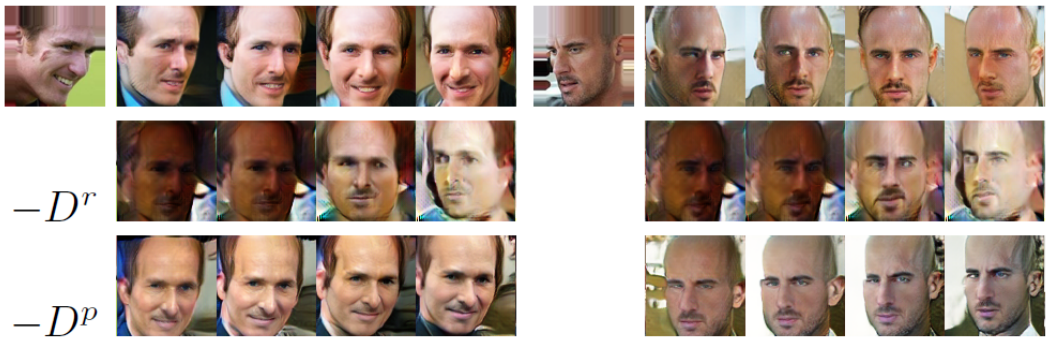


Figure A5: Generated faces of DR-GAN and its partial variants.

A4.2 Ablation study

Discriminator Components. Our discriminator is designed as a multi-task CNN with three components, namely D^g, D^d, D^p, for real/fake, identity and pose classification respectively. While D^d plays a critical role to guide the generator to preserve the input identity, we would like to study the role of the remaining components. Table A1 presents the recognition performance of single-image DR-GAN partial variants with each of D components removed. While the variant without adversarial loss has a slightly performance drop, the model without pose classification task has more severe drop. This shows the important of generating face images in different poses. Also, the role of each component is shown in generated faces (Fig. A5). When removing D^r, generated images has lower quality although they can be realized as faces and in correct poses. When removing D^p, the pose of generated images can't be controlled by the pose code and usually affected by the input face's pose. This can be caused by pose information residing in the feature

representation. This also explains the severe drop in the model's recognition performance.

Disentangled Representation. In DR-GAN, we claim that the learnt representation is disentangled from pose variations via the pose code. To validate this, following the energy-based weight visualization method proposed in [184], we perform feature visualization on the FC layer, denoted as $\mathbf{h} \in \mathbb{R}^{6\times6\times320}$, in G_{dec}. Our goal is to select two out of the 320 filters that have highest responses for identity and pose respectively. The assumption is that if the learnt representation is pose-invariant, there should be separate neurons to encode the identity features and pose features.

Recall that we concatenate $f(\mathbf{x}) \in \mathbb{R}^{320}$, $\mathbf{c} \in \mathbb{R}^{13}$ and $\mathbf{z} \in \mathbb{R}^{50}$ into one feature vector, which multiplies with a weight matrix $\mathbf{W}_{fc} \in \mathbb{R}^{(320+13+50)\times(6\times6\times320)}$ and generates the output $\mathbf{h}$ with $\mathbf{h}^i \in \mathbb{R}^{6\times6}$ being the feature output of one filter in FC. Let $\mathbf{W}_{fc} = [\mathbf{W}_{fx}; \mathbf{W}_c; \mathbf{W}_z]$ denote the weight matrix with three sub-matrices, which would multiply with $f(\mathbf{x}), \mathbf{c}, \mathbf{z}$ respectively. Taking the identity matrix as an example, we have $\mathbf{W}_{fx} = [\mathbf{W}^1_{fx}, \mathbf{W}^2_{fx}, ..., \mathbf{W}^{320}_{fx}]$ where $\mathbf{W}^i_{fx} \in \mathbb{R}^{320\times36}$. We compute an energy vector $\mathbf{s}_d \in \mathbb{R}^{320}$ with each element as: $\mathbf{s}^i_d = ||\mathbf{W}^i_{fx}||_F$. We then find the filter with the highest energy in $\mathbf{s}_d$ as $k_d = \arg\max_i \mathbf{s}^i_d$. Similarly, by partitioning $\mathbf{W}_c$, we find another filter, denoted as k_p, with the highest energy for pose.

Given the representation $f(\mathbf{x})$ of one subject, along with a pose code $\mathbf{c}$ and noise $\mathbf{z}$, we can compute the responses of two filters via $\mathbf{h}^{k_d} = (f(\mathbf{x}); \mathbf{c}; \mathbf{z})^\intercal \mathbf{W}^{k_d}_{fc}$ and $\mathbf{h}^{k_p} = (f(\mathbf{x}); \mathbf{c}; \mathbf{z})^\intercal \mathbf{W}^{k_p}_{fc}$. By varying the subjects and pose codes, we generate two arrays of responses in Fig. A6, for identity ($\mathbf{h}^{k_d}$) and pose ($\mathbf{h}^{k_p}$) respectively. For both arrays, each row represents the responses of the same subject and each column represents the same pose. The responses for identity encode the identity features, where each row shows similar patterns and each column does not share similarity. On contrary, for pose responses, each column share similar patterns while each row is not related. This visualization supports our claim that the learnt representation is pose-invariant.

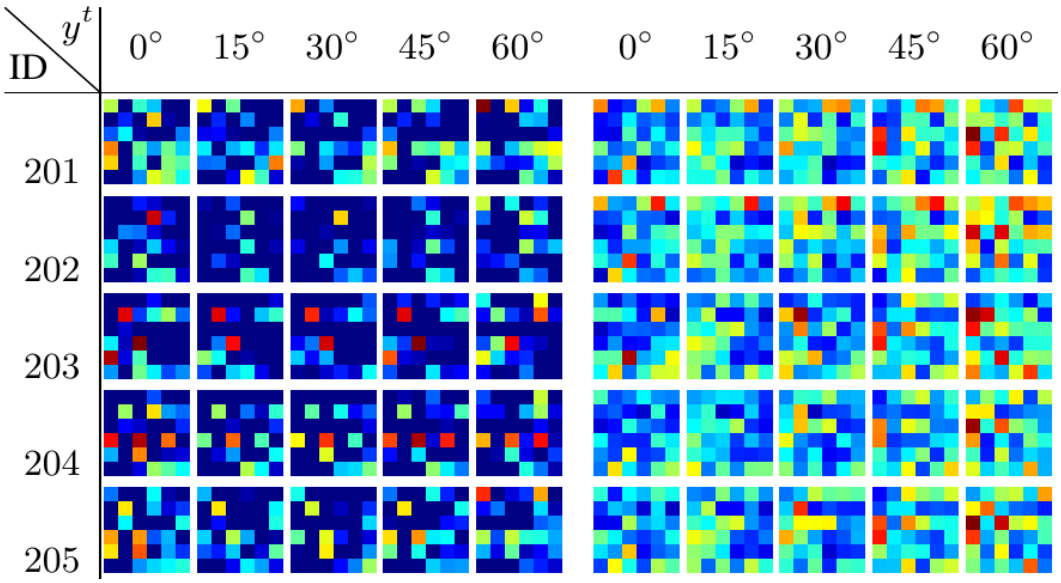


Figure A6: Responses of two filters: filter with the highest responses to identity (left), and pose (right). Responses of each row are of the same subject, and each column are of the same pose. Note the within-row similarity on the left and within-column similarity on the right.

Table A2: Comparison of single vs. multi-image DR-GAN on CFP.

Method	Frontal-Frontal	Frontal-Profile
DR-GAN: n=1	97.13 ± 0.68	90.82 ± 0.28
DR-GAN: n=4	97.86 ± 0.75	92.93 ± 1.39
DR-GAN: n=6	97.84 ± 0.79	93.41 ± 1.17

Single vs. Multiple Image DR-GAN. We evaluate the effect of the number of training images (n) per subject on the face recognition performance on CFP. Specifically, with the *same* training set, we train three models with $n = 1, 4, 6$, where $n = 1$ denotes single-image DR-GAN and $n > 1$ denotes multi-image DR-GAN. The face verification performance on CFP using $f(\mathbf{x})$ of each model are shown in Tab. A2. We observe the advantage of multi-image DR-GAN over the single-image counterpart despite they use the *same amount* of training data, which attributes to more constraints in learning G_{enc} that leads to a better representation. However, we do not keep increasing n due to the limited computation capacity. In the rest of the paper, we use multi-image DR-GAN with $n = 6$ unless specified.

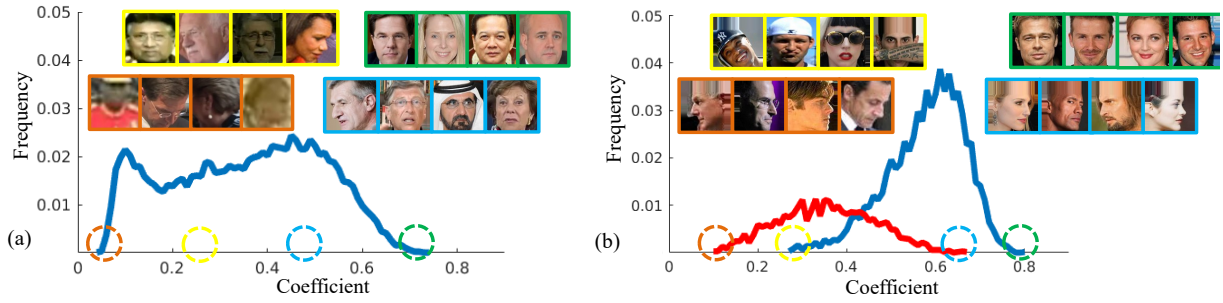


Figure A7: Coefficient distributions on IJB-A (a) and CFP (b). For IJB-A, we visualize images at four regions of the distribution. For CFP, we plot the distributions for frontal faces (blue) and profile faces (red) separately and show images at the heads and tails of each distribution.

A4.3 Confident Coefficients

In multi-image DR-GAN, we learn a confident coefficient for each input image by assuming that the learnt coefficient is indicative of the image quality, i.e., how good it can be used for face recognition. Therefore, a low-quality image should have a relatively poor representation and small coefficients so that it would contribute less to the fused representation. To validate this assumption, we compute the confident coefficients for all images in IJB-A and CFP databases and plot the distribution as shown in Fig. A7.

For IJB-A, we show four example images with low, medium-low, medium-high, and high coefficients. It is obvious that the learnt coefficients are correlated to the image quality. Images with relatively low coefficients are usually blurring, with large poses or failure cropping. While images with relatively high coefficients are of very high quality with frontal faces and less occlusion. Since CFP consists of $5,000$ frontal faces and $2,000$ profile faces, we plot their distributions separately. Despite some overlap in the middle region, the profile faces clearly have relatively low coefficients compared to the frontal faces. Within each distribution, the coefficient are related to other variations expect yaw angles. The low-quality images for each pose group are with occlusion and/or challenging lighting conditions, while the high-quality ones are with less occlusion and

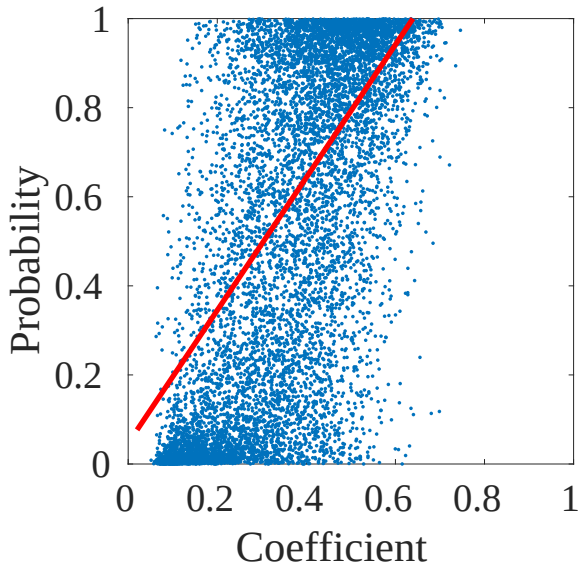


Figure A8: The correlation between the estimated coefficients and the classification probabilities.

under normal lighting.

To quantitatively evaluate the correlation between the coefficients and face recognition performance, we conduct an identity classification experiment on IJB-A. Specifically, we randomly select all frames of one video for each subject and select half of images for training and remaining for testing. The training and testing sets share the same identities. Therefore, in the testing stage, we can use the output of the softmax layer as the probability of each testing image belonging to the right identity class. This probability is an indicator of how well the input image can be recognized as the true identity. Given the estimated coefficients, we plot these two values for the testing set, as shown in Fig. A8. These two values are highly correlated to each other with a correlation of 0.69, which again supports our assumption that the learnt coefficients are indicative of the image quality.

Image selection with ω. One common application of image quality is to prevent low-quality images from contributing to face recognition. To validate whether our coefficients have such usability, we design the following experiment. For each template in IJB-A, we keep images whose

Table A3: Performance of IJB-A when removing images by threshold ω_t. "Selected" shows the percentage of retained images.

ω_t	Selected	Verification		Identification	
	(%)	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
0	100.0	**84.3** $\pm$ 1.4	72.6 $\pm$ 4.4	91.0 $\pm$ 1.5	95.6 $\pm$ 1.1
0.1	94.9	84.2 $\pm$ 1.7	72.7 $\pm$ 2.9	**91.3** $\pm$ 1.3	**95.7** $\pm$ 1.0
0.25	71.9	83.6 $\pm$ 1.2	**73.3** $\pm$ 3.0	90.7 $\pm$ 1.2	95.2 $\pm$ 1.0
0.5	24.6	80.9 $\pm$ 1.9	71.3 $\pm$ 4.7	86.5 $\pm$ 1.9	93.1 $\pm$ 1.6
1.0	5.7	77.8 $\pm$ 2.2	64.0 $\pm$ 6.2	83.4 $\pm$ 2.3	91.6 $\pm$ 1.2

coefficients ω are larger than a predefined threshold ω_t, or if all ω are smaller we keep one image with the highest ω. Tab. A3 reports the performance on IJB-A, with different ω_t. With ω_t being 0, all test images are kept and the result is the same as Tab. A6. These results show that keeping all or majority of the samples are better than removing them. This is encouraging as it reflects the effectiveness of DR-GAN in automatically diminishing the impact of low-quality images, without removing them by thresholding.

Feature fusion with ω. We also would like to show our proposed feature fusion using coefficient ω is effective for the template to template matching purpose. We compare it with multiple fusion methods in both feature level and score level. Table A4 shows comparisons of different fusion methods on our multi-image DR-GAN features. To compare two template with size n_1, n_2, for score-level, min, max, mean are respectively taking minimum, maximum and average of all $n_1 n_2$ possible pairwise distances. Mean-min is the average of $n_1 + n_2$ minimum distances from each feature from one template to the other. All of these methods have the time complexity of $\mathcal{O}(n_1 n_2)$. Softmax, proposed in [2], aggregates multiple weighted averages of the pair-wise scores, where each weight is the function of the score using an exponential function in different scales. It has the time complexity of $\mathcal{O}(m n_1 n_2)$, where m is the number of weight scale. Here, following [101], we use a total of $m = 21$ scales from 0 to 20. For feature-level fusion, max, mean are respectively

Table A4: Fusion schemes comparisons on IJB-A dataset.

		Verification		Identification	
	Method	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
Score	Min	78.3 ± 2.7	46.0 ± 6.9	86.7 ± 1.4	94.0 ± 0.6
	Max	22.8 ± 2.0	12.3 ± 2.3	30.6 ± 2.8	$52.8.0 \pm 2.7$
	Mean	72.8 ± 2.9	49.2 ± 5.3	85.7 ± 1.3	93.1 ± 0.6
	Mean-min	82.4 ± 2.2	58.5 ± 6.3	90.2 ± 1.0	$\mathbf{95.6} \pm 0.5$
	Softmax	$\mathbf{84.3} \pm 1.6$	69.2 ± 6.8	90.1 ± 1.0	95.5 ± 0.8
Feature	Max	19.0 ± 1.3	12.1 ± 1.7	45.4 ± 5.3	62.6 ± 0.9
	Mean	83.0 ± 1.5	67.0 ± 4.8	89.6 ± 1.5	95.4 ± 0.7
	ω-fusion	$\mathbf{84.3} \pm 1.4$	$\mathbf{72.6} \pm 4.4$	$\mathbf{91.0} \pm 1.5$	$\mathbf{95.6} \pm 1.1$

max-pooling and average-pooling along each feature dimension. All feature-level fusion methods, including our ω-fusion, have the time complexity of $\mathcal{O}(n_1 + n_2)$. From Tab. A4, our fusion using estimated ω achieves the best performance among all methods.

A4.4 Representation Learning

Loss Function Comparison. Our G_{dec} and D can be viewed as a loss function for $f(\mathbf{x})$. Typical loss functions used in deep learning-based face recognition can be divided into two categories: probability- and energy-based losses. Probability-based losses (i.e., softmax and its variants) usually compute a distribution of probability to all identities. Meanwhile, energy-based losses (contrastive, triplet, etc.) associate an energy to each configuration. Here, we compare DR-GAN to multiple common loss functions of face recognition. To have a fair comparison on IJB-A, for all functions, we use our G_{enc} network architecture and “mean min" fusion. DR-GAN by itself can surpass all prior loss functions (Tab. A5). Also, any advanced loss function can also be beneficial to DR-GAN: energy-based losses (center, triplet, etc.) can be employed directly on our representation $f(\mathbf{x})$ or probability-based losses (angular, additive-margin softmax, etc.) can be used to replace the D_d's softmax. Empirically, using additive-margin softmax [168] as a softmax replacement on D_d

Table A5: Loss function comparisons. All use “mean min" fusion.

	Verification		Identification	
Method	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
Softmax	75.9 ± 3.9	44.1 ± 9.9	87.8 ± 0.9	94.6 ± 0.6
Center [170]	74.9 ± 3.1	50.3 ± 7.0	87.2 ± 1.4	95.2 ± 0.9
Triplet [138]	74.9 ± 3.1	50.3 ± 7.0	87.2 ± 1.4	95.2 ± 0.9
AM-Softmax [168]	81.3 ± 3.0	52.7 ± 8.9	88.7 ± 0.7	94.3 ± 0.4
DR-GAN$_{\text{single img.}}$	81.2 ± 2.7	56.2 ± 9.1	89.0 ± 1.4	95.1 ± 0.9
DR-GAN	82.4 ± 2.3	58.5 ± 8.0	90.2 ± 1.0	$\mathbf{95.6} \pm 0.5$
DR-GAN$_{\text{AM}}$	$\mathbf{85.7} \pm 1.6$	$\mathbf{70.3} \pm 5.79$	$\mathbf{91.0} \pm 1.5$	$\mathbf{95.6} \pm 1.1$

Table A6: Performance comparison on IJB-A dataset.

	Verification		Identification	
Method	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
GOTS [75]	40.6 ± 1.4	19.8 ± 0.8	44.3 ± 2.1	59.5 ± 2.0
Wang *et al*. [167]	72.9 ± 3.5	51.0 ± 6.1	82.2 ± 2.3	93.1 ± 1.4
DCNN [27]	78.7 ± 4.3	–	85.2 ± 1.8	93.7 ± 1.0
PAM$_{frontal}$ [101]	73.3 ± 1.8	55.2 ± 3.2	77.1 ± 1.6	88.7 ± 0.9
PAMs [101]	82.6 ± 1.8	65.2 ± 3.7	84.0 ± 1.2	92.5 ± 0.8
p-CNN [184]	77.5 ± 2.5	53.9 ± 4.2	85.8 ± 1.4	93.8 ± 0.9
FF-GAN [185]	85.2 ± 1.0	66.3 ± 3.3	90.2 ± 0.6	95.4 ± 0.5
DR-GAN	85.6 ± 1.5	75.1 ± 4.2	91.3 ± 1.6	95.8 ± 1.0
DR-GAN$_{\text{AM}}$	$\mathbf{87.2} \pm 1.4$	$\mathbf{78.1} \pm 3.5$	$\mathbf{92.0} \pm 1.3$	$\mathbf{96.1} \pm 0.7$

can further improve DR-GAN performance, we name this variant as DR-GAN$_{\text{AM}}$.

Results on Benchmark Databases. We compare DR-GAN with state-of-the-art face recognizers on IJB-A, CFP and Multi-PIE.

Table A6 shows the performance of both face identification and verification on IJB-A. For our results, we report results of multi-image DR-GAN using the proposed ω-fusion. The first row shows the performance of presented DR-GAN model (using typical softmax loss). The second row presents the variant using additive margin softmax [168]. Compared to the state of the art, DR-GAN achieves superior results on both verification and identification. These in-the-wild results

Table A7: Performance (Accuracy) comparison on CFP.

Method	Frontal-Frontal	Frontal-Profile
Sengupta et al. [140]	96.40 ± 0.69	84.91 ± 1.82
Sankarana et al. [137]	96.93 ± 0.61	89.17 ± 2.35
Chen et al. [28]	$\mathbf{98.67} \pm 0.36$	91.97 ± 1.70
Human	96.24 ± 0.67	94.57 ± 1.10
DR-GAN	98.13 ± 0.81	93.64 ± 1.51
DR-GAN$_{\text{AM}}$	98.36 ± 0.75	$\mathbf{93.89} \pm 1.39$

Table A8: Identification rate (%) comparison on Multi-PIE dataset.

Method	0°	15°	30°	45°	60°	Average
Zhu et al. [196]	94.3	90.7	80.7	64.1	45.9	72.9
Zhu et al. [197]	95.7	92.8	83.7	72.9	60.1	79.3
Yim et al. [182]	**99.5**	**95.0**	88.5	79.9	61.9	83.3
Using *L*2 loss	95.1	90.8	82.7	72.7	57.9	78.3
DR-GAN	98.1	94.9	91.1	87.2	84.6	90.4
DR-GAN$_{\text{AM}}$	98.1	**95.0**	**91.3**	**88.0**	**85.8**	**90.8**

show the power of DR-GAN for PIFR.

Table A7 shows the comparison on CFP evaluated with Accuracy. Results are reported with the average with standard deviation over 10 folds. Overall, we achieve comparable performance on frontal-frontal verification while having 1.92% improvement on the frontal-profile verification.

Table A8 shows the face identification performance on Multi-PIE compared to the methods with the same setting. Our method shows a significant improvement for large-pose faces, e.g., there is more than 20% improvement margin at $\pm 60^\circ$ poses. The variation of recognition rates across different poses is much smaller than the baselines, which suggests that our learnt representation is more robust to the pose variation.

Representation vs. Synthetic Image for PIFR. Many prior work [58, 194] use frontalized faces for PIFR. To evaluate the identity preservation of synthetic images from DR-GAN, we also perform

Table A9: Representation $f(\mathbf{x})$ vs. synthetic image $\hat{\mathbf{x}}$ on IJB-A.

	Verification		Identification	
Features	@FAR=.01	@FAR=.001	@Rank-1	@Rank-5
$f(\hat{\mathbf{x}})$	78.5 ± 1.9	60.3 ± 3.7	86.9 ± 1.6	94.2 ± 1.3
$D^d(\hat{\mathbf{x}})$	77.1 ± 2.9	53.5 ± 6.2	85.7 ± 1.7	93.6 ± 1.6
$f'(\hat{\mathbf{x}})$	79.2 ± 2.9	60.8 ± 7.3	89.2 ± 1.4	95.3 ± 1.1
$f'(\hat{\mathbf{x}})\&f(\hat{\mathbf{x}})$	83.0 ± 1.8	71.7 ± 3.6	90.7 ± 1.4	$\mathbf{95.6} \pm 1.0$
$f(\mathbf{x})$	$\mathbf{84.3} \pm 1.4$	$\mathbf{72.6} \pm 4.4$	$\mathbf{91.0} \pm 1.5$	$\mathbf{95.6} \pm 1.1$

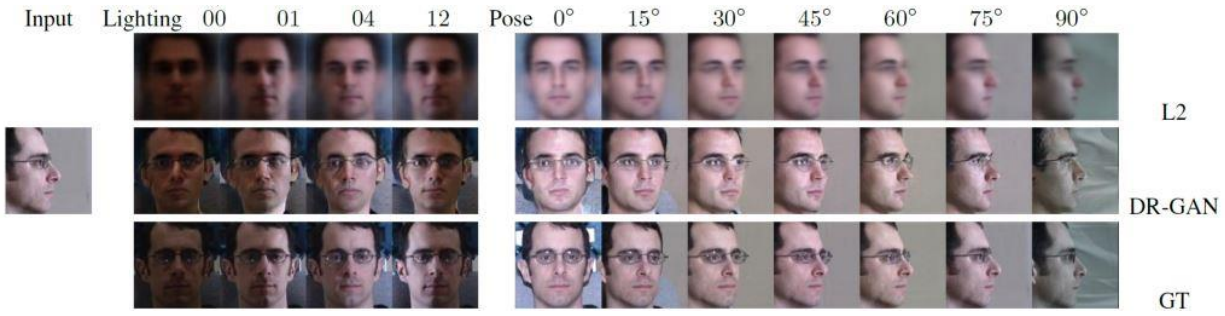


Figure A9: Face rotation comparison on Multi-PIE. Given the input (in illumination 07 and 75° pose), we show synthetic images of $L2$ loss (top), adversarial loss (middle), and ground truth (bottom). Column 2-5 show the ability of DR-GAN in simultaneous face rotation and re-lighting.

face recognition using our frontalized faces. Any face feature extractor could be applied to them, including G_{enc} or D^d. However, both are trained on real images of various poses. To specialize to synthetic frontal faces, we fine-tune G_{enc} with the synthetic images and denote as $f'(\cdot)$. As shown in Tab. A9, although the performance of synthetic images (and its score-level fusion denoted as $f'(\hat{\mathbf{x}})\&f(\hat{\mathbf{x}})$) is not as good as the learnt representation, using the fine-tuned G_{enc} on synthetic frontal still achieves comparable perfromance to the previous methods, which shows the identity preservation ability of DR-GAN.

A4.5 Face Rotation

Adversarial Loss vs. L2 loss. Prior work [196, 182, 179] on face rotation normally employ the $L2$ loss to learn a mapping between two views. To compare the $L2$ loss with our adversarial loss,

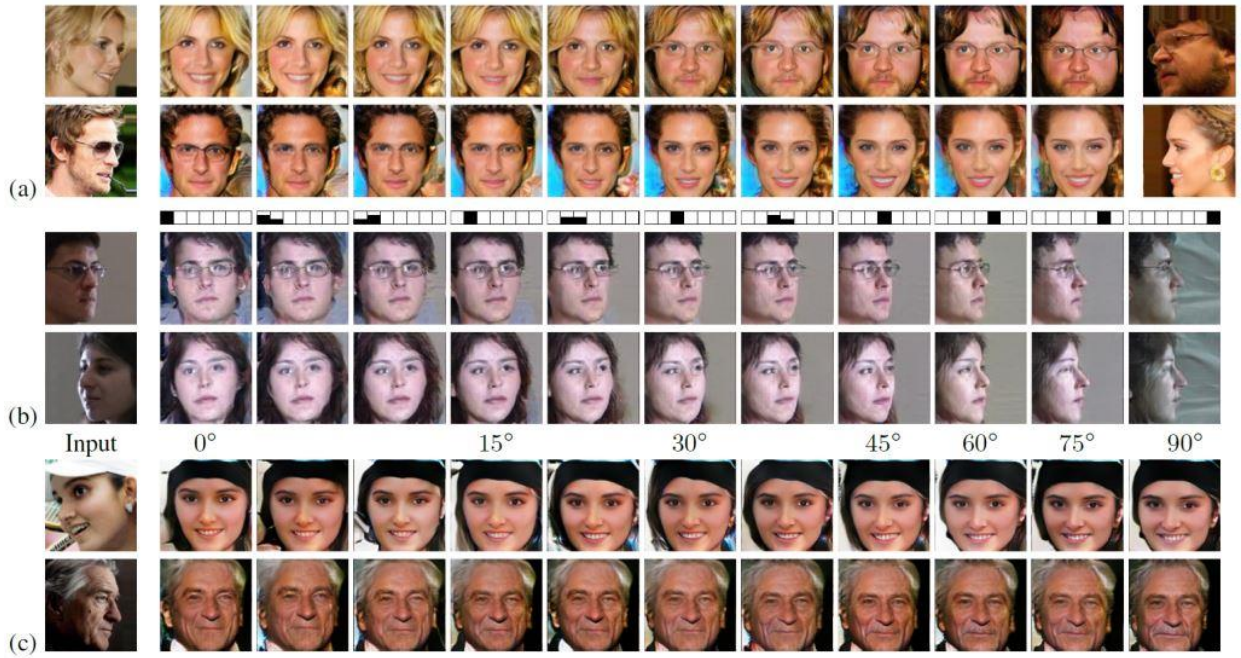


Figure A10: Interpolation of $f(\mathbf{x})$, $\mathbf{c}$, and $\mathbf{z}$. (a) Synthetic images by interpolating between the identity representations of two faces (Column 1 and 12). Note the smooth transition between different genders and facial attributes. (b) Pose angles 0°, 15°, 30°, 45°, 60°, 75°, 90° are available in the training set. DR-GAN interpolates in-between *unseen* poses via *continuous* pose codes, shown above Row 3. (c) For each image at Column 1, DR-GAN synthesizes two images at $\mathbf{z} = -\mathbf{1}$ (Column 2) and $\mathbf{z} = \mathbf{1}$ (Column 12), and in-between images by interpolating along two $\mathbf{z}$.

we train a model where G is supervised by an $L2$ loss on the ground truth face with the target view. The training process is kept the same for a fair comparison. As shown in Fig. A9, DR-GAN can generate far more realistic faces that are similar to the ground truth faces in all views. Meanwhile, images synthesized by the $L2$ loss cannot maintain high frequency components and are blurry. In fact, $L2$ loss treats each pixel equally, which leads to the loss of discriminative information. This inferior synthesis is also reflected in the lower PIFR performance in Tab. A8. In contrast, by integrating the adversarial loss, we expect to learn a more discriminative representation for better recognition, and a more generative representation for better face synthesis.

Variable Interpolations. Taking two images of different subjects $\mathbf{x}_1, \mathbf{x}_2$, we extract features $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ from G_{enc}. The interpolation between $f(\mathbf{x}_1)$ and $f(\mathbf{x}_2)$ can generate many repre-

Figure A11: Face rotation on CFP: (a) input, (b) frontalized faces, (c) real frontal faces, (d) rotated faces at 15°, 30°, 45° poses. We expect the frontalized faces to preserve the identity, rather than all facial attributes. This is very challenging for face rotation due to the in-the-wild variations and extreme profile views. The artifact in the image boundary is due to image extrapolation in pre-processing. When the inputs are frontal faces with variations in roll, expression, or occlusions, the synthetic faces can remove these variations.

sentations, which can be fed to G_{dec} to synthesize face images. In Fig. A10 (a), the top row shows a transition from a female subject to a male subject with beard and glasses. Similar to [124], these smooth semantic changes indicate that the model has learned essential identity representations for image synthesis.

Similar interpolation can be conducted for the pose codes as well. During training, we use a one-hot vector $\mathbf{c}$ to specify the *discrete* pose of the synthetic image. During testing, we could generate face images with *continuous* poses, whose pose code is the weighted average, i.e., interpolation, of two neighboring pose codes. Note that the resultant pose code is no longer a one-hot vector. As in Fig. A10 (b), this leads to smooth pose transition from one view to many views *unseen* to the training set.

We can also interpolate the noise vector $\mathbf{z}$. We synthesize frontal faces at $\mathbf{z} = -\mathbf{1}$ and $\mathbf{z} = \mathbf{1}$ (a vector of all 1s) and interpolate between two $\mathbf{z}$. Given the fixed identity representation and pose

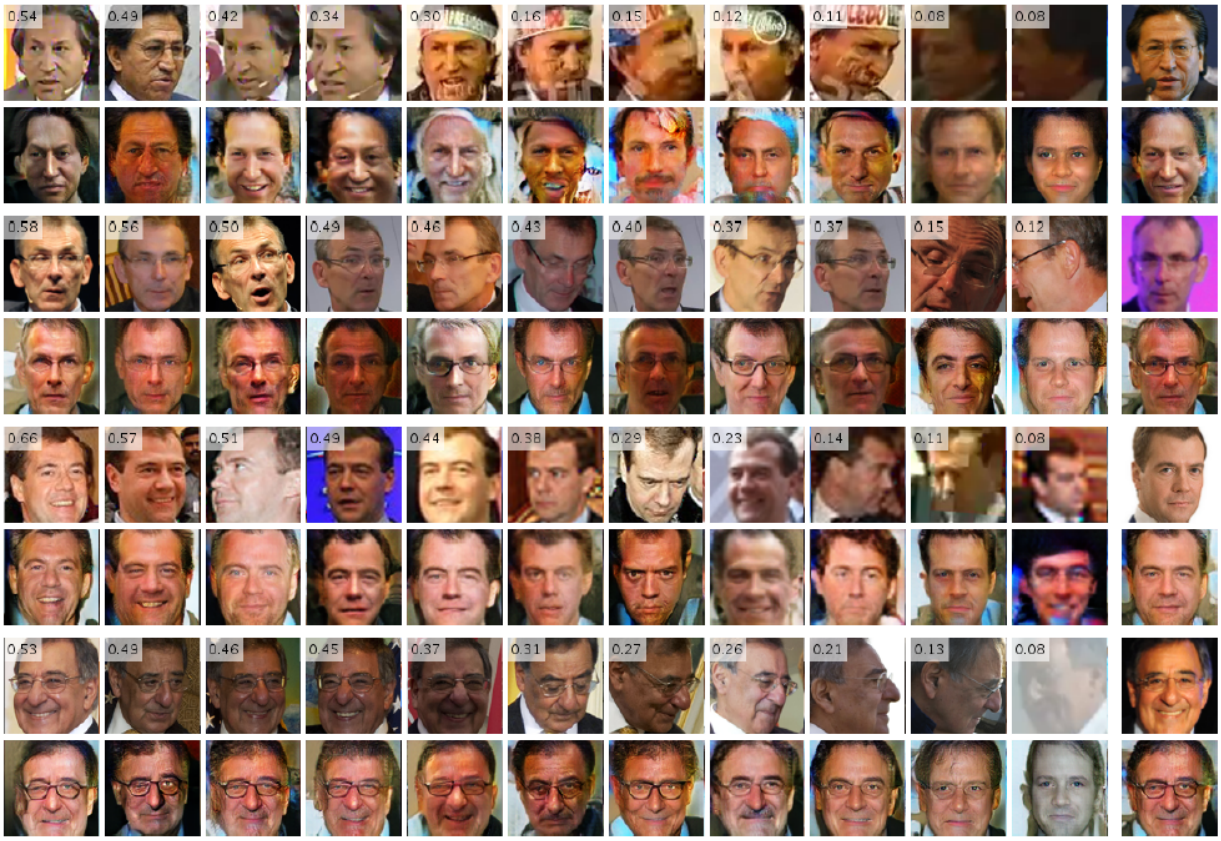


Figure A12: Face frontalization on IJB-A. For each of four subjects, we show 11 input images with estimated coefficients overlaid at the top left corner (first row) and their frontalized counter part (second row). The last column is the groundtruth frontal and synthetic frontal from the fused representation of all 11 images. Note the challenges of large poses, occlusion, and low resolution, and our *opportunistic* frontalization.

code, the synthetic images are identity-preserved frontal faces. As in Fig. A10 (c), the change of **z** leads to the change of the background, illumination condition, and facial attributes such as beard, while the identity is well preserved and faces are of the frontal view. Thus, **z** models less significant face variations.

Face Rotation on Benchmark Databases. Our generator is trained to be a face rotator. Given one or multiple face images with arbitrary poses, we can generate multiple identity-preserved faces at different views. Figure A9 shows the face rotation results on Multi-PIE. Given an input image at any pose, we can generate multi-view images of the same subject but at a different pose by specifying different pose codes or in a different lighting condition by varying illumination code.

Figure A13: Face frontalization on IJB-A for an image set (first subject) and a video sequence (second subject). For each subject, we show 11 input images (first row), their respective frontalized faces (second row) and the frontalized faces using *incrementally* fused representations from all previous inputs up to this image (third row). In the last column, we show the groundtruth frontal face.

The rotated faces are similar to the ground truth with well-preserved attributes such as eyeglasses.

One application of face rotation is face frontalization. Our DR-GAN can be used for face frontalization by specifying the frontal-view as the target pose. Figure A11 shows the face frontalization on CFP. Given an extreme profile input image, DR-GAN can generate a realistic frontal face that has similar identity characteristics as the real frontal face. To the best of our knowledge, this is the first work that is able to *frontalize a profile-view in-the-wild face image*. When the input image is already in the frontal view, the synthetic images can correct the pitch and roll angles, normalize illumination and expression, and impute occluded facial areas, as shown in the last few examples of Fig. A11.

Figure A12 shows face frontalization results on IJB-A. For each subject or template, we show 11 images and their respective frontalized faces, and the frontalized face generated from the fused representation. For each input image, the estimated coefficient ω is shown on the top-left corner

of each image, which clearly indicates the quality of the input image as well as the frontalized image. For example, coefficients for low-quality or large-pose input images are very small. These images will have very little contribution to the fused representation. Finally, the face from the fused representation has superior quality compared to all frontalized images from a single input face. This shows the effectiveness of our multi-image DR-GAN in taking advantage of multiple images of the same subject for better representation learning.

To further evaluate face frontalization results w.r.t. different numbers of input images, we vary the number of input images from 1 to 11 and visualize the frontalized images from the *incrementally* fused representations. As shown in Fig. A13, the individually frontalized faces have varying degrees of resemblance to the true subject, according to the qualities of different input images. The synthetic images from fused representations (third row) improve as the number of images increases.

A5 Conclusions

This paper presents DR-GAN to learn a disentangled representation for PIFR, by modeling the face rotation process. We are the first to construct the generator in GAN with an encoder-decoder structure for representation learning, which can be quantitatively evaluated by performing PIFR. Using the pose code for decoding and pose classification in the discriminator lead to the disentanglement of pose variation from the identity features. We also propose multi-image DR-GAN to leverage multiple images per subject in both training and testing to learn a better representation. This is the first work that is able to frontalize an extreme-pose in-the-wild face. We attribute the superior PIFR and face synthesis capabilities to the discriminative yet generative representation learned in *G*. Our representation is discriminative since the other variations are explicitly disentangled by the

pose/illumination codes, and random noise, and is generative since its decoded (synthetic) image would still be classified as the original identity.

PUBLICATIONS

Journal Papers

1. Luan Tran and Xiaoming Liu, "On Learning 3D Face Morphable Model from In-the-wild Images," in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), July 2019.

2. Luan Tran, Xi Yin, and Xiaoming Liu, "Representation Learning by Rotating Your Faces," in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), September 2018.

Conference Papers

1. Feng Liu, Luan Tran, and Xiaoming Liu, "3D Face Modeling from Diverse Raw Scan Data," Proceeding of IEEE International Conference on Computer Vision (ICCV) 2019, Seoul, South Korea, October, 2019. (Oral presentation)

2. Bangjie Yin*, Luan Tran*, Haoxiang Li, Xiaohui Shen, Xiaoming Liu, "Towards Interpretable Face Recognition," Proceeding of IEEE International Conference on Computer Vision (ICCV) 2019, Seoul, South Korea, October, 2019. (Oral presentation) (* denotes equal contribution by the authors).

3. Luan Tran, Feng Liu, Xiaoming Liu, and "Towards High-fidelity Nonlinear 3D Face Morphable Model," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019, Long Beach, California, June, 2019.

4. Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker, "Gotta Adapt 'Em All: Joint Pixel and Feature-Level Domain Adaptation for Recognition in the

Wild," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019, Long Beach, California, June, 2019.

5. Ziyuan Zhang, Luan Tran, Xi Yin, Yousef Atoum, Jian Wan, Nanxin Wang, and Xiaoming Liu, "Gait Recognition via Disentangled Representation Learning," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019, Long Beach, California, June, 2019. (Oral presentation)

6. Anurag Chowdhury and Yousef Atoum, Luan Tran, Xiaoming Liu, Arun Ross "MSU-AVIS dataset: Fusing Face and Voice Modalities for Biometric Recognition in Indoor Surveillance Videos," in Proceeding of International Conference on Pattern Recognition (ICPR), Beijing, China, August, 2018.

7. Luan Tran and Xiaoming Liu, "Nonlinear 3D Face Morphable Model," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018, Salt Lake City, Utah, June, 2018. (Spotlight presentation)

8. Luan Tran, Xi Yin, and Xiaoming Liu, "Disentangled Representation Learning GAN for Pose-Invariant Face Recognition," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Honolulu, Hawaii, July, 2017. (Oral presentation)

9. Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin, "Missing Modalities Imputation via Cascaded Residual Autoencoder," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Honolulu, Hawaii, July, 2017.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] A. Abaza, M. A. Harrison, T. Bourlai, and A. Ross. Design and evaluation of photometric image quality measures for effective face recognition. *IET Biometrics*, 2014.

[2] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, R. Nevatia, and G. Medioni. Face recognition using deep multi-pose representations. In *WACV*, 2016.

[3] M. Abdel-Mottaleb and M. H. Mahoor. Application notes-algorithms for assessing the quality of facial images. *IEEE Computational Intelligence Magazine*, 2007.

[4] R. Abiantun, U. Prabhu, and M. Savvides. Sparse feature extraction for pose-tolerant face recognition. *TPAMI*, 2014.

[5] O. Aldrian and W. A. Smith. Inverse rendering of faces with a 3D morphable model. *TPAMI*, 2013.

[6] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3D face recognition with a morphable model. In *FG*, 2008.

[7] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *CVPR*, 2007.

[8] T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling facial geometry using compositional VAEs. In *CVPR*, 2018.

[9] A. D. Bagdanov, A. Del Bimbo, and I. 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.

[10] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *TPAMI*, 2013.

[11] D. Berthelot, T. Schumm, and L. Metz. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv:1703.10717*, 2017.

[12] S. Bharadwaj, M. Vatsa, and R. Singh. Biometric quality: A review of fingerprint, iris, and face. *EURASIP JIVP*, 2014.

[13] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.

[14] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *TPAMI*, 2003.

[15] T. Bolkart and S. Wuhrer. A groupwise multilinear correspondence optimization for 3D faces. In *ICCV*, 2015.

[16] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 1989.

[17] J. Booth, E. Antonakos, S. Ploumpis, G. Trigeorgis, Y. Panagakis, and S. Zafeiriou. 3D face morphable models "In-the-wild". In *CVPR*, 2017.

[18] J. Booth, A. Roussos, E. Ververas, E. Antonakos, S. Poumpis, Y. Panagakis, and S. P. Zafeiriou. 3D reconstruction of "In-the-wild" faces in images and videos. *TPAMI*, 2018.

[19] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway. A 3D morphable model learnt from 10,000 faces. In *CVPR*, 2016.

[20] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM TOG*, 2013.

[21] G. Brazil and X. Liu. Pedestrian detection with autoregressive network phases. In *CVPR*, 2019.

[22] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks). In *ICCV*, 2017.

[23] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM TOG*, 2014.

[24] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3D facial expression database for visual computing. *TVCG*, 2014.

[25] X. Chai, S. Shan, X. Chen, and W. Gao. Locally linear regression for pose-invariant face recognition. *TIP*, 2007.

[26] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[27] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep CNN features. In *WACV*, 2016.

[28] J.-C. Chen, J. Zheng, V. M. Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *ICIP*, 2016.

[29] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *ACCV*, 2018.

[30] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[31] Y. Chen, S. C. Dass, and A. K. Jain. Localized iris image quality using 2-D wavelets. In *ICB*, 2006.

[32] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang. FSRNet: End-to-end learning face super-resolution with facial priors. In *CVPR*, 2018.

[33] Z. Chen, K. Yin, M. Fisher, S. Chaudhuri, and H. Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *ICCV*, 2019.

[34] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019.

[35] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, 2016.

[36] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman. Face synthesis from facial identity features. In *CVPR*, 2017.

[37] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *TPAMI*, 2001.

[38] A. Dai, C. Ruizhongtai Qi, and M. Nießner. Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In *CVPR*, 2017.

[39] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[40] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *TMM*, 2015.

[41] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *TIST*, 2016.

[42] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010.

[43] P. Dou, S. K. Shah, and I. A. Kakadiaris. End-to-end 3D face reconstruction with deep neural networks. In *CVPR*, 2017.

[44] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[45] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017.

[46] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3D face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018.

[47] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Perez, and C. Theobalt. Vdub: Modifying face video of actors for plausible visual alignment to a dubbed audio track. In *Computer Graphics Forum*, volume 34, pages 193–204. Wiley Online Library, 2015.

[48] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM TOG*, 2013.

[49] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. Reconstruction of personalized 3D face rigs from monocular video. *ACM TOG*, 2016.

[50] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016.

[51] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[52] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *IVC*, 2010.

[53] P. Grother and E. Tabassi. Performance of biometric quality measures. *TPAMI*, 2007.

[54] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. Atlasnet: A papier-mâché approach to learning 3D surface generation. In *CVPR*, 2018.

[55] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *ECCV*, 2008.

[56] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3D object reconstruction. In *3DV*, 2017.

[57] A. W. Harley, K. G. Derpanis, and I. Kokkinos. Segmentation-aware convolutional networks using local attention masks. In *CVPR*, 2017.

[58] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *CVPR*, 2015.

[59] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[60] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[61] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[62] R. Huang, S. Zhang, T. Li, R. He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *ICCV*, 2017.

[63] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos. Large pose 3D face reconstruction from a single image via direct volumetric cnn regression. In *ICCV*, 2017.

[64] M. Janner, J. Wu, T. D. Kulkarni, I. Yildirim, and J. Tenenbaum. Self-supervised intrinsic image decomposition. In *NeurIPS*, 2017.

[65] Z.-H. Jiang, Q. Wu, K. Chen, and J. Zhang. Disentangled representation learning for 3D face shape. In *CVPR*, 2019.

[66] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[67] A. Jourabloo and X. Liu. Pose-invariant 3D face alignment. In *ICCV*, 2015.

[68] A. Jourabloo and X. Liu. Large-pose face alignment via CNN-based dense 3D model fitting. In *CVPR*, 2016.

[69] A. Jourabloo and X. Liu. Pose-invariant face alignment via CNN-based dense 3D model fitting. *IJCV*, 2017.

[70] A. Jourabloo and X. Liu. Pose-invariant face alignment via CNN-based dense 3D model fitting. *IJCV*, 2017.

[71] A. Jourabloo, X. Liu, M. Ye, and L. Ren. Pose-invariant face alignment with a single CNN. In *ICCV*, 2017.

[72] M. Kan, S. Shan, H. Chang, and X. Chen. Stacked Progressive Auto-Encoders (SPAE) for face recognition across poses. In *CVPR*, 2014.

[73] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[74] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[75] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *CVPR*, 2015.

[76] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCVW*, 2011.

[77] P. Koppen, Z.-H. Feng, J. Kittler, M. Awais, W. Christmas, X.-J. Wu, and H.-F. Yin. Gaussian mixture 3D morphable face model. *Pattern Recognition*, 2017.

[78] E. Krichen, S. Garcia-Salicetti, and B. Dorizzi. A new probabilistic iris quality measure for comprehensive noise detection. In *BTAS*, 2007.

[79] A. Krishnaswamy and G. V. Baranoski. A biophysically-based spectral model of light interaction with human skin. In *Computer Graphics Forum*, volume 23, pages 331–340. Wiley Online Library, 2004.

[80] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015.

[81] H. Kwak and B.-T. Zhang. Ways of conditioning generative adversarial networks. In *NIPSW*, 2016.

[82] E. H. Land and J. J. McCann. Lightness and retinex theory. *Josa*, 1971.

[83] C. Li, K. Zhou, and S. Lin. Simulating makeup through physics-based manipulation of intrinsic image layers. In *CVPR*, 2015.

[84] S. Li, X. Liu, X. Chai, H. Zhang, S. Lao, and S. Shan. Morphable displacement field based image matching for face recognition across pose. In *ECCV*, 2012.

[85] F. Liu, L. Tran, and X. Liu. 3D face modeling from diverse raw scan data. In *ICCV*, 2019.

[86] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3D face reconstruction. In *ECCV*, 2016.

[87] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3D face reconstruction. In *ECCV*, 2016.

[88] F. Liu, R. Zhu, D. Zeng, Q. Zhao, and X. Liu. Disentangling features in 3D face shapes for joint face reconstruction and recognition. In *CVPR*, 2018.

[89] S. Liu, W. Chen, T. Li, and H. Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. In *ICCV*, 2019.

[90] X. Liu. Discriminative face alignment. *TPAMI*, 2009.

[91] X. Liu. Video-based face model fitting using adaptive active appearance model. *Image and Vision Computing*, 2010.

[92] X. Liu and T. Chen. Pose-robust face recognition using geometry assisted probabilistic modeling. In *CVPR*, 2005.

[93] X. Liu, J. Rittscher, and T. Chen. Optimal pose for face recognition. In *CVPR*, 2006.

[94] X. Liu, P. Tu, and F. Wheeler. Face model fitting on low resolution images. In *BMVC*, 2006.

[95] X. Liu, P. Tu, and F. Wheeler. Face model fitting on low resolution images. In *BMVC*, 2006.

[96] Y. Liu, A. Jourabloo, W. Ren, and X. Liu. Dense face alignment. In *ICCVW*, 2017.

[97] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[98] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *ICLRW*, 2015.

[99] R. Marc'Aurelio, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[100] S. R. Marschner, H. W. Jensen, M. Cammarano, S. Worley, and P. Hanrahan. Light scattering from human hair fibers. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 780–791. ACM, 2003.

[101] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *CVPR*, 2016.

[102] I. Masi, A. T. Tran, T. Hassner, J. T. Leksut, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *ECCV*, 2016.

[103] I. Masi, A. T. Tran, T. Hassner, J. T. Leksut, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *ECCV*, 2016.

[104] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440*, 2015.

[105] D. S. Matovski, M. Nixon, S. Mahmoodi, and T. Mansfield. On including quality in applied automatic gait recognition. In *ICPR*, 2012.

[106] J. McDonagh and G. Tzimiropoulos. Joint face detection and alignment with a deformable Hough transform model. In *ECCV*, 2016.

[107] A. Meka, M. Zollhöfer, C. Richardt, and C. Theobalt. Live intrinsic video. *ACM TOG*, 2016.

[108] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019.

[109] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014.

[110] U. Mohammed, S. J. Prince, and J. Kautz. Visio-lization: generating novel facial images. *TOG*, 2009.

[111] D. Muramatsu, Y. Makihara, and Y. Yagi. View transformation model incorporating quality measures for cross-view gait recognition. *IEEE transactions on cybernetics*, 2016.

[112] C. Nhan Duong, K. Luu, K. Gia Quach, and T. D. Bui. Beyond principal components: Deep Boltzmann Machines for face modeling. In *CVPR*, 2015.

[113] Y. Nirkin, I. Masi, A. T. Tran, T. Hassner, and G. M. Medioni. On face segmentation, face swapping, and face perception. In *FG*, 2018.

[114] A. Odena. Semi-supervised learning with generative adversarial networks. In *ICMLW*, 2016.

[115] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.

[116] N. Ozay, Y. Tong, F. Wheeler, and X. Liu. Improving face recognition with a quality-based probabilistic framework. In *CVPRW*, 2009.

[117] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[118] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[119] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[120] A. Patel and W. A. Smith. 3D morphable face models revisited. In *CVPR*, 2009.

[121] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *AVSS*, 2009.

[122] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 1975.

[123] P. O. Pinheiro, N. Rostamzadeh, and S. Ahn. Domain-adaptive single-view 3D reconstruction. In *ICCV*, 2019.

[124] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[125] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[126] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, 2018.

[127] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[128] E. Richardson, M. Sela, and R. Kimmel. 3D face reconstruction by learning from synthetic data. In *3DV*, 2016.

[129] E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. In *CVPR*, 2017.

[130] J. Roth, Y. Tong, and X. Liu. Unconstrained 3D face reconstruction. In *CVPR*, 2015.

[131] J. Roth, Y. Tong, and X. Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *CVPR*, 2016.

[132] J. Roth, Y. Tong, and X. Liu. Adaptive 3D face reconstruction from unconstrained photo collections. *TPAMI*, 2017.

[133] J. Roth, Y. Tong, and X. Liu. Adaptive 3D face reconstruction from unconstrained photo collections. *TPAMI*, 2017.

[134] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 2016.

[135] C. Sagonas, Y. Panagakis, S. Zafeiriou, and M. Pantic. Robust statistical face frontalization. In *ICCV*, 2015.

[136] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NIPS*, 2016.

[137] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. In *BTAS*, 2016.

[138] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[139] M. Sela, E. Richardson, and R. Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *ICCV*, 2017.

[140] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016.

[141] F. Shi, H.-T. Wu, X. Tong, and J. Chai. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM TOG*, 2014.

[142] J. Shi, Y. Dong, H. Su, and S. X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *CVPR*, 2017.

[143] Z. Shu, S. Hadap, E. Shechtman, K. Sunkavalli, S. Paris, and D. Samaras. Portrait lighting transfer using a mass transport approach. *TOG*, 2018.

[144] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

[145] F. C. Staal, A. J. Ponniah, F. Angullia, C. Ruff, M. J. Koudstaal, and D. Dunaway. Describing crouzon and pfeiffer syndrome based on principal component analysis. *Journal of Cranio-Maxillofacial Surgery*, 2015.

[146] D. Stutz and A. Geiger. Learning 3D shape completion from laser scan data with weak supervision. In *CVPR*, 2018.

[147] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3D: Dataset and methods for single-image 3D shape modeling. In *CVPR*, 2018.

[148] E. Tabassi and C. L. Wilson. A novel approach to fingerprint image quality. In *ICIP*, 2005.

[149] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[150] R. Teixeira and N. Leite. A new framework for quality assessment of high-resolution fingerprint images. *TPAMI*, 2016.

[151] A. Tewari, M. Zollhoefer, F. Bernard, P. Garrido, H. Kim, P. Perez, and C. Theobalt. High-fidelity monocular face reconstruction based on an unsupervised model-based face autoencoder. *TPAMI*, 2018.

[152] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz. In *CVPR*, 2018.

[153] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt. MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, 2017.

[154] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183:1–183:14, 2015.

[155] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of RGB videos. In *CVPR*, 2016.

[156] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. FaceVR: Real-time facial reenactment and eye gaze control in virtual reality. *arXiv:1610.03151*, 2016.

[157] Y. Tong, F. Wheeler, and X. Liu. Improving biometric identification through quality-based face and fingerprint biometric fusion. In *CVPRW*, 2010.

[158] A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *CVPR*, 2017.

[159] A. T. Tran, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. Medioni. Extreme 3D face reconstruction: Looking past occlusions. In *CVPR*, 2018.

[160] L. Tran and X. Liu. Nonlinear 3D morphable model. In *CVPR*, 2018.

[161] L. Tran and X. Liu. On learning 3D face morphable model from in-the-wild images. *TPAMI*, 2019.

[162] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR*, 2017.

[163] L. Tran, X. Yin, and X. Liu. Representation learning by rotating your faces. *TPAMI*, 2018.

[164] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017.

[165] S. Tulyakov and N. Sebe. Regressing a 3D face shape from a single image. In *ICCV*, 2015.

[166] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. In *TOG*, 2005.

[167] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *TPAMI*, 2016.

[168] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 2018.

[169] Y. Wang, L. Zhang, Z. Liu, G. Hua, Z. Wen, Z. Zhang, and D. Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *TPAMI*, 2009.

[170] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[171] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *CVPRW*, 2011.

[172] H. Wu, X. Liu, and G. Doretto. Face alignment via boosted ranking models. In *CVPR*, 2008.

[173] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3D shape reconstruction via 2.5D sketches. In *NeurIPS*, 2017.

[174] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, and J. B. Tenenbaum. Learning shape priors for single-view 3D completion and reconstruction. In *ECCV*, 2018.

[175] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *ICCV*, 2015.

[176] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3D: A large scale database for 3D object recognition. In *ECCV*, 2016.

[177] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3D object detection in the wild. In *WACV*, 2014.

[178] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann. DISN: Deep implicit surface network for high-quality single-view 3D reconstruction. In *NeurIPS*, 2019.

[179] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In *NIPS*, 2015.

[180] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014.

[181] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, et al. A scalable active framework for region annotation in 3D shape collections. *TOG*, 2016.

[182] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim. Rotating your face using multi-task deep neural network. In *CVPR*, 2015.

[183] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3D facial expression database for facial behavior research. In *FGR*, 2006.

[184] X. Yin and X. Liu. Multi-task convolutional neural network for face recognition. *TIP*, 2017.

[185] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *ICCV*, 2017.

[186] R. Yu, S. Saito, H. Li, D. Ceylan, and H. Li. Learning dense facial correspondences in unconstrained images. In *ICCV*, 2017.

[187] X. Yu and F. Porikli. Ultra-resolving face images by discriminative generative networks. In *ECCV*, 2016.

[188] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[189] E. Zell, J. Lewis, J. Noh, M. Botsch, et al. Facial retargeting with automatic range of motion alignment. *TOG*, 2017.

[190] L. Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *TPAMI*, 2006.

[191] Y. Zhang, M. Shao, E. K. Wong, and Y. Fu. Random faces guided sparse many-to-one encoder for pose-invariant face recognition. In *ICCV*, 2013.

[192] J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. Tenenbaum, and B. Freeman. Visual object networks: image generation with disentangled 3D representations. In *NeurIPS*, 2018.

[193] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3D solution. In *CVPR*, 2016.

[194] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *CVPR*, 2015.

[195] X. Zhu, X. Liu, Z. Lei, and S. Li. Face alignment in full pose range: A 3D total solution. *TPAMI*, 2017.

[196] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *ICCV*, 2013.

[197] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *NIPS*, 2014.

[198] M. Zollhöfer, J. Thies, D. Bradley, P. Garrido, T. Beeler, P. Péerez, M. Stamminger, M. Nießner, and C. Theobalt. State of the art on monocular 3D face reconstruction, tracking, and applications. *Eurographics*, 2018.