AUTOMATIC RECONSTRUCTION OF HIGH-FIDELITY BLENDSHAPE MODELS FROM IMAGES IN THE WILD

A Dissertation

by

PEIHONG GUO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jinxiang Chai
Committee Members,	John Keyser
	Andreas Klappenecker
	Eduardo Gildin
Head of Department,	Dilma Da Silva

May 2018

Major Subject: Computer Science

ABSTRACT

Facial rigs are essential to facial animation in movies, games and virtual reality applications. Among various facial models, blendshape models are widely adopted in many applications because artists can easily create complicated facial expressions through linear interpolation. However, creating high quality blendshapes for a specific subject is still a challenging task. It typically requires hours of manual work of a well trained artists to create hundreds of blendshapes in order to achieve good visual quality. Several semi-automatic and automatic systems have been proposed to generate personalized facial rigs previously; however, such systems usually requires specialized hardware (*e.g.*, laser scanner) or specific input(*e.g.*, well-lit high resolution video). This thus limits the application of such systems to a wider audience.

We present an automatic facial rigging system for generating person-specific 3D facial blendshapes from images in the wild (*e.g.*, Internet images of *Hillary Clinton*), where the face shape, pose, expression, and illuminations are all unknown. Our system initializes the 3D blendshapes with sparse facial features detected from the input images using a mutli-linear model and then refines the blendshapes via per-pixel shading cues with a new blendshape retargeting algorithm. Finally, we introduce a new algorithm for recovering detailed facial features from the input images. To handle large variations of face poses and illuminations in the input images, we also develop a set of failure detection schemes that can robustly filter out inaccurate results in each step. Our method greatly simplifies the 3D facial rigging process and generates a more faithful face shape and expression of the subject than multi-linear model fitting. We validate the robustness and accuracy of our system using images of a dozen subjects that exhibit significant variations of face shapes, poses, expressions, and illuminations.

DEDICATION

To my parents.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Chai for his continuous support of my study, for his patience, insightful advices, and immense knowledge. His guidance helped me tremedously with my research and the writing of this thesis. I could not have imagined making any meaningful progress without his constant help during my PhD study. I would also like to thank the rest of my thesis committee: Dr. Keyser, Dr. Klappenecker, and Dr. Gildin, for their insighful comments and encouragement.

In addiction, a thank you to Dr. Xin Tong from Microsoft Research Asia who provided valuable feedback on algorithm development and experiment design in my research work.

I thank my fellow labmates and friends, especially Fuhao Shi, Jianjie Zhang, Peizhao Zhang, Donghui Han, Tengteng Zhang, and Qing Li, for the stimulating discussions, for the help that saved me days of work, for the time we spent together tackling all sorts of challenges, and for the fun we had in the past few years.

Last but not the least, I would like to thank my family: my parents and my wife Gabby for supporing me throughout my academic pursuit and my life in general. I would not have made it through the tough life of a PhD student otherwise.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Jinxiang Chai [advisor] and John Keyser and Andreas Klappenecker of the Department of Computer Science and Engineering and Professor Edurado Gildin of the Department of Petroleum Engineering.

Dr. Xin Tong from Microsoft Research Asia contributed to algorithm development and experiment design in this work.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a teaching assistantship and research assistantship from Texas A&M University.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

CONTRIBUTORS AND FUNDING SOURCES v

TABLE OF CONTENTS vi

LIST OF FIGURES ix

LIST OF TABLES xii

1. INTRODUCTION 1

    1.1 Contributions 5
    1.2 Organization 6

2. LITERATURE REVIEW 7

    2.1 Face Modeling with Blendshape Models 7
        2.1.1 Rigging by Capturing and Modeling 8
        2.1.2 Rigging by Retargeting 9
        2.1.3 Rigging with Statistical Models 10
    2.2 Facial Reconstruction in the Wild 11

3. OVERVIEW 14

    3.1 Facial Reconstruction in the Wild 14
    3.2 Point Cloud Based Blendshapes Retargeting 15
    3.3 Detail Recovery 16

4. FACIAL RECONSTRUCTION IN THE WILD 18

    4.1 Parametric 3D Face Models 19
    4.2 Feature Point Detection and Robust PCA Based Failure Detection 22
    4.3 Face Reconstruction with Multi-linear Model 28
        4.3.1 Shape Reconstruction Algorithm 28
        4.3.2 Individual Reconstruction 32
        4.3.3 Joint Reconstruction 33

4.3.4 Progressive Reconstruction 33
4.4 Subset Selection 34

5. POINT CLOUD BASED BLENDSHAPE RETARGETING 41

5.1 Point Cloud Recovery and Cleanup 41
5.1.1 Shape from Shading 41
5.1.1.1 Objective Function 43
5.1.1.2 Point Cloud Recovery 45
5.1.2 Albedo Generation and Skin Detection 46
5.1.3 Point Cloud Region Selection 48
5.2 Point Clouds Based Blendshape Retargeting 49
5.2.1 Correspondence Generation 51
5.2.2 Neutral Face Optimization 52
5.2.3 Blendshape and Expression Weights Optimization 53
5.3 Iterative Blendshapes Refinement 53

6. FINE-SCALE DETAIL RECOVERY 56

6.1 Motivation 56
6.2 Fine-Scale Detail Regression 57

7. RESULTS AND EVALUATION 63

7.1 System Performance 63
7.2 Method Validation 63
7.2.1 Algorithm Convergence 63
7.2.2 Quantitative Evaluation 65
7.2.3 Contributions of Each Component 67
7.2.4 Robustness to Number of Images 68
7.3 Results for Real World Data 69
7.4 Discussions and Limitations 70
7.4.1 Error Accumulation 70
7.4.2 Shape-from-Shading vs. Photometric Stereo 70
7.4.3 Comparisons with Existing Methods 72
7.4.3.1 Compare with Garrido et al. 72
7.4.3.2 Compare with Shi et al. 73
7.4.3.3 Compare with Roth et al. and Kemelmacher-Shilizerman et al. 75

8. MORE BLENDSHAPE AND RECONSTRUCTION RESULTS 77

9. CONCLUSION AND FUTURE WORK 88

REFERENCES 92

APPENDIX A. EXTERNAL IMAGE REFERENCES 100

A.1 Andy Lau 100
A.2 Barack Obama 106
A.3 Bruce Willis 107
A.4 George Bush 107
A.5 Hillary Clinton 108
A.6 Jessica Stroup 110
A.7 Kevin Spacey 110
A.8 Rosario Dawson 112
A.9 Ziyi Zhang 113
A.10 Others 114

LIST OF FIGURES

FIGURE Page

1.1 Synthesizing facial expressions using blendshape models. 1

1.2 Our proposed system for generating blendshape models from images in the wild. 2

1.3 Examples for images in the wild. 3

1.4 Challenges of images in the wild. 4

2.1 The USC light stage. 11

2.2 Illustration of face reconstruction with deep neural network. 12

2.3 The pipeline of our automatic blendshapes generation system. 13

3.1 Comparison of direct blendshapes optimization and point cloud based blendshapes retargeting. 15

4.1 Inconsistent identity weights from direct reconstruction. 19

4.2 Examples of facial feature points detection. 24

4.3 Results by our failure detection algorithm. 27

4.4 Individual reconstruction results. 32

4.5 Joint reconstruction results. 34

4.6 Progressive reconstruction results. 35

4.7 Example of reconstruction result selection. 37

4.8 Outlier detection result of a single person. 38

4.9 Image selected for joint reconstruction by the subset selection algorithm. 39

4.10 Images excluded from joint reconstruction by the subset selection algorithm. 40

5.1 A sample SfS result. 46

5.2 Skin detection examples. 47

5.3 Initial albedo generation. 49

5.4 Point cloud region selection. 50

5.5 Initial blendshapes generated by multi-linear model. 51

5.6 Correspondence generation result. 53

5.7 The same set of blendshapes in Figure 5.5 after 1 iteration of optimization. 54

5.8 Iterative blendshapes generation. 55

6.1 Importance of fine-scale detail. 56

6.2 Comparison of displacement map based and corrective normal maps based fine detail recovery. 58

6.3 Examples of fine-scale detail recovery. 62

7.1 Blendshapes reconstruction error and point cloud fitting error on synthetic data. 64

7.2 Point cloud fitting error of the multilinear models fitting and our method on real world data. 65

7.3 Examples of point cloud fitting error. 66

7.4 Examples of point cloud fitting error compared to ground truth. 67

7.5 Blendshapes error of the multi-linear models fitting and our method on FaceWarehouse data. 68

7.6 Cumulative distribution of blendshapes error using different configurations. 68

7.7 Comparisons of the face reconstruction results using blendshapes generated from different number of input images. 69

7.8 Cumulative distribution of blendshapes error using 5, 10 and 20 input images. 70

7.9 Experiment results on real world data. 71

7.10 Comparing the results by our system and Garrido et al. 73

7.11 Comparing the results by our system and Shi et al. 74

7.12 Comparing the results by our system, Roth et al. and Kemelmacher-Shilizerman et al. 75

8.1 Automatic blendshapes generation - subject 1. 78

8.2 Automatic blendshapes generation - subject 2. 79

8.3 Automatic blendshapes generation - subject 3. 80

8.4 Automatic blendshapes generation - subject 4. 81

8.5 Automatic blendshapes generation - subject 5. 82

8.6 Automatic blendshapes generation - subject 6. 83

8.7 Automatic blendshapes generation - subject 7. 84

8.8 Automatic blendshapes generation - subject 8. 85

8.9 Facial reconstructions produced in each step of our progressive reconstruction and after each iteration of blendshapes retargeting - part 1. 86

8.10 Facial reconstructions produced in each step of our progressive reconstruction and after each iteration of blendshapes retargeting - part 2. 87

9.1 Limitations of our system. 90

LIST OF TABLES

TABLE Page

7.1 Configurations for testing contributions of each component. 69

1. INTRODUCTION

Blendshape models [5] have been widely used to animate 3D faces of virtual characters in movies, video games, and visual simulations. As an intuitive and generative facial rigging scheme, blendshape models can easily synthesize complicated facial expressions through linear interpolation(Figure 1.1). However, creating a high quality blendshape basis for a specific subject is still a challenging task. A major problem for facial rigging is to make the rigging method robust and scalable against extremely challenging inputs, such as images in the wild. This thesis addresses how to generate high quality blendshapes directly from a set of unconstrained, unstructured images.

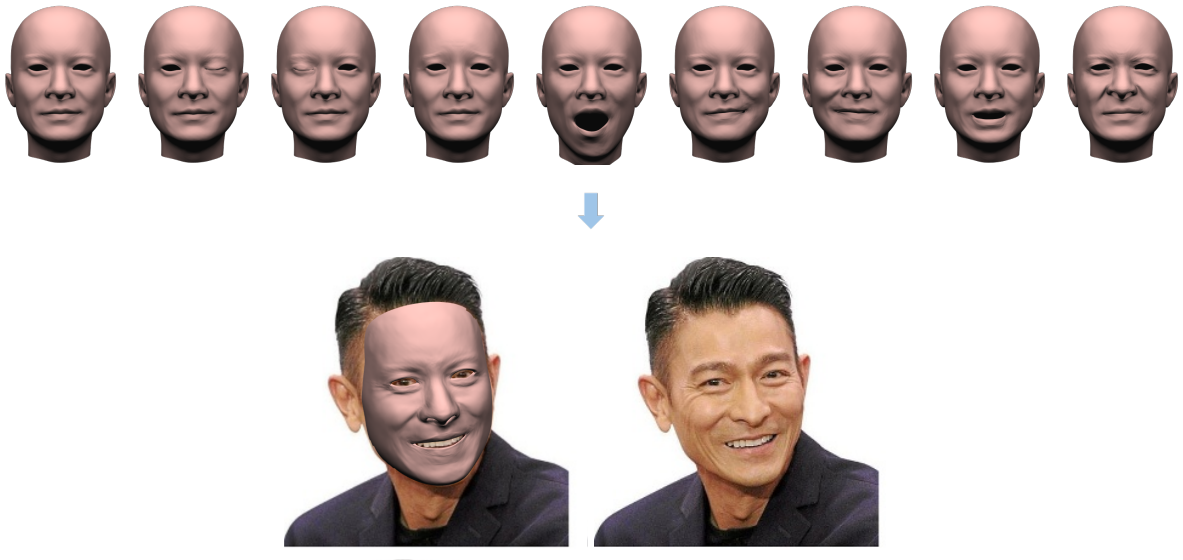


Figure 1.1: Synthesizing facial expressions using blendshape models. Top: blendshapes basis of the person. Bottom: synthesized face model(left) that matches the expression in input images(right) using the blendshapes above. Photograph reprinted from [AL2].

Many facial rigging methods have been proposed previously. A straightforward method for creating facial rigs is to directly capture the personalized blendshapes via dedicated device setups [6, 7]. Although these methods can obtain high fidelity blendshapes, they require expensive devices and considerable manual work in capturing and data processing. Example based face rigging

[8] simplifies the hardware requirement for blendshape acquisition by transferring a set of generic blendshapes to a new subject. However, it still requires a high fidelity neutral 3D shape and aligned 3D expressions of the new subject for generating high quality results. A video based facial rigging method [1] further simplifies the capturing process and generates personalized 3D facial rigs with fine-scale details from a monocular video sequence. Their system requires high resolution, well-lit video as input and represents the resulting facial rigs with a non-traditional multi-layer representation, which limits its application to more general situations. Other real-time 3D facial tracking systems [9, 10, 11, 12, 13, 14, 2, 15, 16] construct personalized blendshapes for tracking 3D facial expressions from monocular RGB/RGBD video input. Unfortunately, the resulting blendshapes are derived from a multi-linear model built upon a small group of people, which may not faithfully represent the facial geometry of the target subject.

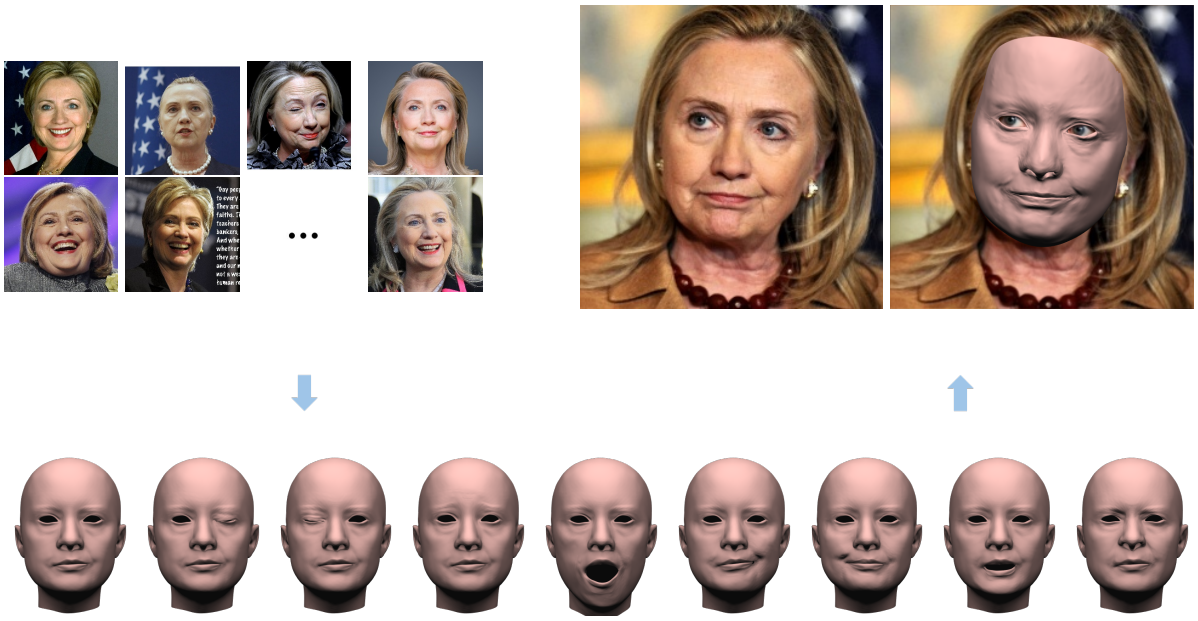


Figure 1.2: We propose the first fully automatic system that generates high fidelity blendshape models from images in the wild. From a collection of unconstrained images, our system produces blendshapes with faithful large-scale deformation. *Top Row:* Input in-the-wild images(left) and face reconstructions with the blendshapes generated with our method(right). *Bottom Row:* Blendshapes generated with our proposed system. Photographs reprinted from [HC1], [HC2], [HC5], [HC6], [HC8], [HC12], [HC11], [HC9].

We present a facial rigging system for generating high quality personalized blendshapes from a set of *images in the wild*, i.e., real world images of human faces obtained from personal albums or downloaded from Internet. Given face images of a subject with variant yet unknown poses, expressions, and illuminations, as well as a 3D blendshape template, our method automatically generates personalized 3D blendshapes from the input images(Figure 1.2). The resulting blendshape basis follows the canonical expressions in the blendshape template, while their combinations can faithfully represent the expressions and facial shapes exhibited in the input images.

Figure 1.3: Images in the wild typically have large variations in head poses, facial expressions and illumination. Photographs reprinted from [BW1], [BW2], [BW3], [BW4], [BW5], [BW6], [BW7].

Automatic facial rigging from images in the wild is a non-trivial task due to two technical challenges. First, existing facial feature detection and reconstruction algorithms are prone to fail on images with sharp lighting(Figure 1.3), significant occlusions(Figure 1.4), or extreme face poses(Figure 1.3) and result in poor blendshape results. However, automatically detecting these outlier images from the input is very difficult. Second, the faces in different images need to be aligned for modeling personalized blendshapes. Existing methods either depend on continuous video frames or face shape priors to build the correspondence between images. In our scenario, building correspondence between images is a difficult task because the input images are sparse while the underlying blendshapes are unknown.

To tackle these technical challenges, we develop a three-step solution for constructing high quality blendshapes from input images. In the first step, we leverage the multi-linear model to reconstruct an initial blendshape. For this purpose, we detect sparse facial feature points from the

input images and then reconstruct initial blendshapes with multi-linear models. A set of outlier detection schemes have been developed to automatically identify good results and remove the outliers in this step. In the second stage, we refine the blendshapes from per-pixel image values of detailed facial geometry. To this end, we first reconstruct the high resolution 3D point clouds of the face from each input image with the help of current blendshapes and then optimize the personalized blendshapes from the recovered 3D point clouds with a new blendshape retargeting algorithm. We repeat this process to iteratively update the blendshapes and point clouds until they are consistent with each other. Finally, we extract the fine-scale geometric details from the point clouds and integrate them into the generated blendshapes using a normal-based representation.

Figure 1.4: Occlusion caused by facial hair and make up makes reconstructing facial models directly from images more difficult. Non-face images such as cartoon also makes building a fully automatic pipeline very difficult. Photographs reprinted from [BW7], [BW8], [BW9], [BW10], [ZZ1], [OP1], [GB1], [GB2].

Our system is not a trivial combination of existing components. The three key technical con-

tributions of our system are outlier detection schemes, the blendshape retargeting algorithm, as well as the fine-scale facial geometry reconstruction algorithm. The outlier detection schemes make our system robust to images with extreme illuminations, face poses, and occlusions. Also, the blendshape retargeting and detail reconstruction algorithms generate personalized blendshapes consistent with the input images. Our method is robust and fully automatic, generating higher quality personalized blendshapes than the methods that are based on multi-linear models.

Our method simplifies the input for facial rigging to the greatest extent possible, making the tool universally accessible to general users. Compared to previous systems such as direct capturing and the video-based method[1], our system does not require any extra hardware and removes all constraints previous systems impose on the input. Our system therefore enables facial rigging from previously unusable sources, such as legacy images of historic figures. We validate our method on both synthetic and real world datasets and evaluate the effectiveness of our new modules in improving blendshape quality. We also test our methods on images collected from the Internet and demonstrate its robustness and accuracy for input images with various challenging combinations of head poses, facial expressions and illuminations.

1.1 Contributions

Our fully automatic facial rigging system contributes a set of algorithms to advance the state-of-the-art in facial rigging in the wild.

- We introduce several quality control modules to make the system very robust. Automatic facial rigging in the wild is extremely challenging and our system is the first one capable of constructing high quality blendshapes from a sparse set of unconstrained images.

- We develop a novel example based facial rigging algorithm using high resolution point clouds as intermediate representations for facial geometry. The intermediate representation decouples the geometry recovery and blendshapes generation step, which simplifies solution space and allows our system to identify confident image regions used for blendshapes generation.

- We design a normal-based fine-scale detail recovery algorithm to integrate fine-scale geometry into our final blendshapes. The normal-based representation is well-suited for our case because it does not suffer from the bas-relief effect typically appear in displacement based representations. Our system is not a trivial combination of existing components.

1.2 Organization

In the next chapter, we review related work on facial model, with a focus on blendshape models. Chapter 3 provides an overview for our system, and Chapter 4 to Chapter 6 describe key components, including facial reconstruction in the wild, point cloud based blendshape retargeting and fine-scale detail recovery, in our system. Chapter 7 and Chapter 8 show results and evaluation of our system. Chapter 9 concludes the discusses future work.

2. LITERATURE REVIEW

Face model has been an active research topic since the early days of computer graphics. Various methods have been proposed for face modeling over the years. These models include low-level mesh-based models[17, 18], parameterized models[19], physically-based muscle models[20, 21, 22] data-driven models[23, 24] and blendshapes[5]. While low-level mesh-based models are the most straight forward approach in representing human faces, they usually lack the flexibility of synthesizing novel face shapes with complex expressions. Parameterized models[19] provides flexible control over the facial geometry and expression through a set of tunable parameters. However, a huge set of parameters are required to produce realistic facial appearance with this model. Physically-based muscle models work well for creating realistic facial expressions, but they are often difficult for artists to control. Blendshape models[5] offer a good balance between easiness of control and realism of synthesized shapes, and have become de-facto standard in animating characters. However, it remains a major challenge to construct high quality blendshapes since hundreds of blendshapes are usually required to capture realistic facial expressions. Data driven models such as the PCA-based models[23] and the multi-linear models[24] are linear models similar to the blendshape models, but they are represented with a compressed orthogonal basis which is not intuitive for controlling facial animation. Data driven models are also limited by the samples included in their databases, leading to limited generalization ability of such models. Thus far, blendshape model is the most popular choice for facial animation in animation studio, game development and movie industry.

2.1 Face Modeling with Blendshape Models

In this section, we focus our discussion on methods and systems developed for acquiring facial blendshape models. For a comprehensive survey of facial rigging and blendshape techniques, we refer the reader to [25] and [26].

2.1.1 Rigging by Capturing and Modeling

Rigging by capturing and modeling directly acquires 3D blendshapes from various 3D data sources. Artist-crafted blendshapes [7] have been successfully used in movie making [27, 28]. However, manually creating high quality blendshapes is time consuming and requires specially trained experts. Acquiring facial rigs from 3D capturing provides a high precision alternative to manual crafting. A straightforward approach directly captures facial performances with high resolution laser scans [24, 29, 6]. Structured lighting systems [30, 31] and multiple-view stereo [18, 32, 33] recover high quality facial geometry via 3D reconstruction. These systems often involve complicated system setups and a controlled environment. [16] proposed to use mobile devices as low cost alternatives for scanning human faces, and their method is able to produce visually appealing avatars. Nonetheless, their method only works in situations where the person of interest is accessible. Recent advances in consumer depth cameras have made it possible to obtain 3D scans with a simplified setup at low cost [34, 10, 13, 35]. However, the 3D scans are usually very noisy and require a substantial amount of post-processing. By contrast, our system only uses a set of unconstrained images as input and is fully automatic, thus offers an extremely low cost option for facial rigging.

Suwajanakorn et al.[36] reconstructed 3D facial animations from video clips by combining shape from shading and optical flow techniques. However, the unstructured meshes recovered by this method cannot be easily used for generating new facial animations. Recently, Garrido et al. [1] proposed a multi-scale performance capture system that builds highly detailed, personalized 3D face rigs from a monocular video clip. Given a high resolution, well-lit video clip, their system generates a multi-layer blendshapes model, where the large-scale blendshape is modeled by a multi-linear model, and the personalized shape and expressions are represented by a corrective field and per-triangle deformation gradients that are derived from each input frame using expensive online regression. Different from this approach, our method takes images in the wild as input for high quality blendshape generation. For this purpose, our method has to handle challenging issues typically arise only from our input images in the wild, such as illumination change and significant

occlusion. Moreover, we cannot follow the method in [1] designed for dense video sequences to reconstruct a dense middle-level representation from sparse images. Instead, we model the personalized shape and expressions out of the multi-linear model space with personalized blendshapes and introduce a new optimization method for reconstructing personalized blendshapes from sparse input images.

2.1.2 Rigging by Retargeting

Rigging by retargeting generates personalized blendshapes by adapting a blendshape template to a specific subject's face. Early approaches create blendshapes by fitting a generic face model to sparse 2D feature points [37] or 3D mocap data [38, 39]. While these methods successfully create personalized blendshapes with sparse constraints, the resulting blendshapes generally lack fine-scale details compared to manually crafted blendshapes. Later methods resolve this issue by adapting blendshapes with dense correspondence. Noh et al. [40] cloned expressions of a source face model to a target face by transferring the vertex motion vectors of the source mesh to the target mesh. Orvalho et al. [41] introduced a facial rig transfer system using a combination of landmark-guided dense correspondence and attributes transfer. Xu et al. [42] decomposed detailed facial expressions at different scales and transferred them separately. Although these approaches only require one 3D neutral face of the target subject for retargeting, the generated blendshapes may not always appear as expected. As a result, at artist's input is always required to control the blendshape's quality. Li et al. [8] proposed a method for generating blendshapes from an aligned neutral 3D face and several 3D expressions of a subject using a constrained deformation transfer algorithm. Although this method significantly improves the blendshape quality, it needs a neutral 3D face and well established correspondence between 3D input poses for facial rigging. The new example based facial rigging scheme introduced in our system eliminates these constraints and enables high quality blendshapes retargeting directly from unaligned and noisy point cloud data.

Cashman et al. [43] proposed a method for constructing blendshapes of deformable 3D objects (e.g dolphin or pigeons) from images in the wild, where the 2D object contours labeled in each image are used to deform the 3D template mesh. Unfortunately, this method cannot be used for

automatic facial rigging. Different from our method that aims to recover personalized facial animations using per-pixel shading cues, their method can only derive large-scale object animations from contours and ignores the difference between different identities. Moreover, their method needs a lot of manual work for labeling contours in the input images, while our method is fully automatic.

2.1.3 Rigging with Statistical Models

Rigging with statistical models derives the blendshapes from various inputs by fitting a statistical model of face shapes and expressions, such as the PCA model[23] or the multi-linear model [24, 35]. Bouaziz et al. [10] created personalized blendshapes by transferring expressions from template blendshapes to a neutral shape obtained with a PCA identity model. Real time video-based 3D facial tracking systems [11, 14, 15] generated user-specific blendshapes using shape regression in a multi-linear model, while other offline 3D expression reconstruction approaches [2, 44] also derive personalized blendshapes by fitting the 2D face image detected in video frames with a multi-linear model. Since these models are constructed from a relatively small group of people, they are only able to generate facial rigs within the model space and thus may not faithfully reflect the true facial geometry. Our system overcomes this limitation by fitting the point clouds recovered from per-pixel visual cues in the input images with free-form deformed blendshape templates. As a result, the blendshapes generated by our system are both visually appealing and geometrically closer to the true facial shape.

Recently, Ichim and colleagues proposed to use physics-based simulation to improve realism of blendshape models under different physical interaction status[45]. Their combine muscle control, volumetric deformation energy and rigid bone structure constraints in their simulation to produce visually plausible facial movements. However, their method requires very expensive physics-based simulation that limits its application scenario. Li et al proposed to learn a statistical model using 4D scans[46], i.e. 3D scans that capture the continuous facial movements within certain time span. Their method is able to produce high quality blendshape models by utilizing both the accurate 3D geometry and space-time continuity in the 4D scans. Despite the impressive results, their method requires a much more complicated data collection setup which is not applicable to our case of

creating blendshape models from images in the wild.

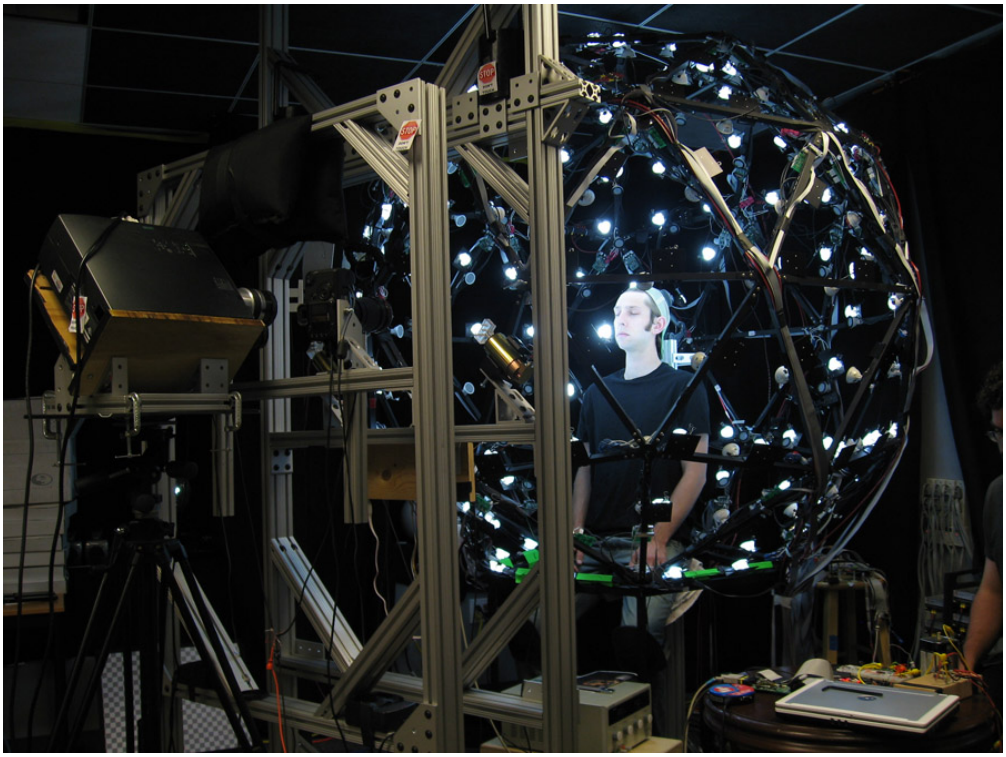


Figure 2.1: The USC light stage[47]. The light stage is an expensive hardware setup with multiple cameras and precisely controlled light sources. Photograph reprinted from [OP2].

2.2 Facial Reconstruction in the Wild

Face reconstruction has long been an active research topic in computer vision community. The general goal of face reconstruction is to recover accurate 3D facial geometry from 2D inputs(images or video). Multiple-view stereo based facial geometry reconstruction is a well studied topic and related technologies have been widely used in industry to create metrically correct human face models. However, multi-view based facial reconstruction usually requires an expensive setup with many high resolution cameras and specialized lighting equipments[48, 7, 49](Figure 2.1).

In contrast to the multi-view stereo face reconstruction under well-controlled environment, using a few unconstrained images, and to the very extreme of only a single image, to recover facial

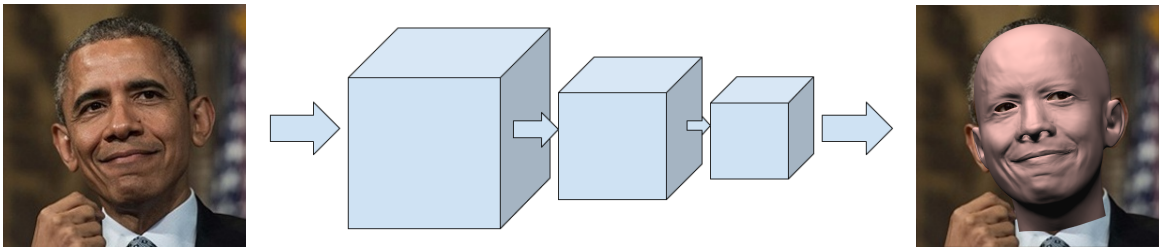


Figure 2.2: Illustration of face reconstruction with deep neural network. The neural network acts as a black box function approximator that mimics the highly non-linear facial reconstruction process. Photograph reprinted from [BO8].

geometry remains a highly challenging problem to date. Recent work of face reconstruction in the wild models a static 3D human face with neutral expression from a set of images [4, 3] using a combination of 3D morphable models and photometric stereo. However, these methods only recover the personalized 3D face identity and ignore facial expressions.

The rapid advancements in machine learning using deep neural network also inspire a series of work on facial reconstructions and face modeling using convolutional neural networks[50, 51, 52]. These methods attempt to model the facial reconstruction process as an inverse mapping from 2D images to 3D facial geometry via highly non-linear functions encoded in deep neural network(Figure 2.2). However, such method typically requires a large amount of high quality training data to achieve similar level of reconstruction quality of previous methods such as[4, 3].

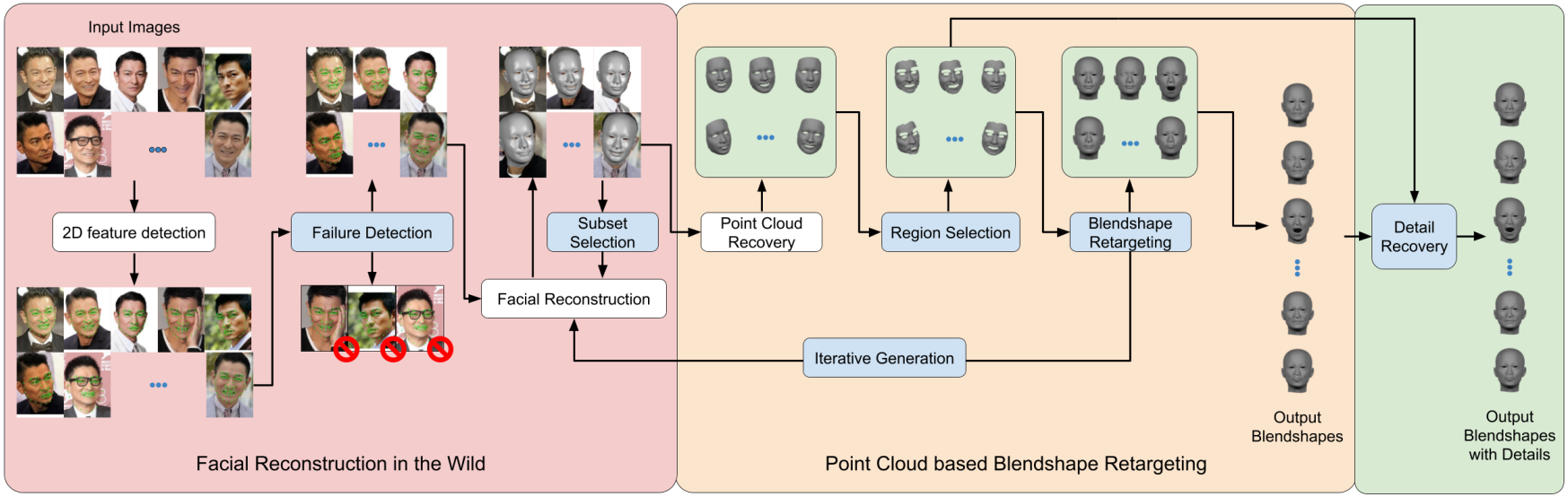


Figure 2.3: The pipeline of our automatic blendshapes generation system. We first detect facial features for all input images. Failed cases in facial feature detection are automatically identified and excluded. We then reconstruct large-scale facial deformation using a multi-linear model. We include a subset selection scheme to obtain consistent facial reconstructions. Next, we recover high resolution point clouds utilizing the large-scale reconstruction and per-pixel visual cues. A set of high fidelity blendshapes are then generated from the point clouds using deformation gradients prior from template blendshapes. We also feed the generated blendshapes back into our pipeline to iteratively improve the quality of the generated blendshapes. Lastly, we recover detailed geometry using the point clouds and the generated blendshapes and integrate it into the final blendshapes. The blue boxes highlight our novel modules in the proposed system. Photographs reprinted from [AL1], [AL2], [AL7], [AL10], [AL22], [AL8], [AL47], [AL26] .

3. OVERVIEW

Our system aims at generating high quality blendshape models directly from images in the wild. Given a set of input images $I_i (i = 1, 2, \cdots N)$ of a subject with unknown head poses (i.e. translations T_i and rotations R_i) and illuminations L_i, as well as a set of template blendshapes B^t, our goal is to construct a set of personalized blendshape bases B for the subject that follow the template blendshapes and can well reconstruct the facial appearance in the input images. Ideally, this can be achieved by optimizing

$$\begin{aligned} \underset{B,w_i,R_i,T_i,\alpha,L_i,c_i}{\operatorname{argmin}} & \sum_{i=1}^{N} \|I_i - \tilde{I}_i(B, w_i, R_i, T_i, \alpha, L_i, c_i)\|^2 \\ & + \lambda \|B - D(B^t)\|^2 \end{aligned} \tag{3.1}$$

where w_i is the blendshape weights for the facial expression in image I_i, c_i is the camera parameters of image I_i, and α is the face albedo of the subject. $\tilde{I}_i$ is the rendered image of the reconstructed 3D face using the generated blendshapes B and estimated parameter set $\{w_i, R_i, T_i, \alpha, L_i, c_i\}$, and $D(B^t)$ is the deformation of the blendshape templates. Directly solving all unknown parameters from the input images is a highly ill-posed problem(Figure 3.1). As shown in Figure 2.3, we instead derive the blendshapes from the input images in two steps: a facial reconstruction in the wild step and a point cloud based blendshape retargeting step.

3.1 Facial Reconstruction in the Wild

In this step, we reconstruct a 3D face for each input image and initialize the blendshapes using the 2D sparse features in each image and the multi-linear model. To this end, our system first detects 2D facial feature points from the input images and then removes inaccurate detection results with a new failure detection algorithm (Section 4.2). We then reconstruct the 3D head pose and facial expression for each input image by fitting the feature points with a multi-linear facial model. The variety of head poses, facial expressions and ambiguity due to the loss of depth

information in the 3D to 2D projection could lead to failure in facial reconstructions. We thus develop an algorithm for automatically identifying a consistent image subset for obtaining good facial reconstructions(Section 4.4). Given the selected image subset, our system performs 3D face reconstruction separately for each image and jointly for all images with the multi-linear model in an iterative way to mitigate 3D-2D ambiguity.

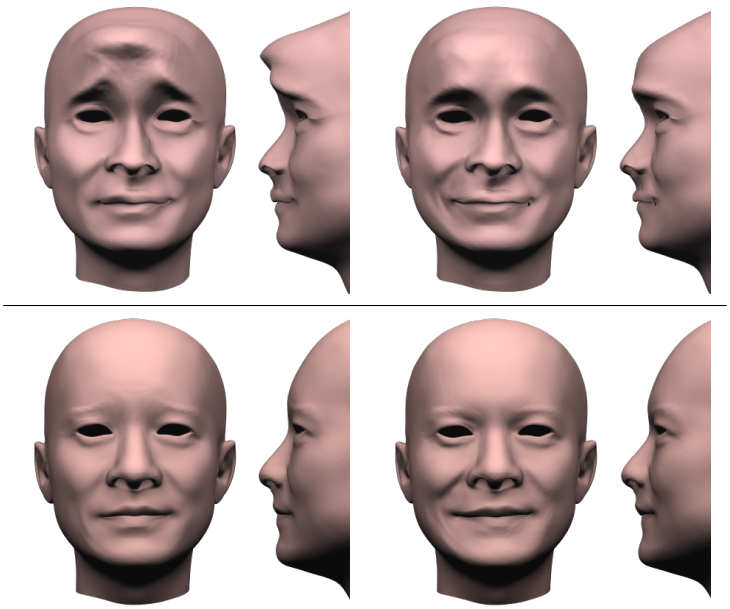


Figure 3.1: Comparison of direct blendshapes optimization and point cloud based blendshapes retargeting. Top: blendshapes generated by direct optimization. Bottom: results by our method using point cloud based blendshapes retargeting. Note that solving the highly ill-posed problem directly via optimization can easily get stuck at local minima and result in incorrect geometry.

3.2 Point Cloud Based Blendshapes Retargeting

The previous step filters out outlier images from the inputs and provides a good estimation for head poses (T_i and R_i) and camera parameters c_i. However, the blendshapes and 3D faces reconstructed with the multi-linear model cannot faithfully resemble the true face geometry due to the limited generality of the multi-linear model. In this step, we refine the blendshapes by fitting the per-pixel facial appearance in the input images with the retargeted blendshape template as in

Equation 3.1. However, this problem is still ill-posed even with known T_i, R_i , and c_i. Moreover, the occlusions in the facial region such as from hair and eyebrow have to be detected and removed from the image for this optimization. To tackle these challenges, we introduce the 3D point cloud P_i of a face as an intermediate representation and perform the optimization iteratively in three steps. We first recover a high resolution point cloud P_i for each input image using shape-from-shading by optimizing

$$\underset{P_i,\alpha,L_i}{\operatorname{argmin}} \|I_i - \tilde{I}_i(P_i(B, w_i), R_i, T_i, \alpha, L_i, c_i)\|^2 \tag{3.2}$$

where the 3D point cloud is constrained by the 3D face modeled by current blendshapes and weights to resolve the bas-relief ambiguity (Section 5.1). A novel region selection method is then applied to automatically detect confident regions in the point clouds and exclude undesired noise in the point cloud caused by occlusions. Finally, we significantly extend the retargeting algorithm in [8]. For generating personalized blendshapes (Section 5.2), we fit the high resolution point clouds with the retargeted blendshape templates

$$\underset{B,w_i}{\operatorname{argmin}} \sum_{i=1...N} \|P_i - B \otimes w_i)\|^2 + \lambda \|B - D(B^t)\|^2 \tag{3.3}$$

Here $B \otimes w_i$ is the linear combination of blendshape basis B using the weights w_i. After this step, we use the resulting blendshape to update the point cloud reconstruction in the next iteration. The optimization is executed until the resulting blendshapes are not changed.

3.3 Detail Recovery

Finally, our system recovers fine-scale facial geometry using a novel algorithm based on surface normal regression. Deformation/displacement based representations of the geometric details are commonly used in video-based methods[2, 15, 1]. However, this method does not work well in our case due to the extremely challenging input. Typically, displacement-based methods typically suffer from over-smoothing and undesired distortion caused by the bas-relief effect. We develop a surface normal based representation to address this issue. We build a PCA model of the difference between the surface normal of the high resolution point clouds and the synthesized shapes by the

generated blendshapes. An optimal mapping from expression weights to PCA coefficients is then computed with ridge regression and used to synthesize detailed normal maps for new expressions (Chapter 6).

4. FACIAL RECONSTRUCTION IN THE WILD

Given a set of input images, we recover the head poses and reconstruct the 3D facial model using the multi-linear model in this step. To this end, we first detect the 2D facial feature points from input images using a recently developed regression-based face alignment technique[53]. This technique iteratively refines an initial estimate of feature point locations using random regression forests with local binary features. Although this algorithm generally is robust for a wide range of head poses and facial expressions, it still fails in some challenging cases(Figure 4.2 (b)). These inaccurate detection results will lead to poor facial reconstructions. We thus apply a new robust PCA based algorithm to evaluate the quality of the detected 2D features and automatically exclude images with inaccurate detection results.

We then fit the detected 2D feature points with a multi-linear facial model for reconstructing head poses and 3D face shapes modeled by multi-linear weights. Without depth information, the sparse 2D features in a single image can be explained by different combinations of head poses, expressions or identities and thus lead to inconsistent reconstruction results (as shown in Figure 4.1). To tackle this challenge, our key idea is that all input images share the same identity. Therefore, we can identify good candidates from all possible reconstruction results by estimating their identity weights.

To this end, we develop a progressive algorithm for reconstructing 3D facial models from the input images. Our algorithm consists of three steps: individual reconstruction, subset selection and joint reconstruction. We first reconstruct face shapes for each input image individually by solving a non-linear optimization problem. Specifically, we follow the method in [2] to seek the head poses (rotation R and translation T), camera parameters c, and identity and expression weights(w_{id} and w_{exp}) of a multi-linear model that best fit the detected 2D features of the input image. We perform the optimization without the constraint of uniform identity weights to maximize our exploration of the potential identity weights. The estimated parameters are then used to initialized the unknowns in the joint reconstruction step. However, the individual reconstruction results are not always

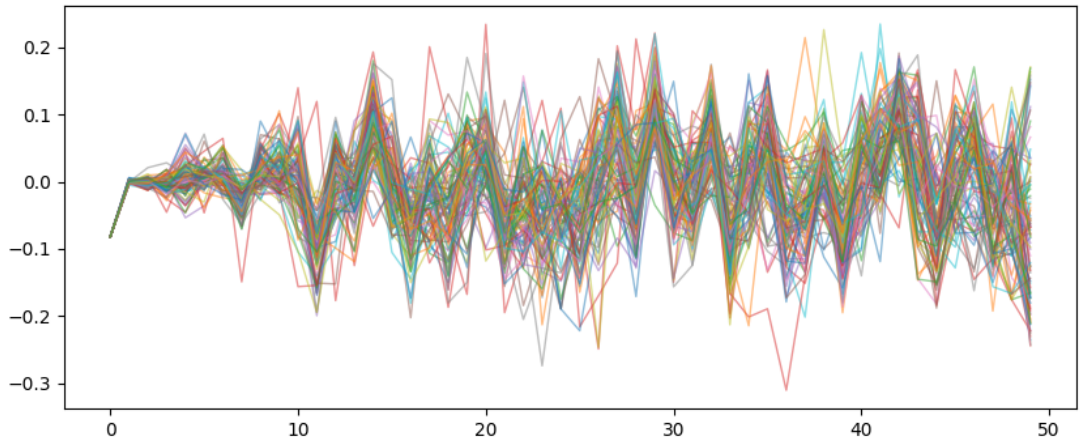


Figure 4.1: Inconsistent identity weights from direct reconstruction of 100 images of Andy Lau. The horizontal axis represents the 50 dimensions of the identity weights. Each polyline in the plot represents an identity weight vector of a reconstruction.

reliable due to the inherent ambiguity of estimating 3D shape from 2D constraints. We thus filter these results with a subset selection step to find a consistent subset suitable for joint reconstruction. During joint reconstruction, we solve a unified optimization similar to [2] over the selected subset with common identity weights w_{id}^*. We then use these common identity weights as the initial guess for per-image identity weights, and iterate the reconstruction process a few times to achieve the best reconstruction results.

In the following sections of this chapter, we introduce the technical details of the key components in our system: parametric 3D face models, facial feature points detection and robust PCA based failure detection, face reconstruction with multi-linear model and subset selection.

4.1 Parametric 3D Face Models

Estimating 3D facial geometry from 2D images is a typical ill-posed problem because of the goal to recover 3D shape in the absence of depth information. Prior knowledge about the 3D facial shape, which provides strong constraints for viable solutions, is thus crucial in solving this problem. Previous work typically encode such prior knowledge in the form of some parametric

models, such as the PCA model used in 3D morphable model[23] and the multi-linear models in [24]. With the prior model, the problem of recovering 3D face shape can be simplified to estimating model parameters. For example, given a PCA shape model $\{A_j\}$, a 3D shape S can be represented using the mean shape A_0 and principal components A_i:

$$S = A_0 + \sum_i w_i A_i$$

where $\{w_i\}$ is the model parameters, while S and $\{A_i\}$ are the vector formed by stacking the coordinates of all vertices together. Recovering 3D face shape using a PCA model can then be formulated as estimating the model parameters $\{w_i\}$.

PCA models are typically used to model neutral face shape[23], but it can also be extended to include facial expressions using extra components:

$$S = A_0 + \sum_i w_i A_i + \sum_j w_j^e A_j^e$$

where $\{w_j^e\}$ are the weights for interpolating the expression components $\{A_j^e\}$

PCA models provides a compact representation of 3D face shapes and allows synthesizing different face shapes through linear interpolation. However, PCA models are typically difficult to interpret and manipulate. The principal components form an orthogonal shape space that each principal component encodes independent shape variations that could combine changes in several different region on the face. For example, different sizes of nose and chin could be combined into a single principal component so that the PCA model is more compact. This makes it difficult to directly control the model parameters to achieve desired face shape using PCA models. Similarly, each expression component in a PCA model could encode the motion of several facial muscles simultaneously, making it difficult to support fine grain expression control.

Blendshape models, on the other hand, provides a linear representation of 3D face that is easy to interpret and manipulate. Similar to PCA models, a 3D face can be synthesized by linearly

interpolating the basis in a blendshape model:

$$S = B_0 + \sum_j w_j (B_j - B_0)$$

where B_0 is the neutral face shape and $\{B_j\}$ are the face basis with different expressions. The above formulation is the so-called *additive formulation* since the difference between neutral face and a specific expression B_j is added to the neutral face with weight w_j.

The linear basis in a blendshape models is a set of predefined expressions with clear semantics, *e.g.*, mouth open, eye closed, etc. Artists can easily adjust the linear combination weights to achieve specific facial expression with blendshapes because of it. Similar to PCA-based models, we can recover the 3D face shape of a person using a set of blendshapes built for the person by estimating a proper combination of expression weights.

A major difference between blendshapes and PCA-based model is that a set of blendshape models a single person only. Multi-linear models generalizes blendshapes to model face variations in several separable attributes simultaneously. For example, bi-linear models with variations in both identity and expression is formed by stacking multiple sets of blendshapes with consistent expression variations together. Intuitively, the 3D face shapes in a bi-linear model are organized in a matrix where each element is a single face shape for a specific person and expression:

	Expression 1	Expression 2	...	Expression M
Person 1	S_{11}	S_{12}	...	S_{1M}
Person 2	S_{21}	S_{22}	...	S_{2M}
⋮	⋮	⋮	⋮	⋮
Person N	S_{N1}	S_{N2}	...	S_{NM}

where S_{ij} is the face shape for the i-th person and the j-th expression. With a multi-linear model, a face shape can be generated by linear interpolation in all attributes. For example, a face shape can be generated through interpolation in both identity and expression dimensions.

Formally, a multi-linear model is typically represented as a high order tensor $\mathcal{T} = (t_{i_1 i_2 \ldots i_N})$

that can be linearly interpolated to produce a specific face shape[24]. Different modes of the tensor are used to encode different attributes of the data. For example, Vlasic et al.[24] constructed a multi-linear model with 3 modes, i.e. "identity", "expression" and "viseme", and store the data as a 3-tensor.

A specific face shape can be synthesized with a multi-linear model through linear interpolation. The interpolation operation can be expressed as mode-n product. Specifically, the interpolation of an attribute is expressed as a mode product of the mode associated with that attribute. For example, the interpolation of the "identity" attribute can be expressed as $\mathcal{T} \times_{id} w_{id}^T$, where w_{id} is the interpolation weights. Given a 3 modes multi-linear face model with "identity", "expression" and "viseme", a face shape can be synthesized as

$$\mathcal{S} = \mathcal{T} \times_{id} w_{id}^T \times_{exp} w_{exp}^T \times_{vis} w_{vis}^T$$

where w_{exp} and w_{vis} are the interpolation weights for expression and viseme, respectively.

The size of a high order tensor representing a multi-linear face model can grow rapidly with the increase of data samples in the tensor, while the new information contained in the new data samples may be incremental. It is therefore very common to perform tensor decomposition to compress a large tensor $\mathcal{T}$ into a much more compact core tensor $\mathcal{C}_r$. This can be achived by performing N-mode SVD on the original tensor $\mathcal{T}$[54].

Since multi-linear model is parameterized by several attributes, recovering 3D shape using multi-linear model boils down to estimating proper combination of the weights for all relevant attributes. In the case of bi-linear model, we need to estimate both identity and expression weights to recover a 3D face shape.

4.2 Feature Point Detection and Robust PCA Based Failure Detection

Reconstructing 3D face shape from 2D image in its essence is determining 3D location of each pixel in the 2D image. This is an ill-posed problem since 2D images typically do not provide any depth information. With a prior model, recovering 3D face shape reduces to the problem of

estimating proper model parameters, along with camera parameters and head poses. To find out the model parameters, we need to determine the correspondence between the 3D model and the pixels in 2D images and use them as constraints.

Given a 3D shape prior model, the correspondence between 3D points on the model and 2D pixels is unknown. Without extra knowledge, it is impossible to establish dense correspondence between the 3D model and 2D pixels. On the other hand, it is relatively easy to identify several distinct facial feature points in the 2D image and their corresponding 3D points on the face model. For example, mouth corners, eye corners, nose tip are distinct feature that could be identify with significantly less effort in 2D images and on the 3D model. To this end, we utilize the correspondence between important 2D and 3D facial feature points to estimate the parameters in the prior model.

The 3D facial feature points can be manually labeled since all the 3D meshes in the prior model have identical topology, i.e. the location of 3D facial feature points are fixed on the 2D manifold of the 3D face regardless identity and expression change. For 2D facial feature points, we use the random forest based cascading regression method[53] to detect them in the input images. Facial keypoints detection techniques improved significantly in recent years since the introduction of cascaded regression method[55]. Cao and colleagues adapted the cascaded pose regression algorithm to work on facial feature points detection and achieved impressive results. Supervised decent method proposed by Xiong et al.[56] formalized the cascaded regression method in the supervised gradient descent framework. Ren and colleagues [53] further improved both detection quality and speed of the detection algorithm with combination of random forests and local binary features. Generally, the facial feature detection algorithms initialize the face shape with a template shape S^0 inside the detected face region, and compute a series of shape incremental $\{\Delta S^t\}$ in a cascade of several stages:

$$\Delta S^t = W^t \Phi^t(I, S^{t-1})$$

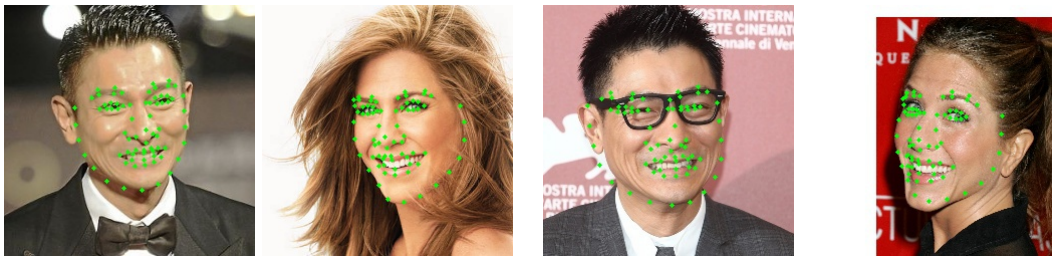
where Φ^t is a function that extracts useful features from input image I using the predicted face shape in the previous stage S^{t-1}, and W^t is a matrix that maps the feature vectors extracted by Φ^t

to shape incremental ΔS^t. The final prediction of the face shape is the summation of the initial face shape S^0 and all the shape incrementals:

$$S^{final} = S^0 + \sum_{t=1}^{T} \Delta S^t$$

where T is the total number of stages in the cascade.

Despite the improvements in the detection quality of facial feature points detection algorithms, these detectors can still fail on challenging input images. The input images our system handles usually contains extreme head poses, facial expressions and illuminations, all of which are very likely to cause failure in the facial feature points detection algorithm. As shown in Figure 4.2, the facial feature detection may fail for some input images.



(a) (b)

Figure 4.2: Examples of facial feature points detection. (a) successful cases; (b) failed cases: inaccurate eyes, nose and contour points(*left*) and drifted contour points(*right*). Photographs reprinted from [AL1], [OP3], [AL38], [OP4], .

The quality of the predicted facial feature points are critical for obtaining good large-scale reconstruction results. Inaccurate facial feature points directly lead to failure in large-scale facial reconstruction, and such error would propagate through our system causing disastrous results. While improving the quality and robustness of the detection algorithm could reduce the failure rate, it is extremely difficult to make sure the detection algorithm generate good results given arbitrary images in the wild. To make our large-scale facial reconstruction work for images in the

wild, we need to filter out failure cases in feature points detection results.

Our key idea of finding the failed detection results is based on the observation that for failed detection, either the face shape (i.e. the polygonal shape defined by the feature points) or the texture inside the face shape are significantly different from that of a good detection. However, it is impractical to compare the detected face regions in different images directly because of the large variations of head poses, expressions and illuminations. We therefore construct a statistical model for the detected face shape and texture simultaneously, and compare the detection results using the model. Similar to the active appearance model[57], our model consists of a shape model $\mathcal{M}_s$ and a texture model $\mathcal{M}_t$. These two models are constructed by performing PCA on the input data.

Algorithm 1 Failure detection for facial feature points

Input: Source image and feature points pairs $\{I_i, \{\mathbf{p}_{ij}\}\}$
Output: Set of inliers $O = \{i_k\}$

inliers $\leftarrow \{I_i, \{\mathbf{p}_{ij}\}\}$
while not converged **do**
  $\mathcal{M}_t, \mathcal{M}_s \leftarrow$ RobustPCA(inliers)
  for all Input Image and feature points pair $(I_k, \{\mathbf{p}_{kj}\})$ in current_set **do**
    $I'_k, \mathbf{p}'_{kj} \leftarrow$ PCAReconstruction$(\mathcal{M}_t, \mathcal{M}_s, I_k, \{\mathbf{p}_{kj}\})$
    $e_k \leftarrow w_t$RMSE$(I_k, I'_k) + w_s \sum_j \|\mathbf{p}_{kj} - \mathbf{p}'_{kj}\|^2$
  end for
  $\mu, \sigma =$ ComputeMeanAndStd$(\{e_k\})$
  inliers $\leftarrow$ inliers $- \{i_k$ for $e_k > \mu + \omega\sigma\}$
end while
return inliers

A key difference in our model is that the input data are corrupted with outliers, i.e. failed detection results. A straightforward PCA model will capture the noise from the outliers and the model would not be able to distinguish good detections from bad ones. We instead use robust PCA[58] to factor out the outliers and construct a model for the denoised data. Consider the texture model $\mathcal{M}_t$, we know the input texture data X_t is corrupted with failure cases. The uncorrupted texture data is the facial texture under different illumination, which can be well approximated with

PCA[59]. The texture data X_t therefore can be decomposed into a low-rank component L_t and a sparse noise component S_t:

$$X_t = L_t + S_t$$

To find out outliers in current texture data X_t, we need to estimate the low-rank component L_t from the corrupted texture data X_t. Classical PCA is able to achieve similar goal since it seeks a low-rank approximation $\hat{L}_t$ of X_t by solving the following optimization problem

$$\begin{aligned} &\text{minimize } \|X_t - \hat{L}_t\| \\ &s.t. \text{ rank}(L) \leq r \end{aligned}$$

where r is the a number smaller than rank(X_t). However, the noisy component S_t is not excluded in the low-rank representation $\hat{L}_t$ in classical PCA method. In our case, the classical PCA captures the noisy texture information from the failed feature point detection results, which defeats our purpose of using the PCA model to identify outliers.

Robust PCA[58], on the other hand, models the sparse noise explicitly in its optimization that seeks the low-rank representation L_t:

$$\begin{aligned} &\text{minimize } \|L_t\|_* + \lambda\|S_t\|_1 \\ &s.t. \; L_t + S_t \approx X_t \end{aligned} \tag{4.1}$$

where $\|L_t\|_*$ is the nuclear form of L, i.e. the sum of the singular values of L. λ controls how much information we intend to encode in the sparse noise component S_t. Note in this formulation, the noisy texture information can be encoded into S_t so that L_t closely approximates the uncorrupted texture information. We choose $\lambda = 1/\sqrt{\max(N_r, N_c)}$ following [58], which is shown to be the optimal choice that gives exact solution of the optimization problem in Equation. 4.2 and produces reasonably good results. N_r and N_c are the number of rows and columns in X_t, respectively.

We construct both the texture and shape PCA model by first performing a low-rank decomposition of the data matrix X_* to obtain its low-rank representation L_*, which is the denoised data

matrix. A regular PCA model is then constructed directly using the denoised data matrix. We then use the constructed PCA model to fit the input data vectors, and the noisy input data vectors can be identified as the ones with large reconstruction error. Since we do not have any information about how much corruption is in the data at the beginning, we use an iterative approach to gradually remove outliers. We stop the outlier removal process when no more outliers can be identified. The failure detection algorithm is summarized in Algorithm 1.

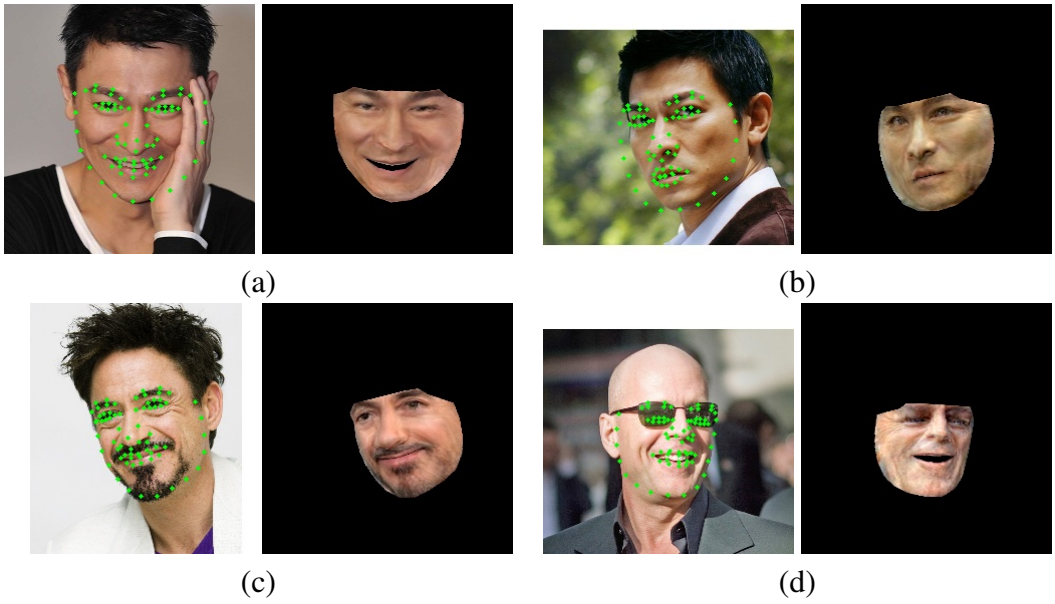


Figure 4.3: Results by our failure detection algorithm. For (a) to (d): the left image is input images with green dots marking the detected facial feature points, and the right image is the fitted face region by our texture model. (a) Failure due to occlusion by hand; (b) Failure due to drifted contour points; (c) Failure due to facial hair: feature point around the nose are inaccurate; (d) Failure due to occlusion by glasses: feature points for eyes are inaccurate. Photographs reprinted from [AL10], [AL22], [OP5], [BW11].

This robust model contains personalized face shape and texture information that captures reasonable variations in head poses, expressions and illuminations. It is able to fit the good detection results nicely but not for failed detections (Figure 4.3). We measure the difference between the detection results and reconstruction results, and use it to determine whether the detection failed for a specific image/feature points pair. We experimentally set $w_t = 1.0$ and $w_s = 0.1$, and the

algorithm converges after 3 to 5 iterations. Figure 4.8 shows a typical case of the outliers detected by the robust PCA based method.

4.3 Face Reconstruction with Multi-linear Model

With the images selected in the previous step, we create the initial facial shape reconstructions using multi-linear models [24, 35] as 3D shape prior. We generate the shape reconstructions using a 2 dimensional multi-linear model with variations in “identity” and “expression”. The model used in our experiments is created using the facial data from FaceWarehouse[35], which consists of 150 identities and 47 facial expressions. Our bi-linear model is a reduced model with 50 identity parameters and 25 expression parameters created by performing HOSVD[54] for the original tensor. In this model, a face shape can be represented as a linear interpolation in the "identity" and "expression" dimensions:

$$\begin{aligned} S &= R(\mathcal{C}_r \times_{id} w_{id}^T \times_{exp} w_{exp}^T) + T \\ &= R(\mathcal{C}_r \times_{id} w_{id}^T \times_{exp} U_{exp}\hat{w}_{exp}^T) + T \end{aligned} \tag{4.2}$$

where S is the face shape generated with the multi-linear model $\mathcal{C}_r$, identity weights w_{id} and expression weights w_{exp}. U_{exp} is the expression weights mapping matrix from the original 47 dimensional expression space to the 25 dimensional orthogonal expression space, and $\hat{w}_{exp}$ a 47 dimensional expression weights vector. R and T depict the rigid transformation of the face shape.

4.3.1 Shape Reconstruction Algorithm

To recover large-scale facial deformation, we need to estimate the camera parameters c_i, head poses (R_i, T_i), identity weights w_{id} and expression weights $w_{exp,i}$ for all input images. Using the detected 2D facial features and their corresponding 3D vertices, we formulate the shape reconstruction as a non-linear optimization problem to minimize the reprojection error of the facial feature

points:

$$\begin{aligned}\underset{c_i,R_i,T_i,w_{id},w_{exp,i}}{\arg\min} (&\sum_{i}^{N}(E_{feat}(c_i,R_i,T_i,w_{id},w_{exp,i}) + w_1 E_{exp}(w_{exp,i}) + w_2 E_{sp}(w_{exp,i})) \\ &+ w_3 E_{id}(w_{id}))\end{aligned} \tag{4.3}$$

The main cost function term in the above optimization problems is the reprojection error term E_{feat}, which measures the distance between the reprojected location of the labeled 3D vertices and the corresponding detected 2D facial feature points. We also include several regularization terms, E_{id}, E_{exp} and E_{sp} to stabilize the optimization process.

Reprojection Error Term Assuming a weak perspective camera model, the 2D locations of the facial feature points $\{\mathbf{p}^{(k)}\}$ can be generated by projecting their corresponding 3D vertices $\{\mathbf{v}^{(k)}\}$ to the image plane:

$$\begin{aligned}\mathbf{p}^{(k)}(c,R,T,w_{id},w_{exp}) &= \pi(c)R\left(\mathcal{C}_r \times_{id} w_{id} \times_{exp} w_{exp}\right)^{(k)} + T \\ &= \pi(c)R\left(\mathcal{C}_r \times_{id} w_{id} \times_{exp} U_{exp}\hat{w}_{exp}\right)^{(k)} + T\end{aligned} \tag{4.4}$$

where $\pi(c)$ is the weak perspective projection that is a function of camera parameters c. The reproject error for the facial feature points is therefore

$$E_{feat}(c,R,T,w_{id},w_{exp}) = \sum_{k=1}^{N_k} \|\mathbf{p}^{(k)}(c,R,T,w_{id},w_{exp}) - \mathbf{q}^{(k)}\|^2 \tag{4.5}$$

where $\mathbf{q}^{(k)}$ is the 2D location of the k-th feature point obtained from the face alignment step, and N_k is the number of facial feature points available in an image.

Regularization Terms Due to the sparsity of the constraints from the facial feature points, it is necessary to regulate the non-linear optimization with extra constraints to avoid local minima and overfitting. The regularization terms we used are two prior terms to minimize the deviation of the

reconstructed identity weights and expression weights from their prior values, respectively:

$$\begin{aligned} E_{id}(w_{id}) &= (w_{id} - w_{id}^{prior})^T \Sigma_{id}^{-1} (w_{id} - w_{id}^{prior}) \\ E_{exp}(w_{exp}) &= (w_{exp} - w_{exp}^{prior})^T \Sigma_{exp}^{-1} (w_{exp} - w_{exp}^{prior}) \\ &= (U_{exp}\hat{w}_{exp} - w_{exp}^{prior})^T \Sigma_{exp}^{-1} (U_{exp}\hat{w}_{exp} - w_{exp}^{prior}) \end{aligned} \tag{4.6}$$

E_{id} is the regularization term for identity weights and E_{exp} is the for expression weights. w_{id}^{prior} and w_{exp}^{prior} are the mean identity and expression weights are the mean identity and expression weights for all people, while Σ_{id} and Σ_{exp} are the covariance matrices for identity and expression weights.

We additionally introduce a sparsity term for the 47 dimensional expression weights to improve robustness of the optimization process, since the 47 dimensional expression space is not orthogonal:

$$E_{sp}(\hat{w}_{exp}) = \|\hat{w}_{exp}\|_1 \tag{4.7}$$

Algorithm 2 Face Shape Reconstruction. $S_i^{init} = (R_i, T_i, c_i, w_{id,i}, w_{exp,i})$ is the set of parameters representing the reconstructed shape for i-th input image, and w_{id}^* is the identity weight for the target person.

Input: Source image and feature points pairs $\{I_i, \{\mathbf{p}_{ij}\}\}$
Output: Parametric reconstruction results $\{S_k^{init}\}$

$w_{id}^* \leftarrow \bar{w}_{id}$
while not converged **do**
  for all Input Image and feature points pair $(I_i, \{\mathbf{p}_{ij}\})$ **do**
    $w_{id,i} \leftarrow w_{id}^*$
    $S_i^{init} \leftarrow$ IndividualReconstruction$(I_i, \{\mathbf{p}_{ij}\})$
  end for
  $K \leftarrow$ SelectGoodReconstructions$(\{S_i\}, \{I_i\}, \{\{\mathbf{p}_{ij}\}\})$
  $w_{id}^* \leftarrow$ JointReconstruction$(\{S_k\}, \{I_k\}, \{\{\mathbf{p}_{kj}\}\}), k \in K$
end while
for all $k \in K$ **do**
  $S_k^{init} \leftarrow$ IndividualReconFixedIdentity$(w_{id}^*, I_k, \{\mathbf{p}_{kj}\})$
end for
return $\{S_k^{init}\}$

The optimization problem defined in Equation. 4.3 is very difficult to solve without good initialization. This is because it seeks to minimize the cost function in a high dimensional space with limited number of constraints. In a typical case with 100 input images, the solution space has 5650 dimensions, while the number of available constraints is 7400. Since the dimensionality of the solution space is so large and the number of available constraints is similar to the dimensionality of the solution space, solving the optimization directly is very challenging and could easily ends up with poor solutions. In this case, constraining the problem to a lower dimensional solution space could significantly simplify the problem and could potentially lead to feasible solutions.

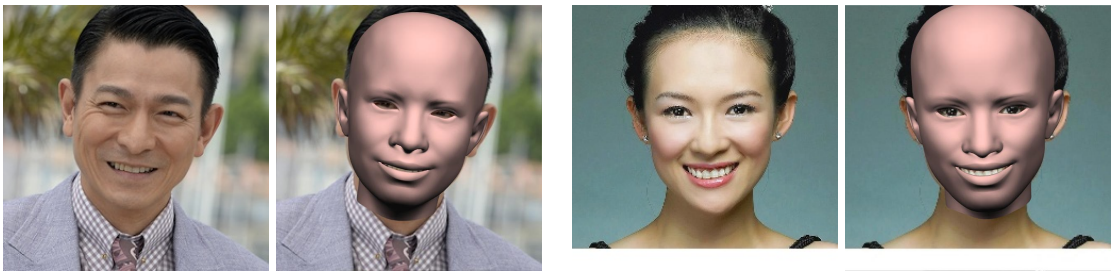
To this end, the initial reconstruction is done iteratively with 3 major steps: individual reconstruction, subset selection and joint reconstruction, as is outlined below. In the individual reconstruction step, we recover the 3D face shape for each input image individually. The purpose of initial reconstruction is to produce an initial face shape estimation for the input images, as well as an estimation of a common identity weights that fits best with all input images. For individual reconstruction, the solution space is much more manageable since the number of unknowns is only 106. Even with only 74 facial feature points as constraints in each input image, the optimization is still well-behaved with proper regularization. The individual reconstruction step produces a series of reconstruction results, each with a unique identity weights vector. Due to the inherent ambiguity of recovering 3D shape from 2D images, some individual reconstructions results may not be reliable. Including such results in the joint reconstruction step could pollute the joint optimization process and lead to worse reconstructions. The individual results are therefore filtered in the subset selection step that finds out a consistent subset of reconstruction results suitable for joint reconstruction. In joint reconstruction, all selected inputs are included in a single optimization problem that solves for the identity weights best explains the input data. The three steps are repeated several times until convergence.

4.3.2 Individual Reconstruction

We first reconstruct a face shape for each input image individually by solving the following non-linear optimization problem:

$$\underset{R,T,c,w_{id},w_{exp}}{\arg\min} \; (E_{feature}(c,R,T,w_{id},w_{exp}) + w_1 E_{id}(w_{id}) + w_2 E_{exp}(w_{exp}) + w_3 E_{sparsity}(w_{exp})) \quad (4.8)$$

The weights for the cost function terms are determined experimentally that strike a good balance among the these terms. Their values decreases gradually in the optimization process. We initialize $w_1 = 10, w_2 = 5$ and $w_3 = 10$ in all of our experiments.



(a) (b)

Figure 4.4: Individual reconstruction results. For both (a) and (b): input images (left) and result of individual reconstruction (right). The face shape and expressions matches the input image generally, but facial geometry is apparently different from the input images. Photographs reprinted from [AL26] and [ZZ2].

The main purpose of individual reconstruction is to obtain good initial guess for the joint reconstruction. We achieve this by reconstructing face shapes for each input image without the constraint of uniform identity weights. This maximizes our exploration of the potential identity weights region in the weight space and gives us a series of reconstructions that best fit the detected facial feature points. The individual reconstructions provide good initial guesses for the joint optimization both for the identity weights and the expression weights. The initial estimation for head poses is also obtained in this step since they are easier to optimize for individual image.

When generating the initial reconstructions in the first pass, the prior value for both identity weights and expression weights are set to the mean value in the multi-linear model. In the subsequent passes, the prior value for identity weights is set to the value computed by joint reconstruction in previously. Since the reconstruction for each image is solved individually, we have M sets of parameters after this step, each has its own identity and expression weights. Note that the identity weights are different for each input image.

4.3.3 Joint Reconstruction

We perform joint reconstruction to estimate a common identity weights after individual reconstruction. Before performing joint reconstruction, we introduce a subset selection step to address the issues of the naïve reconstruction method. The details of the subset selection method is discussed in Section 4.4.

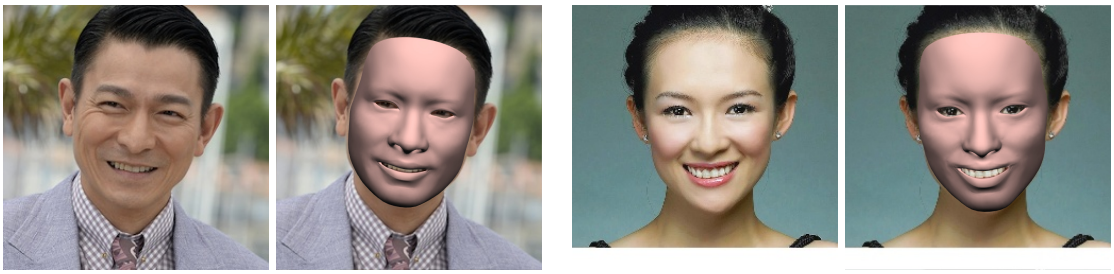
Once a consistent subset is selected, we use the selected reconstruction results to solve the following optimization problem:

$$\underset{w_{id},\{R,T,c,w_{exp}\}_k}{\arg\min} \sum_{k\in K}(E_{feature}(c_k, R_k, T_k, w_{id}, w_{exp,k})+w_1 E_{exp}(w_{exp,k})+w_2 E_{sparsity}(w_{exp,k}))+w_3 E_{id} \tag{4.9}$$

where K is the selected subset from previous step. The main purpose for joint optimization is to enforce uniform identity weights for all reconstructions. The prior value for identity weights used in this step is set to the centroid weight vector computed in the subset selection step. The weights w_1 and w_2 are chosen similar to that in the individual reconstruction step.

4.3.4 Progressive Reconstruction

A naïve optimization with all input images simultaneously produces poor reconstructions due to issues such as 2D-3D ambiguity. We address this issue by selecting subsets of inputs for joint optimization and reconstructing the face shapes progressively. We start with reconstructing for each input image individually, then select a good subset of individual reconstructions to perform joint optimization. The joint optimization produces a uniform identity weights closer to the true



(a) (b)

Figure 4.5: Joint reconstruction results. For both (a) and (b): input images(left) and result of joint reconstruction(right). The results by joint reconstruction significantly improves the facial geometry compared to individual reconstruction, however, their appearance is still limited by the generalization ability of the multilinear model. Photographs reprinted from [AL26] and [ZZ2].

identity weights since it is the weights that best explains multiple input images. We then use the estimated identity weights from joint optimization as a stronger prior to repeat the entire reconstruction process again to obtain better reconstructions. This iterative reconstruction process is repeated several times until convergence is reached. In our experiments 3 iterations are sufficient. We experimentally set the fraction of individual reconstructions chosen for joint reconstruction to 0.4, 0.6 and 0.8 in each iteration. Note that we have a set of per-image shape parameters and a uniform identity weights after the initial reconstruction.

4.4 Subset Selection

While the failure detection algorithm described in the previous section successfully excludes failed facial feature detection results, the 2D-3D ambiguity issue could still cause poor reconstruction results. On the other hand, the failure detection algorithm itself may not exclude all non-ideal detections. For example, skin-tone occlusions caused by hands can deceive the failure detection, while slight drift of feature points caused by extreme lighting could similarly escape detection. Such bad detection results could hurt the reconstruction quality if we simply use all images with "good" detection for joint reconstruction. We therefore need to carefully select a good subset of individual reconstructions for joint estimation of the identity weights.

We developed a dynamic subset selection method to choose a subset of individual reconstruc-

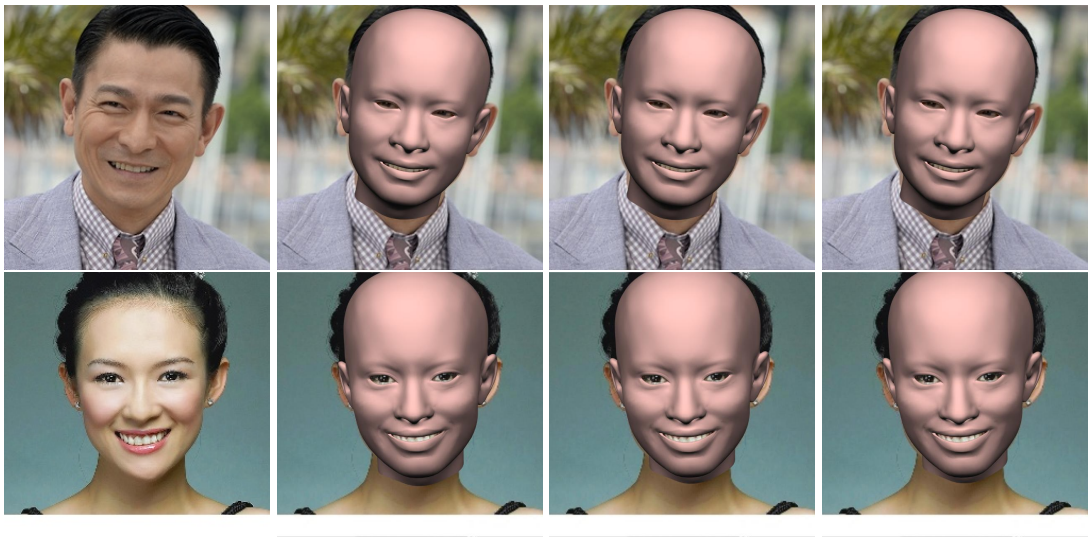


Figure 4.6: Progressive reconstruction results. From left to right: input image, reconstruction result after 1, 2, and 3 iterations. Progressive reconstruction fine-tunes the head poses, identity and expressions simultaneously. Photographs reprinted from [AL26] and [ZZ2].

tion results for joint reconstruction. Our key idea is that a good subset should be consistent in terms of identity weights and have good photo-consistency. We formulate the subset selection as a ranking problem: given a list of individual reconstruction results, we select the best k results for joint optimization. We use the following metrics for ranking the individual reconstruction results:

- Identity weights fitness F_{id} computed as the distance between $w_{id,i}$ from the individual reconstruction and the mean identity weights $\bar{w}_{id}$ of all reconstructions:

$$F_{id,i} = (w_{id,i} - \bar{w}_{id})^T \Sigma_{id} (w_{id,i} - \bar{w}_{id}) \tag{4.10}$$

  Here Σ_{id} is the covariance matrix of the current estimations of identity weights.

- Photo-consistency F_{tex} is the RMSE of the face region between the input image and the reconstructed image using a dynamic texture model.

The texture model used here is similar to the previously used model in failure detection. The key difference here is that we extract the texture from the polygon bounded by the reprojected vertex

locations rather than the detected 2D feature locations. This way the texture model captures how well a reconstruction fits the corresponding input image. We then compute the photo-consistency value and use it in the final ranking score.

We combine these two metrics linearly into a single ranking score:

$$\text{score}(w_{id}, F_{tex}) = w_1 F_{id} + w_2 F_{tex} \tag{4.11}$$

A fraction of the individual reconstruction results are then selected for joint reconstruction based on the ranking score. We experimentally set $w_1 = 1.0$ and $w_2 = 10.0$ for the combined ranking score. Figure 4.7 shows some outputs of our selection algorithm.

In our experiments three iterations are sufficient for progressive reconstruction. We experimentally set the fraction of individual reconstructions chosen for joint reconstruction to 0.4, 0.6 and 0.8 in each iteration. This results in 18, 27 and 49 images for a typical case with 93 input images. Figure 4.9 and Figure 4.10 show a typical example of the subset selected and excluded from joint reconstruction.

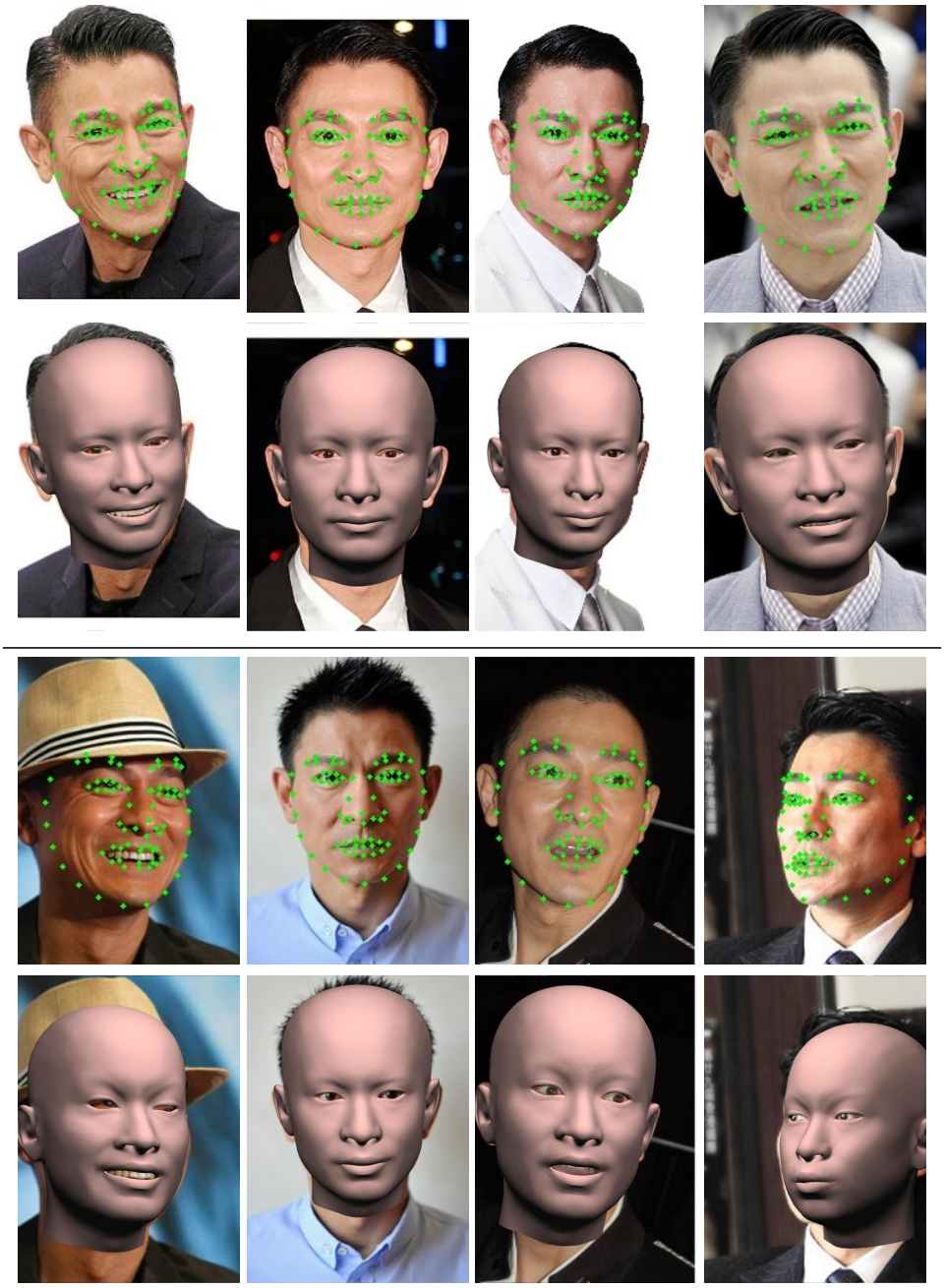


Figure 4.7: Example of reconstruction result selection. Note that the reconstructions here are not our final results, but they are intermediate results obtained using multi-linear models only. Top: Part of the good results selected for joint optimization; Bottom: Part of the results excluded from joint optimization. They are considered inconsistent because of inaccurate feature point locations caused by occlusion, cast shadow, or extreme lighting. Note that the feature points are very close to the correct locations. Our subset selection algorithm successfully identifies these non-ideal reconstructions. Photographs reprinted from [AL2], [AL5], [AL7], [AL85], [AL88], [AL13], [AL60], [AL43] .

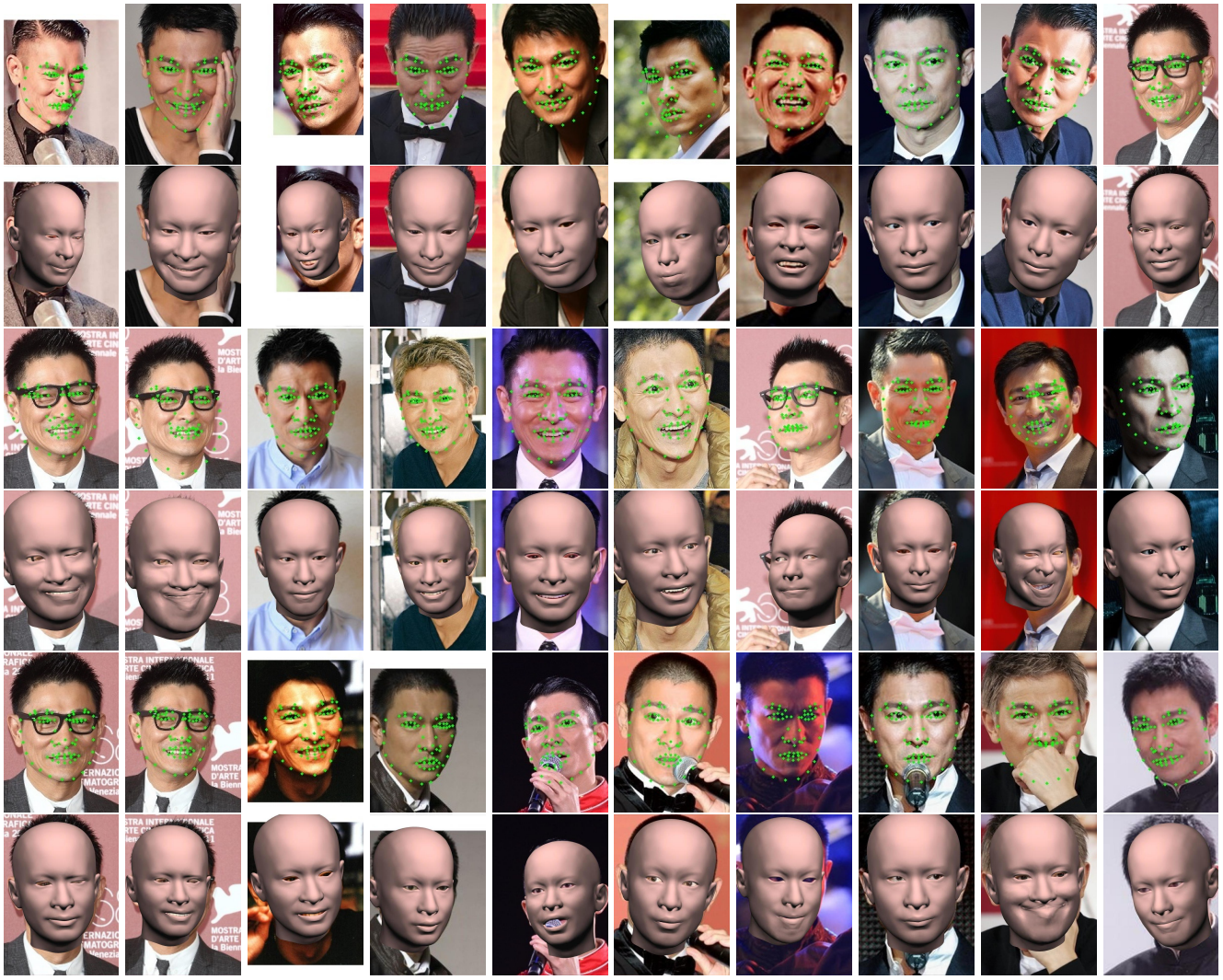


Figure 4.8: Outlier detection result of a single person. Images with occlusion by hand and glasses are successfully detected. Other outliers such as drifting contours and unreliable contour due to extreme lighting are also identified. Photographs reprinted from [AL9], [AL10], [AL11], [AL12], [AL19], [AL22], [AL25], [AL32], [AL33], [AL36], [AL38], [AL47], [AL48], [AL51], [AL58], [AL59], [AL62], [AL64], [AL65], [AL66], [AL67], [AL73], [AL74], [AL77], [AL80], [AL81], [AL86], [AL87], [AL89], [AL90] .

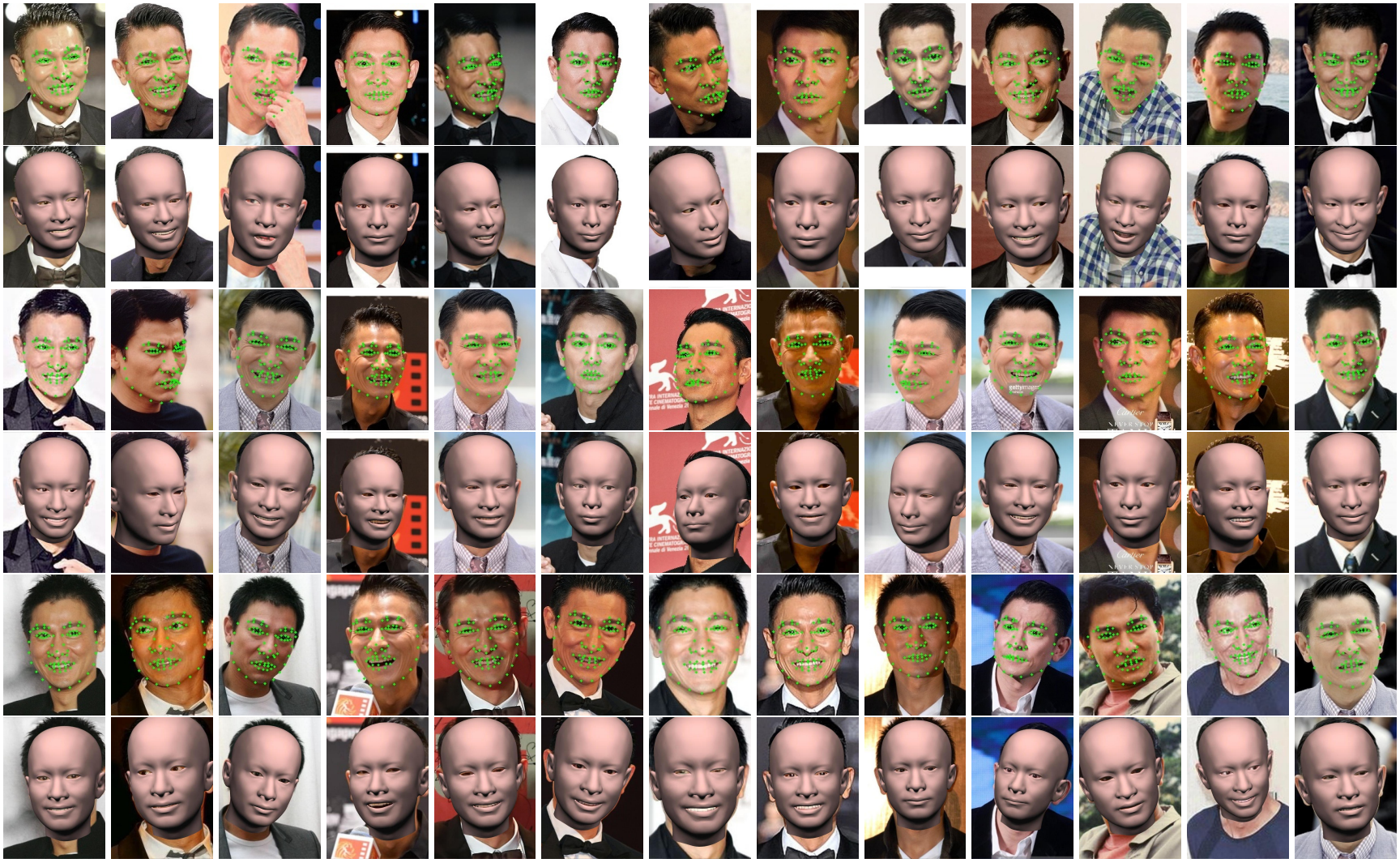


Figure 4.9: Image selected for joint reconstruction by the subset selection algorithm. The 2D feature points in these images are typically more accurate/reliable compared to other images. Photographs reprinted from [AL1], [AL2], [AL3], [AL5], [AL6], [AL7], [AL8], [AL14], [AL15], [AL17], [AL18], [AL20], [AL21], [AL23], [AL24], [AL26], [AL29], [AL31], [AL34], [AL37], [AL39], [AL41], [AL42], [AL14], [AL44], [AL45], [AL46], [AL49], [AL50], [AL52], [AL54], [AL61], [AL63], [AL68], [AL70], [AL71], [AL75], [AL78], [AL85] .

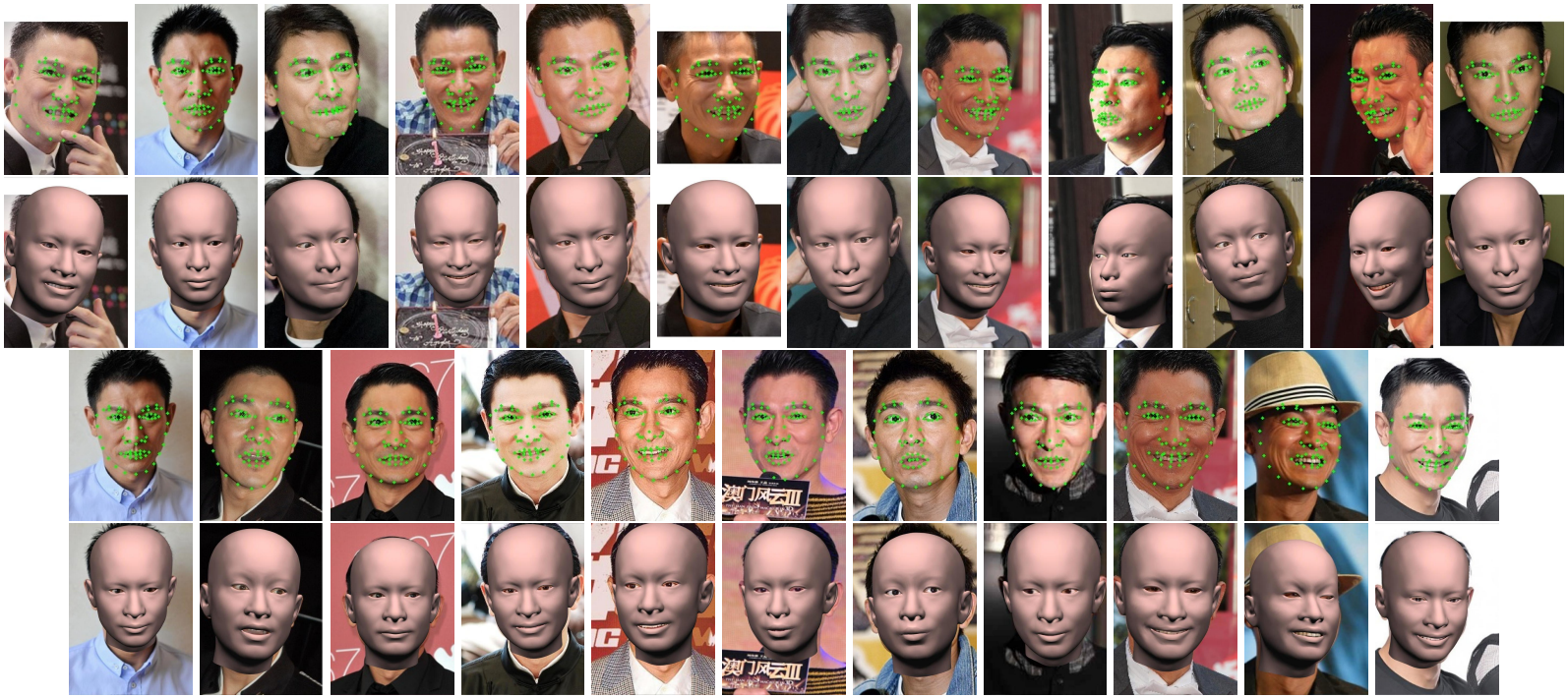


Figure 4.10: Images excluded from joint reconstruction by the subset selection algorithm. Typically, these images have less reliable facial feature points due to extreme lighting or more challenging combination of head poses and expressions, thus are ranked lower by the subset selection algorithm and excluded from joint reconstruction. Note images excluded from joint reconstruction can still be used in later steps. Photographs reprinted from [AL4], [AL13], [AL16], [AL27], [AL28], [AL30], [AL35], [AL40], [AL43], [AL53], [AL55], [AL56], [AL57], [AL60], [AL69], [AL72], [AL76], [AL79], [AL82], [AL83], [AL84], [AL88], [AL91] .

5. POINT CLOUD BASED BLENDSHAPE RETARGETING

The initial face reconstruction in the last step provides good subset images for facial rigging and a good estimate of head poses and camera parameters. However, the blendshapes recovered by the multi-linear model cannot faithfully reconstruct the true facial geometry due to limited generality of the multi-linear model and the sparse features used for fitting. In this step, we generate a set of personalized blendshapes by fitting the input images with retargeted blendshapes. To this end, we introduce the 3D point clouds of the faces as an intermediate representation and perform the optimization iteratively. We first recover high resolution point clouds from the initial reconstructions using shape from shading(SfS)[60, 2] with the help of current blendshapes, where a new point cloud region selection algorithm is used to exclude the noisy and unreliable regions in the poinft clouds. After that, we fit the point clouds with the retargeted template blendshapes with an extended example based rigging algorithm [8]. In particular, we introduce a new neutral face optimization step to reconstruct the neutral face for retargeting and then jointly optimize the blendshapes, the expression weights, and the correspondence between the point clouds and the reconstructed 3D faces. We repeat the whole process and use the resulting blendshapes as the input of the point cloud reconstruction step in the next iteration. Algorithm 3 summarizes our blendshapes generation approach.

5.1 Point Cloud Recovery and Cleanup

We follow the method in [2] to generate a pixel-level normal map from a single image, then integrate it to recover a high resolution point cloud of the face region.

5.1.1 Shape from Shading

The generation of high resolution point clouds is achieved by solving an inverse image irradiance problem[61]. Assuming the face is a Lambertian surface with unknown normals $\mathbf{n}$ and albedo

Algorithm 3 Blendshape Generation

Input: Parametric reconstruction results $\{S_k^{init}\}$ and template blendshapes $\{B_i^{temp}\}$
Output: Refined reconstructions $\{S_k^*\}$ and high quality blendshapes $\{B_i^*\}$

$\{S_k\} \leftarrow \{S_k^{init}\}, k \in K$
while not converged **do**
    $\{P_k\} \leftarrow$ RecoverPointClouds($\{S_k\}$)
    $\{B_i\} \leftarrow$ GenerateBlendshapes($\{P_k\}, \{B_i^{temp}\}$)
    $\{S_k\} \leftarrow$ PerImageFaceReconstruction($\{(I_i, \{\mathbf{p}_{ij}\})\}, \{B_i\}$)
end while
$\{S_k^*\} \leftarrow \{S_k\}, \{B_i^*\} \leftarrow \{B_i\}$
$M \leftarrow$ RecoverDetails($\{S_k^*\}, \{P_k\}, \{B_i^*\}$)
return $\{S_k^*\}$, $\{B_i^*\}$, M

$\mathbf{a}$ under unknown lighting conditions, the reflected radiance at each pixel (x, y) is given by

$$\mathbf{I}(x, y) = \mathbf{a}(x, y)R(x, y) = \mathbf{a}(x, y) \int_{\Omega} l(\omega_i \cdot \mathbf{n}(x, y))d\omega \tag{5.1}$$

where $I(x, y)$ is the color of pixel (x, y), l is a approximation function of the intensity of reflected light, while ω_i and Ω represent a solid angle of incoming light and a hemispherical integration domain centered at 3D position of (x, y), respectively.

Given an image, we can solve the above equation for the unknown surface normals, albedo, and lighting conditions simultaneously. Note that the surface normals can be equivalently expressed as the gradient of the surface depth $z(x, y)$:

$$\mathbf{n}(x, y) = \frac{1}{\sqrt{p^2 + q^2 + 1}}(p, q, -1)^T \tag{5.2}$$

where $p(x, y) = \partial z(x, y)/\partial x$ and $q(x, y) = \partial z(x, y)/\partial y$. We can therefore recover the depth information from the surface normals obtained from solving the image irradiance equation, which could by directly used for generating per-pixel point cloud of the face geometry.

It is shown that the unknown lighting condition can be approximated using spherical harmonics[62], under the assumption that attached shadows are allowed while cast shadows and inter-reflections

are ignored. In this model, The Lambertian surface only reflects low frequency components of the complicated lighting, and the approximation using spherical harmonics can be written as

$$R(x,y) = \sum_{n=0}^{N} \sum_{m=-n}^{n} l_{nm} \alpha_n Y_{nm}(x,y) \tag{5.3}$$

where l_{nm} are lighting coefficients, α_n are normalizing factors and $Y_{nm}(x,y)$ are the spherical harmonic functions at pixel (x,y). The above equation can be equivalently written in a vector form when a low order approximation is used

$$R(x,y) = \mathbf{l}^T \mathbf{Y}(\mathbf{n}(x,y)) \tag{5.4}$$

In case of a second order approximation,

$$\begin{aligned} \mathbf{Y}(\mathbf{n}) = (1, n_x, n_y, n_z, \\ n_x n_y, n_x n_z, n_y n_z, n_x^2 - n_y^2, 3n_z^2 - 1)^T \end{aligned} \tag{5.5}$$

where $\mathbf{n} = (n_x, n_y, n_z)^T$ is the surface normal.

5.1.1.1 Objective Function

The point cloud is generated by solving the image irradiance equation iteratively. Algorithm 4 summarize the algorithm for point cloud generation. We solve the image irradiance equation via non-linear optimization, with a cost function that minimize the per-pixel difference between a synthesized image and the input image:

$$\arg\min_{\mathbf{a},\mathbf{n},\mathbf{l}} \left[w_1 E_{data}(\mathbf{a},\mathbf{n},\mathbf{l}) + w_2 E_{reg}(\mathbf{a},\mathbf{n}) \right] \tag{5.6}$$

where $\mathcal{D}$ is the face region in the input image. Both w_1 and w_2 are set to 1.0 in our experiments.

Data Term This term aims at minimizing the difference between the source image and the image synthesized with the estimated normals, albedo and lighting. It is the sum of pixel difference

Algorithm 4 Point Cloud Generation

Input: Input images-feature points pairs $\{(I_i, \mathbf{p}_{ij})\}$ and per-image reconstructions $\{S_k\}$
Output: Per-image point cloud $\{P_k\}$

for all $k \in K$ **do**
    $\mathcal{R}_k \leftarrow$ FindFaceRegion$(I_k, \mathbf{p}_{ij}, S_k)$
    $\mathcal{T}_k \leftarrow$ ExtractTexture$(I_k, \mathbf{p}_{ij}, S_k)$
end for
$\mathcal{A} \leftarrow$ GenerateInitialAlbedo$(\{\mathcal{T}_k\}, \{\mathcal{R}_k\})$
for all $k \in K$ **do**
    $\mathbf{a}_k, \mathbf{n}_k \leftarrow$ GenerateAlbedoMapAndNormalMap$(S_k, \mathcal{A})$
end for
while not converged **do**
    for all $k \in K$ **do**
        $\mathbf{l}_k \leftarrow$ EstimateLightingCoefficients$(\mathbf{n}_k, \mathbf{a}_k)$
        $\mathbf{a}_k \leftarrow$ UpdateAlbedoMap$(\mathbf{n}_k, \mathbf{l}_k)$
        $\mathbf{n}_k \leftarrow$ UpdateNormalMap$(\mathbf{a}_k, \mathbf{l}_k)$
    end for
    $\mathcal{A} \leftarrow$ GenerateAlbedo$(I_k, R_k, \mathbf{n}_k, \mathbf{l}_k)$
end while
for all $k \in K$ **do**
    $P_k \leftarrow$ RecoverPointCloud$(\mathbf{n}_k, S_k)$
end for
return $\{P_k\}$

between the two images.

$$E_{data}(\mathbf{a}, \mathbf{n}, \mathbf{l}) = \sum_{(x,y)\in\mathcal{D}} \|\mathbf{I}(x, y) - \mathbf{a}(x, y)(\mathbf{l}^T \cdot \mathbf{Y}(\mathbf{n}(x, y)))\|^2 \tag{5.7}$$

Regularization Term Three regularization terms are added to the optimization problem to guide the solution of the optimization problem: an albedo regularization term, a normal regularization term and a surface integrability term. The albedo regularization term controls the deviation of the estimated albedo from the reference albedo, and the normal regularization regulates the estimated normals similarly. The integrability term is crucial in ensuring the estimated normals form an

integrable surface, such that the generated surface has C^2 smoothness[63].

$$
\begin{aligned}
&E_{reg}(\mathbf{a},\mathbf{n}) = w_{albedo}E_{albedo}(\mathbf{a}) + w_{normal}E_{normal}(\mathbf{n}) + w_{int}E_{int}(\mathbf{n}) \\
&E_{albedo}(\mathbf{a}) = \|LoG(\mathbf{a}) - LoG(\mathbf{a}_{ref})\|^2 \\
&E_{normal}(\mathbf{n}) = \|LoG(\mathbf{n}) - LoG(\mathbf{n}_{ref})\|^2 \\
&E_{int}(\mathbf{n}) = \|\frac{\partial^2 z(x,y)}{\partial y \partial x} - \frac{\partial^2 z(x,y)}{\partial x \partial y}\|^2 = \|\frac{\partial}{\partial y}\frac{n_x}{n_z} - \frac{\partial}{\partial x}\frac{n_y}{n_z}\|^2
\end{aligned}
\tag{5.8}
$$

where LoG is the Laplacian of Gaussian operator. The weights used in our experiments are $w_{albedo} = 100.0$, $w_{normal} = 0.1$ and $w_{int} = 1.0$.

5.1.1.2 Point Cloud Recovery

After solving for surface normals, we obtain a dense point cloud by recovering per-pixel depth values for the face region in the input image. We recover the depth values by solving the depth-normal equation, i.e. Eqn.(5.2). Using forward difference approximation, the gradients of depth $p(x,y)$ and $q(x,y)$ can be written as

$$
\begin{aligned}
p(x,y) &= z(x+1,y) - z(x,y) \\
q(x,y) &= z(x,y+1) - z(x,y)
\end{aligned}
\tag{5.9}
$$

This provides a series of linear constraints between the surface depth and surface normal. Since

$$
\begin{aligned}
p(x,y) &= \frac{\partial z(x,y)}{\partial x} = \frac{n_y}{n_z} \\
q(x,y) &= \frac{\partial z(x,y)}{\partial y} = \frac{n_x}{n_z}
\end{aligned}
\tag{5.10}
$$

we can formulate the recovery of depth as an optimization problem with the following cost function

$$
\underset{z(x,y),(x,y)\in\mathcal{D}}{\arg\min} \; (w_{depth}E_{depth}(z) + w_{ref}E_{ref}(z)) \tag{5.11}
$$

where

$$E_{depth}(z) = \sum_{(x,y)\in\mathcal{D}} \|z(x+1,y) - z(x,y) - \frac{n_y}{n_z}\|^2 + \|z(x,y+1) - z(x,y) - \frac{n_x}{n_z}\|^2 \tag{5.12}$$

and

$$E_{ref}(z) = \sum_{(x,y)\in\mathcal{D}} \|z(x,y) - z_{ref}(x,y)\|^2 \tag{5.13}$$

Figure 5.1 shows a sample SfS results by our system.

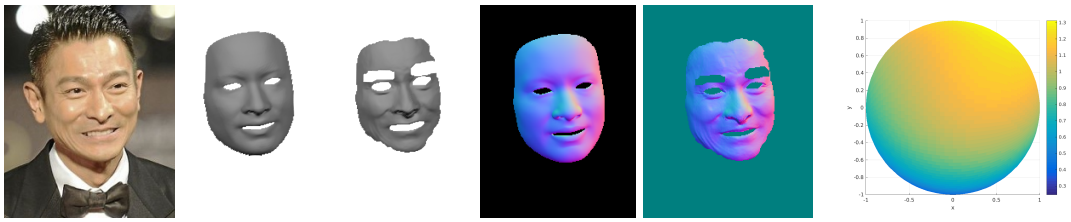


Figure 5.1: A sample SfS result. From left to right: input image, initial point cloud, optimized point cloud, initial normal map, optimized normal map, recovered lighting strength sphere. Photograph reprinted from [AL1] .

5.1.2 Albedo Generation and Skin Detection

One of the key input to the SfS algorithm is a good albedo map for the face region, which is not available initially. It is impractical to simply take the texture in the face region as the albedo because of the diverse variation of shadow and skin color caused by different lighting conditions. What's more, occlusion such as hands or microphones, as well as other noise such as hair and beard, also makes it extreme challenging to create a good albedo map. On the other hand, such occlusion could cause incorrect deformation in the recovered point clouds because of unreliable estimation of albedo and normal. We address these issues by generating the albedo map through skin detection based albedo generation.

Skin Region Detection We are mainly interested in the skin region of a face image since we need them for recovering per-pixel point clouds. Non-skin pixels usually lead to unreliable estimation of albedo and normal, resulting in noisy deformation in the recovered point cloud. We therefore seek to exclude non-skin regions as much as possible through skin region detection. We formulate this as a binary classification problem that labels each pixel as either skin pixel or non-skin pixel(occluded by hair, beard or other object). Considering the variation of skin color caused by external factors such as illumination, we use a dynamic mixture of Gaussian (GMM) to model the skin pixels. We first initialize the GMM with the pixels in the initial face region, then iteratively remove outliers based on the classification result of the GMM. We iterate the process a few times and increase the number of Gaussian components in each iteration, so that the GMM adapts better to the skin color distribution progressively. We convert the input images to YCbCr color space to exclude the effect of illumination as much as possible. The GMM model is constructed on the color channels(i.e. Cb and Cr) only. To make the detection algorithm more robust, we also perform SLIC segmentation[64] for the input image and exclude outliers segments directly. Figure 5.2 shows the result of our skin detection algorithm.

Figure 5.2: Skin detection examples. Photographs reprinted from [AL8], [OP6] and [OP8] .

Albedo Generation A good albedo map is key to successful recovery of high quality point clouds. A major challenge in albedo generation is to properly factor out illumination. We create the initial albedo in two steps to eliminate illumination effects and color difference. The first step is to iteratively remove illumination effects from the face region. We use a uniform skin color as our initial guess for the albedo, which can be easily computed by taking the average of all skin pixels

obtained from skin detection. We then use this as albedo map for the SfS algorithm and optimized individual albedo maps. The SfS results are then used to perform illumination factorization and generate new per-image albedo maps. We factorize the illumination by simply dividing the input image I by the synthesized illumination map I_l:

$$\hat{\mathbf{a}}(x, y) = I(x, y)/I_l(x, y), \forall (x, y) \in \mathcal{R} \tag{5.14}$$

where $\mathbf{a}$ is the albedo map and $\mathcal{R}$ is the face region.

We then take the average of the new per-image albedo maps and use it as the albedo map. Algorithm 5 summarizes the albedo generation process. Note that the albedo generation process is coupled with the SfS process since it requires estimating the per-image illumination maps.

Algorithm 5 Albedo Generation

Input: Input images $\{I_i\}$, and face regions $\{\mathcal{R}_i\}$, normal maps $\{\mathbf{n}_i\}$ and lighting coefficients $\{\mathbf{l}_i\}$
Output: Optimized albedo map $\mathcal{A}$

1: **for all** $k \in K$ **do**
2: $\quad \hat{\mathbf{a}}_k \leftarrow$ FactorizeIllumination$(I_k, \mathcal{R}_k, \mathbf{n}_k, \mathbf{l}_k)$
3: **end for**
4: $\mathcal{A} \leftarrow$ ComputeMeanTexture$(\{\hat{\mathbf{a}}_k\}), k \in K$
5: **return** $\mathcal{A}$

Figure 5.3 shows the effect of this step. After eliminating the illumination in the initial albedo map, we perform a color transfer[65] operation to match the albedo's color with the input image(Figure 5.3).

5.1.3 Point Cloud Region Selection

Since the shape-from-shading method assumes the face to be a smooth Lambertian surface illuminated with low frequent lighting, hard shadows, specular highlights will make the algorithm fail. We thus also develop an algorithm for detecting and removing these outlier regions. Our method determines whether to include a certain point in the point clouds based on two metrics: the

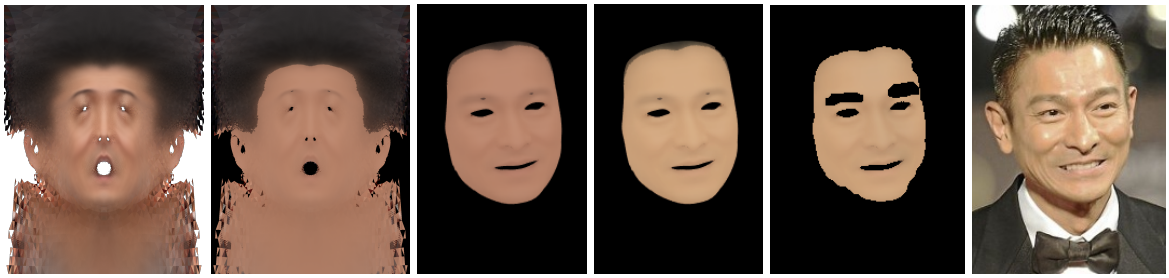


Figure 5.3: Initial albedo generation. From top-left to bottom-right: mean texture, illumination-excluded albedo, projected albedo, albedo after color transfer, final albedo after SfS optimization and input image. Photograph reprinted from [AL1].

SfS image fitting error E_{sfs} and the deformation amount d compared to the reference shape.

$$\texttt{inc}(x,y) = (E_{sfs}(x,y) < E_{max}) \;\; \texttt{\&\&} \;\; (d(x,y) < d_{max}) \tag{5.15}$$

where (x, y) is a pixel in the face region $\mathcal{R}$. We experimentally set $E_{max} = 0.02$ and $d_{max} = 0.05$. Figure 5.4 shows an example of the selection result.

5.2 Point Clouds Based Blendshape Retargeting

Given the point cloud recovered in the last step, we follow the method in [8] to generate the personalized blendshapes from the template blendshapes via deformation transfer,

$$\underset{B,w_i}{\operatorname{argmin}} \; w_d E_d(B, w_i) + w_t E_t(B) + w_m E_m(B) \tag{5.16}$$

where

$$E_d(B, w_i) = \sum_{i=1}^{N} \|P_i - B \otimes w_i)\|^2$$

is the data term that minimizes the difference between the 3D facial expressions reconstructed by blendshape and the point clouds. The retargeting term

$$E_t(B) = \sum_{j=0}^{K} \|B_j - D(B_0, B_0^t, B_j^t)\|^2$$

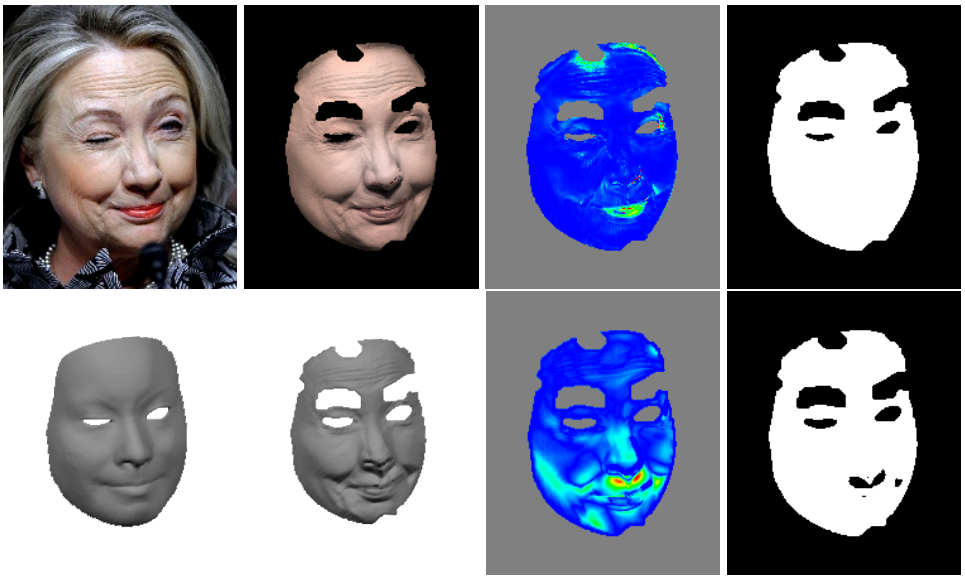


Figure 5.4: Point cloud region selection. From top-left to bottom-right: input image, synthesized image in SfS, SfS image fitting error(using jet color scheme), skin mask, large-scale deformation reconstruction, point cloud by SfS, deformation amount map(red for large deformation and blue for small deformation), and selected point cloud region. Note that some regions are excluded already by our skin detection module. Photograph reprinted from [HC5].

enforces the expression similarity between the resulting blendshapes and the template blendshapes, where K is the number of blendshape bases and $D(B_0, B_0^t, B_j^t)$ transfers the deformation between template blendshape B_j^t and B_0^t to the neutral face B_0 [66]. The third term E_m minimizes the difference between the optimized blendshape and the blendshapes B^m from the previous iteration,

$$E_m(B) = \sum_{j=0}^{K} w_j \|B_j - B_j^m\|^2 \tag{5.17}$$

where B_j^m is the j-th blendshape basis generated in the previous iteration.

In the first iteration, B_j^m is generated by the multi-linear model

$$B_j^m = \mathcal{C} \times_{id} w_{id} \times_{exp} U_{exp}\delta_j.$$

Here $\mathcal{C}$ is a reduced multi-linear model tensor and U_{exp} is the matrix that maps a unit vector in the FACS space to expression weights in the reduced multi-linear space [35]. Figure 5.5 shows an

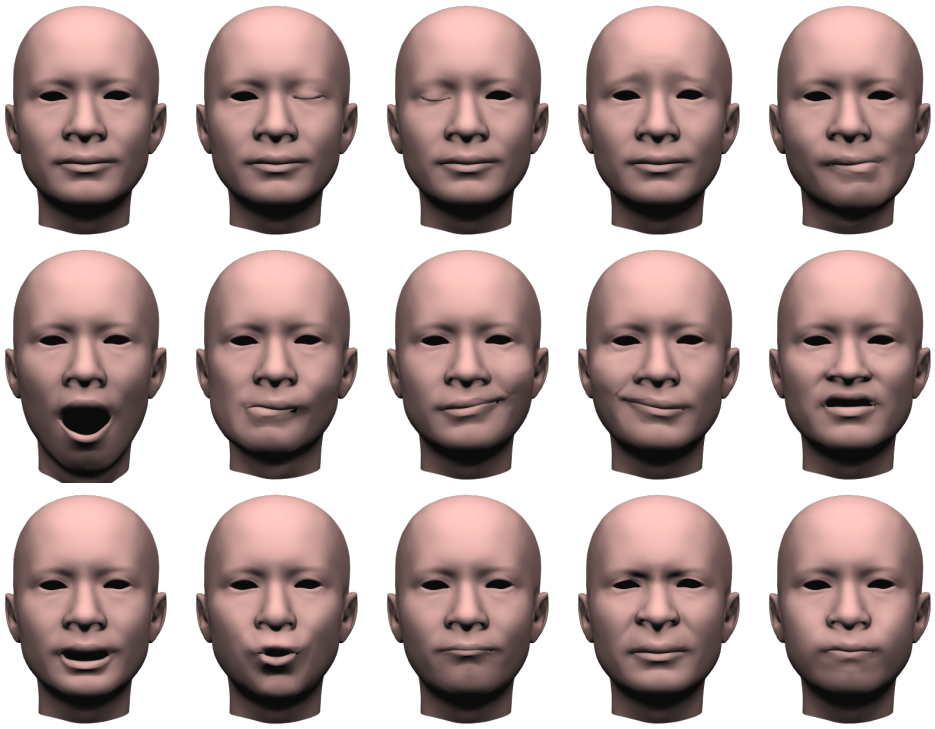


Figure 5.5: Initial blendshapes generated by multi-linear model.

example of the initial blendshapes generated with the multi-linear model.

In Equation 5.17, w_d, w_t, and w_m are weights of the three terms respectively. Note that different from [8], we do not have the neutral face of the subject B_0 for deformation transfer. We also lack the correspondence between the input point clouds for optimization. To this end, we first build the correspondence of the input point clouds by deforming the aligned 3D facial expressions reconstructed from the current blendshapes and then create a personalized neutral faces B_0. After that, we can optimize the blendshapes and expression weights as in [8]. The whole process is iterated until convergence. Algorithm 6 summarizes this point cloud based blendshape retargeting process.

5.2.1 Correspondence Generation

To build the correspondence between point clouds to the 3D blendshapes, we deform the 3D face shape reconstructed by current blendshape basis $M_i = B \otimes w_i$ to fit the recovered point clouds via Laplacian deformation [67]. Here the correspondence between the point clouds P and the face

Algorithm 6 Blendshapes Retargeting

Input: Current blendshapes$\{B_j\}$, template blendshapes $\{B_j^t\}$, point clouds $\{P_i\}$ and per-image expression weights $\{w_{exp,i}\}$.
Output: Refined blendshapes $\{B_j^*\}$

while not converged **do**
    $\{M_i'\} \leftarrow$ GenerateCorrespondence$(\{\{B_j\}, \{w_{exp,i}\}, \{P_i\}\})$
    $B_0 \leftarrow$ OptimizeNeutralFace$(\{B_j^t, \{B_j\}, \{M_i'\}, \{w_{exp,i}\}\})$
    $\{B_j\} \leftarrow$ OptimizeBlendshapes$(\{B_j^t\}, \{B_j\}, \{M_i'\}, \{w_{exp,i}\})$
end while
$\{B_j^*\} \leftarrow \{B_j\}$
return $\{B_j^*\}$

mesh $B \otimes w_i$ is established with an iterative closest point strategy similar to Li's method[68]. Algoirthm. 7 describes the process for correspondence generation.

Algorithm 7 Correspondence Generation

Input: Point cloud P, blendshapes B, expression weights w_i
Output: Deformed mesh M_i'

$M_i' \leftarrow B \otimes w_i$
while not converged **do**
    $C \leftarrow$ FindPointToTriangleCorrespondance(P, M_i')
    $M_i' \leftarrow$ DeformShapeWithPoints(M_i', C, P)
end while
return M_i'

After this step, we obtain the deformed mesh M_i' that has the same topology and is aligned with the underlying blendshape. Therefore, we define $E_d = \sum_{i=1,N} \|M_i' - B \otimes w_i)\|^2$ in the following optimizations.

5.2.2 Neutral Face Optimization

Given the M_i' obtained in the last step, we construct the neutral face B_0 using optimization in Equation 5.16, where we fix the blendshape weights and set $w_d = 1.0, w_t = 0.5$ and $w_m = 0.01$. Also, the weights in Equation 5.17 are set to $w_0 = 1$ and $w_j = 0$ $(\forall 0 < j < K)$. Note that for

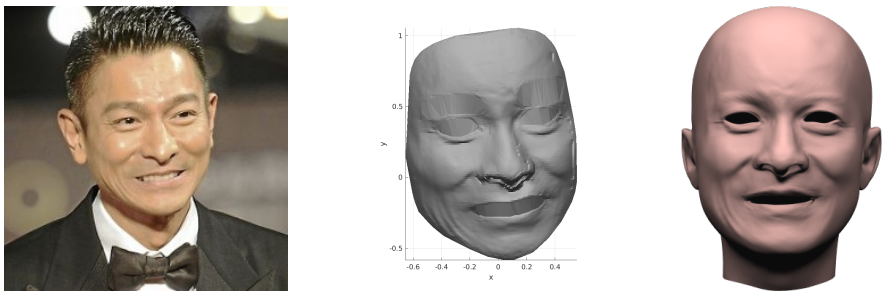


Figure 5.6: Correspondence generation result. From left to right: input image, recovered point cloud and deformed mesh. Photograph reprinted from [AL1].

neutral face generation, we allow aggressive optimization for all blendshapes, including the neutral face. This aggressive optimization is likely to cause over-fitting in other blendshapes since we set $w_j = 0$ $(\forall i > 0)$. We keep only the neutral face and discard the other blendshapes obtained in this optimization.

5.2.3 Blendshape and Expression Weights Optimization

In this step, we fix the neutral face B_0 generated in the last step and optimize all other blendshapes and weights according to Equation 5.16. For this, we set $w_d = 1.0, w_t = 0.5$ and $w_m = 0.25$ for all three terms in Equation 5.16 and $w_j = 1.0$ for all terms in Equation 5.17. We follow the method in [8] to solve blendshapes and their weights iteratively, where the blendshapes B and the expression weights w_i are updated by solving two linear systems.

5.3 Iterative Blendshapes Refinement

With the refined blendshapes, we can repeat the large-scale facial reconstruction process to correct the misalignment between the point cloud and the input images, as well as the head pose error. We therefore repeat the entire blendshapes generation process starting from per-image face reconstruction, using the generated blendshapes instead of multi-linear models for large-scale face deformation reconstruction. The improved blendshapes helps the estimation of head poses and thus further reduces the misalignment between the recovered point clouds and target blendshapes, which in turn improves the quality of the final blendshapes(Figure 5.7). In our experiments, we only need to three iterations to achieve the best blendshape quality. Figure 5.8 shows how our

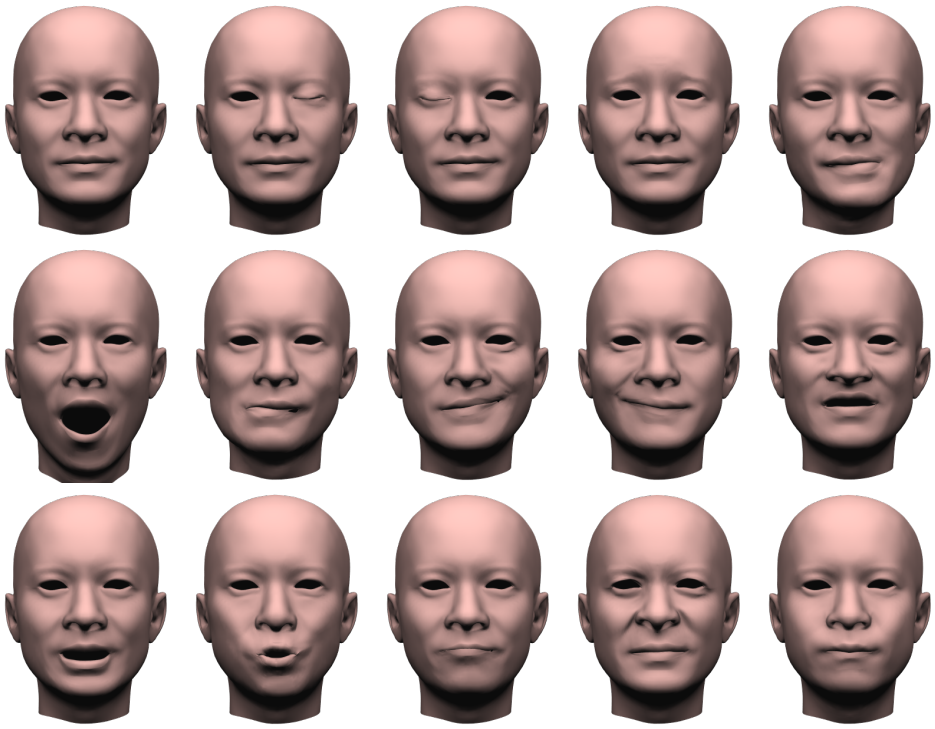


Figure 5.7: The same set of blendshapes in Figure 5.5 after 1 iteration of optimization.

iterative process improves the blendshapes' quality.

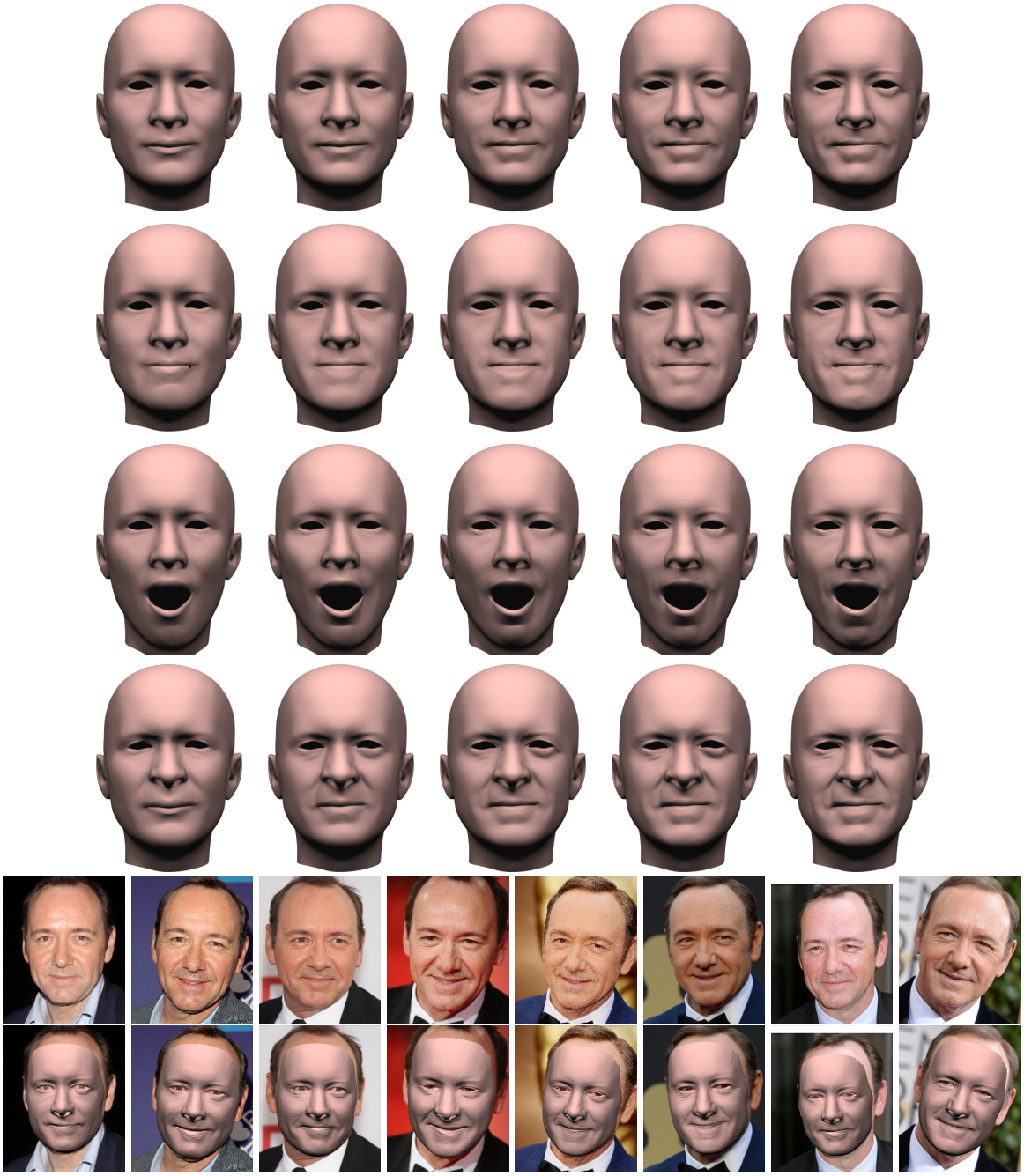


Figure 5.8: Iterative blendshapes generation. 4 blendshapes out of 47 are included here. The first row shows the neutral face shape. In each row of blendshapes, from left to right are: initial blendshapes from multi-linear models, blendshapes after the first, second and third iterations, and final output blendshapes. The last two rows show 8 out of 93 images used for generating these blendshapes, and the reconstructed face using the generated blendshapes. Photographs reprinted from [KS1], [KS3], [KS4], [KS9], [KS13], [KS15], [KS17], [KS19] .

6. FINE-SCALE DETAIL RECOVERY

6.1 Motivation

Fine-scale facial features are very important in capturing subtle yet unique characteristics of a person's expressions. Without fine-scale detail, the perceived facial expression can be significantly different. Figure. 6.1 shows a typical example when the missing fine-scale detail leads to misperception of the facial expression. The subtle contempt expression is well captured by the facial reconstruction with fine detail, while the facial reconstruction without fine detail illustrates a more neutral expression.

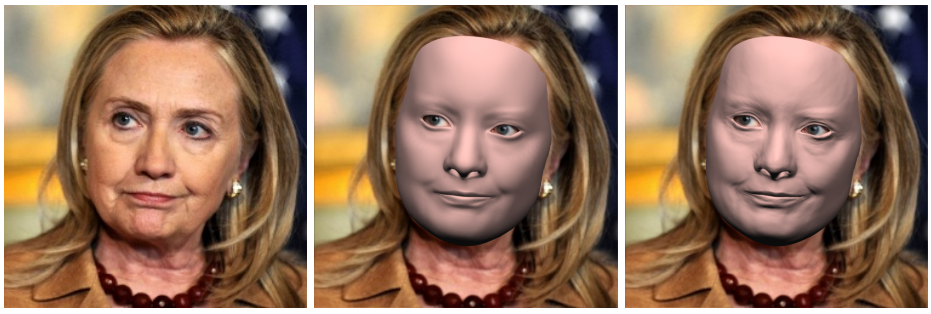


Figure 6.1: Importance of fine-scale detail. From left to right: input image, facial reconstruction without fine-scale detail, facial reconstruction with fine-scale detail. The expression is relatively neutral without fine-scale detail. As a contrast, the subtle contempt expression exhibited in the input image is captured in the facial reconstruction with fine-scale detail. Photograph reprinted from [HC9].

The blendshapes generated by our system capture personalized facial geometry well with the point cloud based blendshapes retargeting method. Given unlimited computation resources, our method can capture fine-scale facial detail by using high resolution meshes, which can model fine-scale detail in the point clouds recovered from input images. However, generating a large number of blendshapes directly with high resolution meshes is usually time consuming, since the complexity of the blendshape generation algorithm is proportional to the complexity of the meshes

being used for blendshapes.

To strike a balance between realism of the final blendshapes and the time cost for generating the blendshapes, we use medium resolution meshes for the generated blendshapes and integrate fine-scale details into the blendshapes by modeling the fine-scale facial features as per-vertex corrective normal maps. The corrective normal maps are applied to the 3D face shapes reconstructed by the blendshapes for computing per-vertex normal directions.

6.2 Fine-Scale Detail Regression

The point clouds recovered in the previous stage contain rich information about fine scale facial features observable from the input images; however, such fine-scale detail is lost because of the use of medium resolution meshes when generating the blendshapes. Since the blendshapes already capture personalized large-scale deformation, we can model the fine-scale detail as an additive component to the blendshapes. To this end, we can use either displacement maps or corrective normal maps for modeling fine-scale detail. Displacement maps model the per-vertex displacment in the mesh such that the actual vertex position $\hat{v}$ of a vertex v in the mesh is given by the sum of v and the displacement vector δv:

$$\hat{v} = v + \delta v$$

Corrective normal maps, in contrast, model the per-vertex corrective normal vector in the mesh such that the actual normal vector $\hat{n}$ of a vertex is given by the sum of the normal vector n and the corrective normal vector δn:

$$\hat{n} = n + \delta n$$

Different from previous work that utilizes displacement map to represent fine-scale details, we choose to use normal map in our system because normal maps are less intrusive compared to displacement maps. Displacement map based method does not work well in our case due to the very challenging input our system handles. The extreme variations of head poses, expressions and illumination cause strong ambiguity in fine-scale facial details, and a displacement map based approach is likely explain the variations in appearance with more drastic geometry changes. This

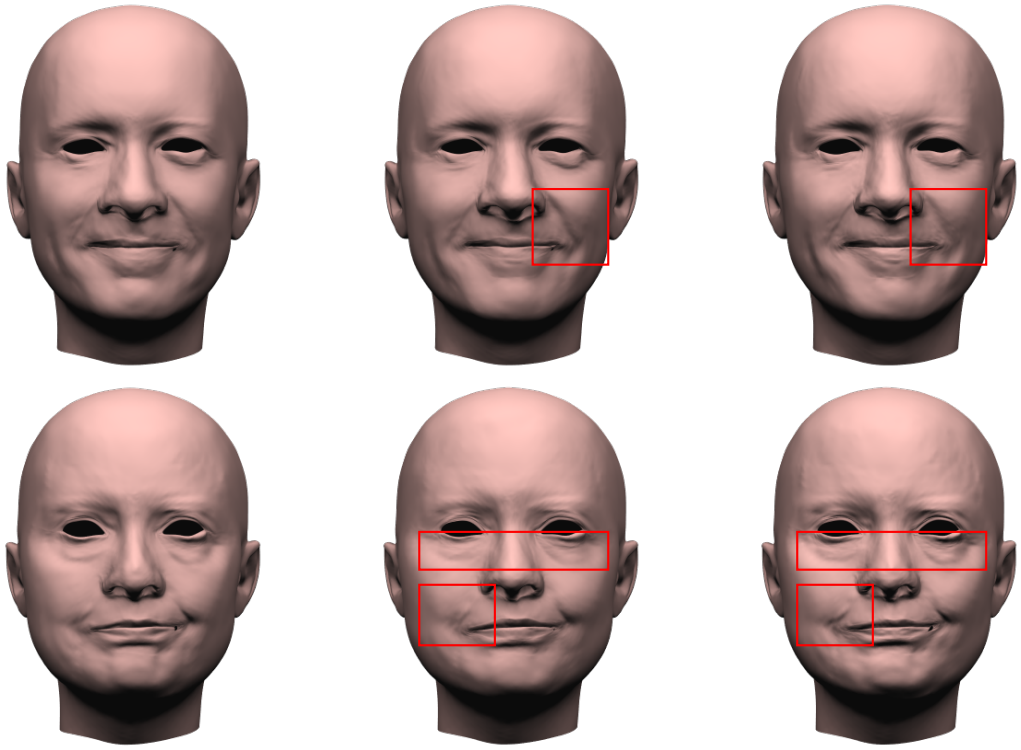


Figure 6.2: Comparison of displacement map based and corrective normal maps based fine detail recovery. From left to right: point cloud visualization, reconstruction with fine-scale detail recovered with displacement maps, reconstruction with fine-scale detail recovered with corrective normal maps. Note displacement maps based method easiliy leads to over-deformation or under-deformation, while corrective normal maps based method generates more faithful fine-scale detail.

directly leads to undesired artifacts in the final blendshapes. On the other hand, normal map based approach models the fine-scale details in the gradient domain. A small change of the normal vector direction could result in large appearance change. The statistical model of fine-scale details constructed in the normal vector domain thus is less likely to generate extreme numeric values compared to displacement map based method. As a result, the normal map based approach generates a more stable model for fine-scale facial detail. Figure. 6.2 illustrates the advantages of using corrective normal maps over displacement maps.

Since blendshape model is a linear model, we need to model fine-scale detail with some linear model in order to maintain the linearity of the blendshape model. Specifically, since we synthesize

a specific facial expression with blendshapes by linear interpolation:

$$F_i = w_i \otimes B$$

the fine-scale detail represented by the corrective normal map δn_i should be synthesized in the same manner

$$\delta n_i = w_i \otimes \mathcal{M}(\Delta N)$$

where $\mathcal{M}(\Delta N)$ is the linear model for the corrective normal maps.

In the previous stages, we recover a high resolution point cloud P_i for each input image I_i and estimate the expression weights w_i that fits P_i best by with the generated blendshapes B. The point clouds P_i and the expression weights $\{w_i\}$ provide rich information about the fine-scale detail as well as the correspondence between them. We can construct a linear model for the corrective normal maps using a data driven approach by extracting the relationship between expression weights w_i and the corrective normal map dn_i.

We therefore formulate the fine-scale detail recovery as a regression problem with the goal of predicting corrective normal map δn_i from a given expression weights w_i. A naïve approach to solve this problem is to construct a direct mapping $S_{\text{naïve}}$ from the input data. We first compute the difference between the normal directions in point cloud P_i and the 3D face shapes reconstructed by the generated blendshapes S_i as corrective normal map δn_i. Note that all point clouds are normalized to frontal view in order to exclude the effect of different head poses. Then we stack all δn_i together as a single matrix ΔN and stack all expression weights w_i together as an expression weights matrix W. The naïve mapping from expression weights to corrective normal vector can then be computed by solving a linear least squares problem

$$\arg\min_{S_{\text{naïve}}} \|\Delta N - W S_{\text{naïve}}\|$$

which has an exact solution of

$$S_{\text{naïve}} = (W^T W)^{-1} W^T \Delta N$$

However, this naïve approach leads to very noisy result in the recovered fine detail because the corrective normal vectors computed from the point clouds is usually noisy and may contain conflicting information. On the other hand, this naïve model is also redundant because there could be many similar expressions in the input image. To address these issues, we construct a compact a compact representation of corrective normal maps $\{\delta n_i\}$ with PCA, then compute a linear transformation S that maps an expression weights vector w_i to this compact representation of corrective normal maps. The use of PCA model here serves two important purposes:

1. the PCA model creates a compressed representation for the corrective normals, resulting in a more compact model;

2. the PCA model allows filtering the noisy part in the corrective normal maps, leading to a more stable model.

To construct the fine-scale detail recovery model, we first construct a PCA model for the corrective normals:

$$\Delta N_{3|V|\times N} \approx PC^T_{N_{pc}\times 3|V|} L_{N_{pc}\times N}$$

where $L_{N_{pc}\times N}$ is a low-dimensional representation of the corrective normals, N_{pc} is the number of principal components, and $PC_{N_{pc}\times 3|V|}$ is the matrix of all principal components in the PCA model. Here, $|V|$ is the number of vertices in the meshes and N is the number of input point clouds. We experimentally choose the value of N_{pc} to capture 95% of variance in the corrective normals ΔN, which varies from 8 to 17 depending on the person being modeled.

A mapping matrix $S_{N_{pc}\times K}$ to transform expression weights w_i to a normal difference vector is

then derived using ridge regression:

$$\arg\min_{S} \|SW - L\|_2 + \alpha\|S\|_2 \tag{6.1}$$

where $W_{K\times N}$ is the matrix formed by stacking all $\{w_i\}$ together. α in Equation. 6.1 is used to control regularization strength in the ridge regression. We experimentally set α to 0.0005, which produces satisfatory results for most cases.

We can then recover the corrective normal vector for a new 3D expression with weights $w_{i'}$ as

$$\delta n_{i'} = (Sw_{i'})^T \cdot PC$$

and add $\delta n_{i'}$ to the normals of the 3D face shapes synthesized by $w_{i'}$ to obtain the final per-vertex detailed normals. Figure. 6.3 shows examples fine-scale detail recovery results.

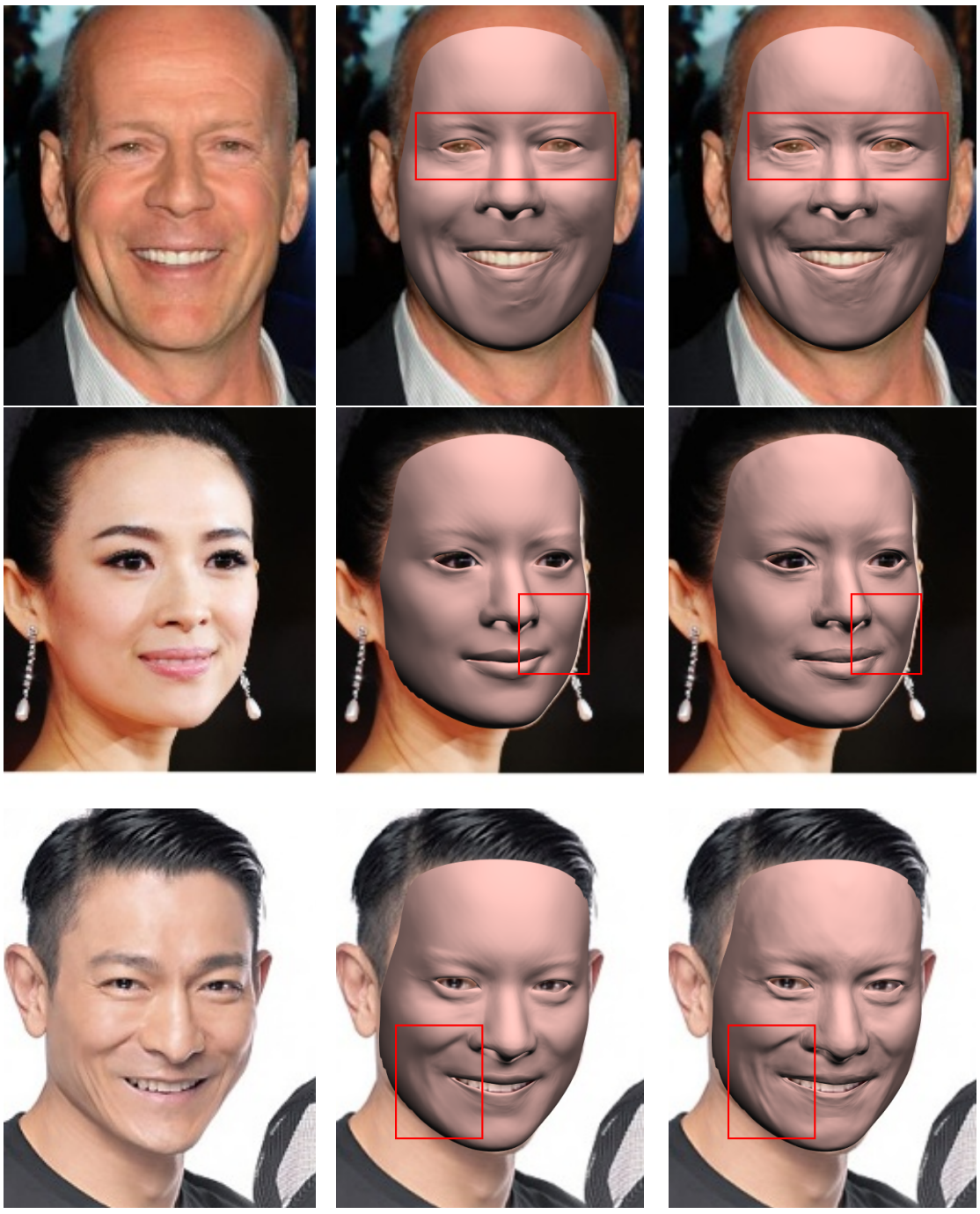


Figure 6.3: Examples of fine-scale detail recovery. From left to right: input image, reconstruction with generated blendshapes and reconstruction with generated blendshapes plus fine-scale detail. Photographs reprinted from [BW12], [ZZ5] and [AL91].

7. RESULTS AND EVALUATION

7.1 System Performance

We have implemented our system on a workstation with an Intel i7-6770K CPU, an Nvidia GTX 1060 GPU and 32GB memory. Both large-scale reconstruction and the blendshapes generation modules are written in C++ with OpenMP acceleration, while the point cloud recovery module is written in MATLAB. For input consisting of 100 images, our system takes about 3000 seconds to generate the final blendshapes: 833 seconds for large-scale reconstruction, 1272 seconds for point clouds recovery, 777 seconds for blendshapes generation, and 34 seconds for detail generation. Note that the most time consuming module, i.e. point cloud recovery module, could be significantly accelerated with a C++ implementation.

7.2 Method Validation

7.2.1 Algorithm Convergence

We first test our system to see if the solution converges as expected. This is a crucial step before doing further evaluation of the system, since the input images our system handles are extremely challenging and could easily cause instability of the system. This experiment verifies that our system behaves as expected given the very challenging input images in the wild.

Synthetic Data We evaluate our blendshapes generation algorithm on synthetic data obtain from FaceWarehouse[35]. We use create a multilinear model with 145 out of 150 person, and use the remaining 5 persons for testing. We synthesize 20 training shapes for each person by linearly interpolating their blendshapes and randomly generating head poses. We firstly synthesize 2D feature points by simulating the projection process of related vertices. Facial reconstructions and initial blendshapes are then created using our progressive reconstruction method. After that, we create high resolution point clouds from the synthesized training shapes by rendering them to depth buffers. Finally we generate blendshapes with the initial blendshapes and the synthesized point clouds. We evaluate the results by two metrics:

- Blendshapes reconstruction error E_B: we directly compute the difference between the generated blendshapes and the source blendshapes used for synthesizing the test data. Since all the input data are perfect, this gives us the lower bound of blendshape generation error in our system.

- Point cloud fitting error E_P: we also use the generated blendshapes to reconstruct facial shapes with the synthesized 2D feature points. This gives us the lower bound of point cloud fitting error with the blendshapes generated by our system.

As shown in Figure 7.1, our system successfully reduces error in terms of both metrics on the synthetic data sets, especially point cloud fitting error. Note that the improvements on the blendshapes are partially affected by the adjustments in expression weights, since blendshapes do not form an orthogonal shape space and the point clouds can be explained by both expression weights and blendshapes basis.

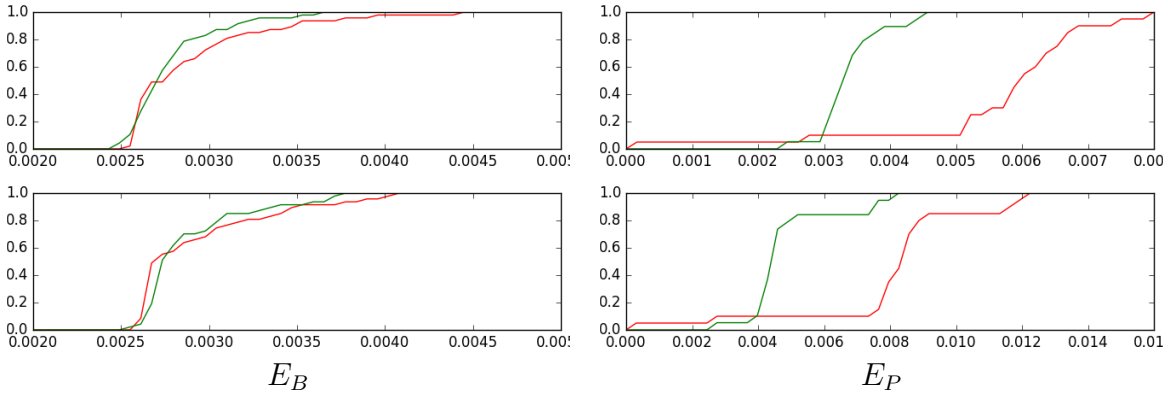


Figure 7.1: Blendshapes reconstruction error(left) and point cloud fitting error(right) on synthetic data. Each row is from an individual test data set.

Real World Data We also evaluate our system on point cloud fitting error with real world data. Since we don't have ground truth in this case, we evaluate our system by looking at the difference of fitting error between the initial blendshapes and final blendshapes. Figure 7.3 shows 2 examples

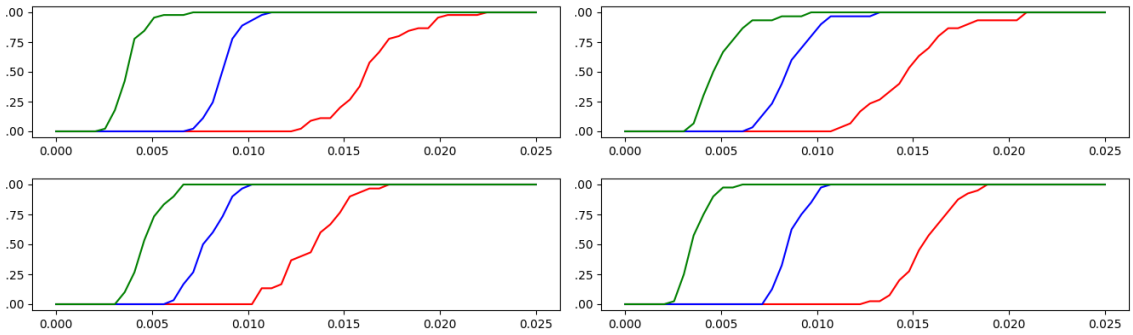


Figure 7.2: Point cloud fitting error of the multilinear models fitting(red curve) and our method(green curve) on real world data. From top left to bottom right are Andy Lau(93 images), George Clooney(104), Hillary Clinton(94), and Kevin Spacey(114).

of the point cloud fitting error with real world data. In both examples, the fitting error decreases significantly with our generated blendshapes. We also show the cumulative distribution curve of fitting error in Figure 7.2(b) for 4 test data sets we experimented with. In all cases, the fitting error with our generated blendshapes are much lower compared to the initial blendshapes.

7.2.2 Quantitative Evaluation

We use the 3D models and images in the FaceWarehouse dataset to quantitatively evaluate the accuracy of the blendshapes generated by our method. This dataset includes images, sparse feature points, and 3D ground truth meshes of 150 identities. For each identity, the dataset consists of 20 photos with different expressions and their corresponding 2D feature points and ground truth mesh. We randomly select 4 subjects as the test data and use the rest of the subjects for training the multi-linear model. For each test subject, we use 20 images in the dataset as the input of our system for recovering the personalized blendshapes.

We measure the result quality with the fitting error E_P, which measures the mesh distance between the ground truth 3D facial expressions and the ones reconstructed from the recovered personalized blendshapes. Figure 7.4 shows the ground truth 3D expressions of two subjects, and results reconstructed by our personalized blendshapes and the error maps between them. For comparison, we also show the multi-linear model reconstruction results and their error maps. Note

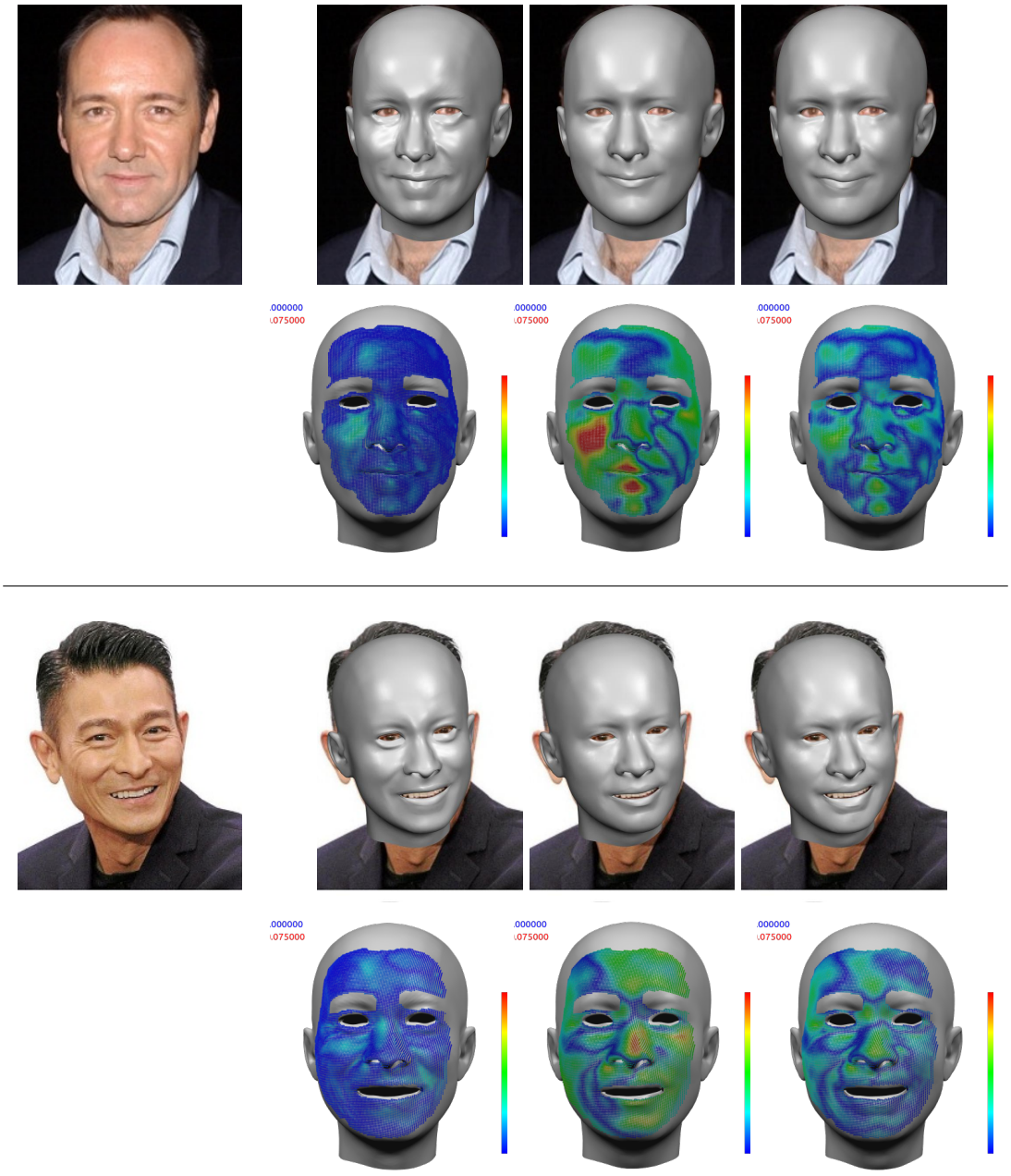


Figure 7.3: Examples of point cloud fitting error. For each section, from top left to bottom right: input image, reconstruction with the generated blendshapes, reconstruction with multi-linear model using sparse constraints, reconstruction with multi-linear model using dense constraints, point cloud fitting error for reconstruction with the generated blendshapes, reconstruction with multi-linear model using sparse constraints and reconstruction with multi-linear model using dense constraints. The error bar has a range of [0, 7.5mm]. Photographs reprinted from [KS1] and [AL2].

that the personalized blendshapes faithfully reconstruct the 3D face geometry of the subjects and generate much better reconstruction results and lower errors than the multi-linear model. We also show the cumulative distribution curve of the blendshape error in Figure 7.5 for all four subjects. Again, the errors of our results are lower than the results generated by multi-linear models.

Figure 7.4: Examples of point cloud fitting error compared to ground truth. From left to right: input image, reconstruction with base multi-linear models, reconstruction with the generated blendshapes, point cloud fitting error with base multi-linear models, point cloud fitting error with the generated blendshapes. The error bar has a range of [0, 10mm]. The input images are reprinted from the FaceWarehouse dataset[35].

7.2.3 Contributions of Each Component

To evaluate the contributions of our new components to the final results, we construct 5 different configurations of our system as in Table 7.1, each of which disables different components. We tested all 5 systems with the ground truth data of a randomly select person in the FaceWarehouse database. We compare the results generated by different configurations and measure blendshapes error.

As shown in Figure 7.6, the novel components introduced in our system play a critical role

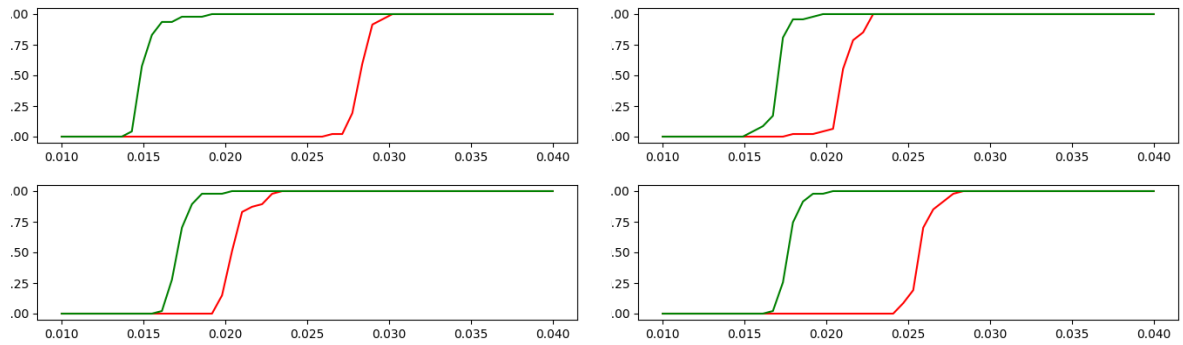


Figure 7.5: Blendshapes error of the multi-linear models fitting(red curve), and our method(green curve) on FaceWarehouse data. The unit for x-axis is dm.

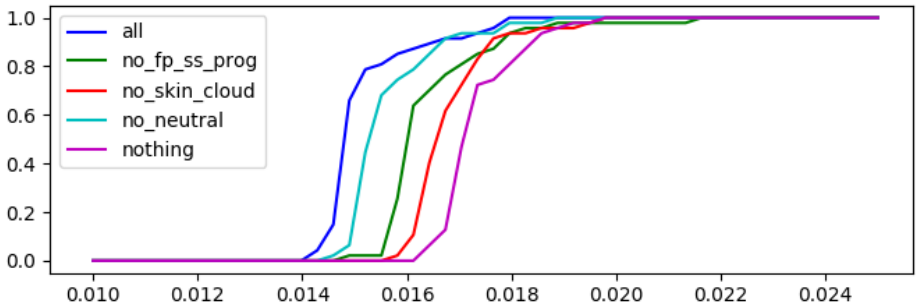


Figure 7.6: Cumulative distribution of blendshapes error using different configurations. Our system performs best when all key components are enabled, and produces inferior results when any of the key components are disabled.

for generating high-quality results. In particular, the skin detection and point cloud region selection makes major contributions to the result quality. The feature point failure detection, subset selection, as well as the progressive reconstruction provide better large-scale facial reconstruction and improve the alignment between the blendshapes and the point clouds. Finally, neutral face optimization further improves the quality by extracting geometric features common to the neutral shape and reduces the risk of over-fitting in individual blendshapes.

7.2.4 Robustness to Number of Images

To test the robustness of our method for different numbers of input images, we tested our system using the image data from FaceWarehouse. We randomly select a person from the database

Configuration	FP	SS	PROG	Skin	Cloud	Neutral
all	✓	✓	✓	✓	✓	✓
no_fp_ss_prog				✓	✓	✓
no_skin_cloud	✓	✓	✓			✓
no_neutral	✓	✓	✓	✓	✓	
nothing						

Table 7.1: Configurations for testing contributions of each component. FP: facial feature points failure detection; SS: subset selection; PROG: progressive reconstruction; Skin: skin detection; Cloud: point cloud region selection; Neutral: neutral shape optimization.

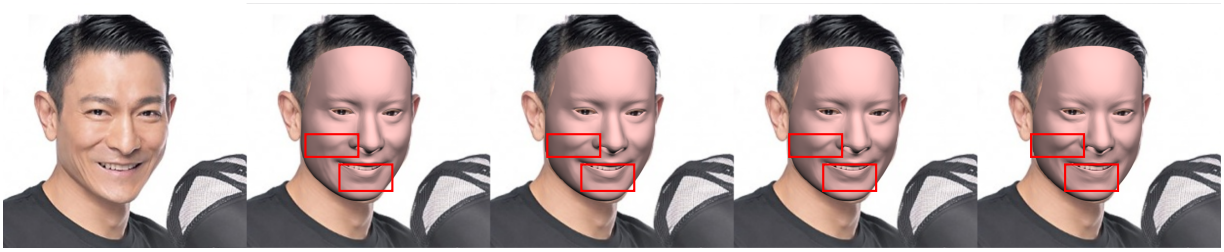


Figure 7.7: Comparisons of the face reconstruction results using blendshapes generated from different number of input images. From left to right: input image, reconstruction from blendshapes generated using 25, 50, 75 and 100 images. Using more input images helps improve facial details such as the nasolabial wrinkle. Photograph reprinted from [AL91].

and use 5, 10 and 20 images of that person to generate blendshapes, and measure the error between the generated blendshapes and the ground truth. As shown in Figure 7.8, as the number of images increases, the quality of the resulting blendshapes consistently increases. We also tested our system using real world images, for which we found using around 100 images yields satisfactory results(Figure 7.7).

7.3 Results for Real World Data

We also tested our system on 10 image sets and 2 videos downloaded from the Internet. The number of images in each set ranges from 25 to 125. Figure 7.9 demonstrates some input images and the 3D facial shapes reconstructed with the result blendshapes. Compared to the multi-linear reconstruction results, the 3D face shapes reconstructed by our blendshapes are visually consistent with the input images (the second column in Figure 7.9). In contrast, the reconstructions by multi-

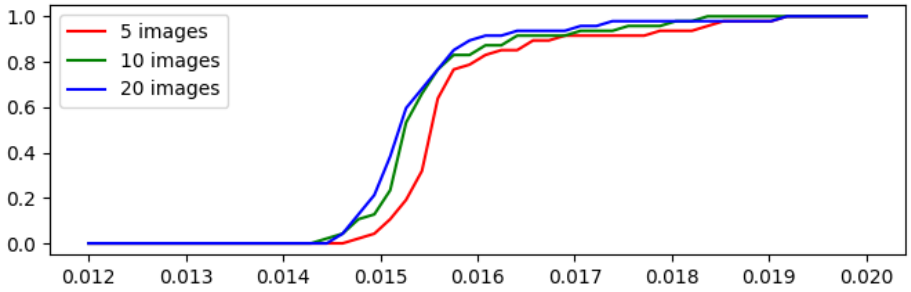


Figure 7.8: Cumulative distribution of blendshapes error using 5, 10 and 20 input images. The unit of the horizontal axis is dm.

linear models only roughly fit to the input images and lacks personalized face shape features and expression details. Please refer to Chapter.8 for more examples.

7.4 Discussions and Limitations

7.4.1 Error Accumulation.

The outlier detection modules used in each step of our algorithm helps us to remove poor quality results and avoid error propagation in our pipeline. Moreover, the iterative algorithms for personalized blendshape reconstruction also eliminate the error accumulation in our pipeline. As discussed in the last section, our method is robust to the number of input images and generates good results.

7.4.2 Shape-from-Shading vs. Photometric Stereo

Our method is based on a shape-from-shading method and solves the albedo, lighting and depth from input images. Similar to previous methods, our method also leverages the low rank space of the face shape and albedo to solve this ill-posed problem. A photometric stereo method as in [4, 3] is helpful to solve this ill-posed problem with more inputs. However, it suffers from the same the bas-relief ambiguity as the shape-from shading method. In our solution, we use the same identity prior to solve the blendshapes from the input images and resolve the bas-relief ambiguity. Compared to photometric stereo based methods, our method significantly reduces the input requirements for the facial rigging method while preserving the result quality.

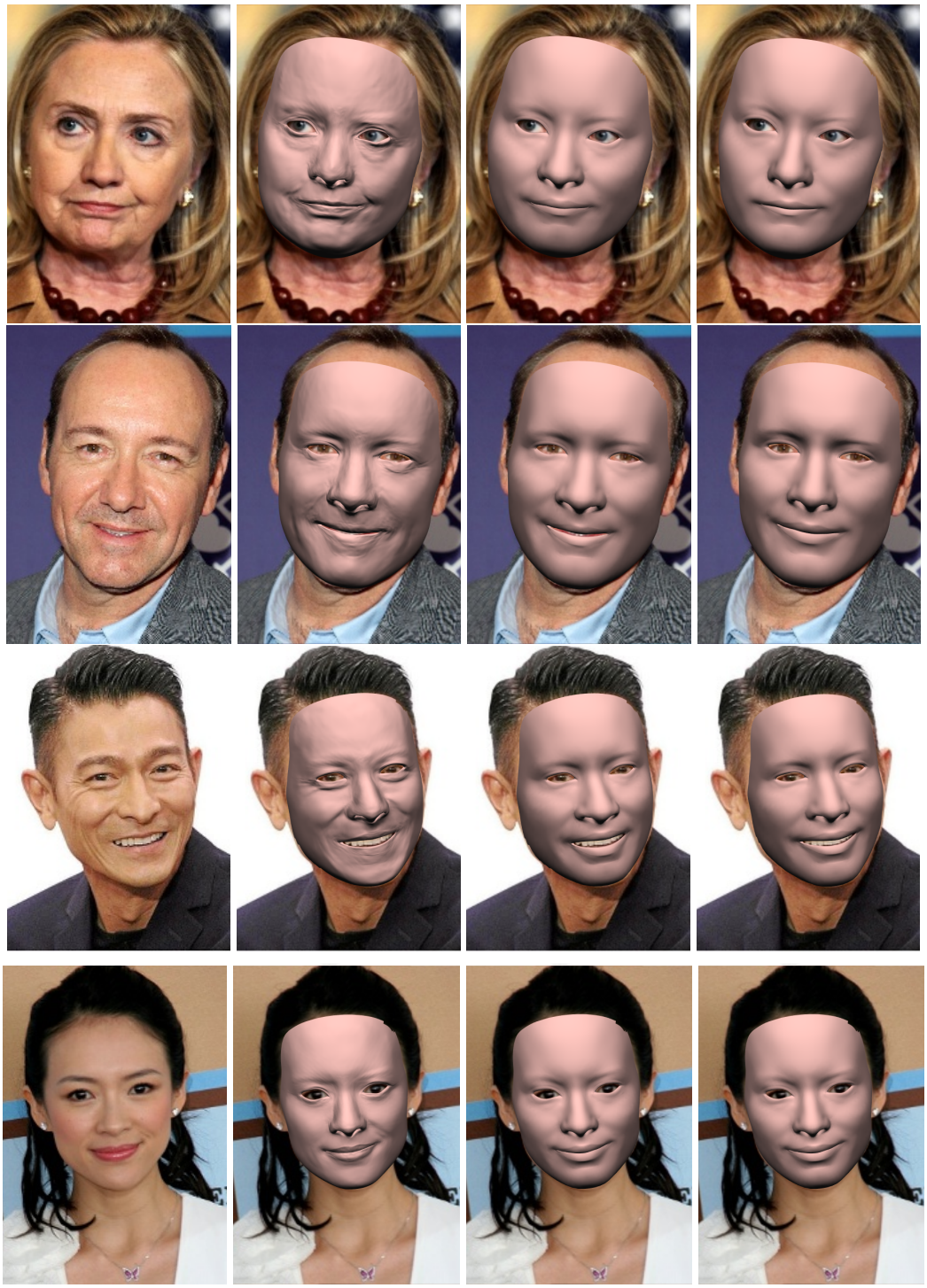


Figure 7.9: Facial reconstructions using our generated blendshapes (column 2), multi-linear model reconstruction from point clouds(column 3), and multi-linear reconstruction from sparse 2D features (column 4). The first column shows the input images. Chapter 8 includes more results. Photographs reprinted from [HC9], [KS3], [AL2], [ZZ12] .

7.4.3 Comparisons with Existing Methods

Because our system is the first fully automatic system for generating high-quality blendshapes directly from images in the wild, we could not compare our method with previous ones with the same inputs. On the other hand, our method can be applied for solving the video-based facial rigging problem and facial reconstruction problem from the input images. To demonstrate the generality of our method for other applications, we compare our method with the methods in [1] and [2] for video based facial rigging; we also compare our method against [3] and [4] for facial reconstruction. Results show that our method achieves better or comparable results compared to previous methods. Please refer to Section B in the supplemental materials for details.

In this section, we compare our system against several related methods to illustrate the effectiveness of our system. Our system is the first-ever system capable of generating high-quality blendshapes from images in the wild. Previous systems such as [1] and [2] also produce personalized blendshapes; however, their system are designed to process video data and could not handle unconstrained images. With the higher degree of complexity in head poses, facial expression and illumination commonly seen in real world images, our system aims to solve a more challenging version of facial rigging problem. Nonetheless, our system can also be used to solve the video-based facial rigging problem. To illustrate the effectiveness of our system, we compare the results by our system with theirs in this section.

7.4.3.1 Compare with Garrido et al.

We compare our method against state-of-the-art video-based facial rigging method[1]. Their system is designed specifically to handle video input data, while our system is capable of processing unconstrained images including frames taken from a video clip. Our system therefore can also be used to generate personalized blendshapes from video input. We test our system with a video clip used in [1]. The test video (Obama's speech) is obtained from the public domain. The resolution is 1920×1080 and the total number of frames is . We sampled 1 frame per second from the first 100 seconds of video for generating blendshapes, then performed per-frame facial reconstruc-

tion with the generated blendshapes. As shown in Figure 7.10, our method generates results with more facial details than [1]. Our blendshapes not only capture correct large-scale deformation, but also some medium scale features such as wrinkles.

Figure 7.10: Comparing the results by our system and Garrido et al.[1]. From left to right: input frame, our method, [1] with medium scale. Note the face shape generated by our system correctly captures the lip shape and nasolabial wrinkles while [1] does not. Portraits of Barack Obama are reprinted from a video downloaded from https://youtu.be/d-VaUaTF3_k.

7.4.3.2 Compare with Shi et al.

We also compare our method against the video-based face capture system by [2]. Their system captures facial performance with fine details from video input, which utilizes a set of personalized blendshapes reconstructed using multi-linear model. The personalized blendshapes in their system suffers from the limited generalization ability, and they alleviate this issue by refining the reconstructions synthesized with their blendshapes using per-pixel shading cues. Note that their system

is not able to handle unstructured input images, and their system does not produce high-quality blendshapes like ours either. However, the blendshapes generated by our system could be directly used for facial reconstruction, therefore we compare the per-frame face reconstructions by their system and ours. We test our system using one of the video in [2]. The test video (Bryan's interview) is obtained from the public domain. The resolution is 1280×720 and the total number of frame is 703. We sampled 100 frames evenly from the video for generating blendshapes, then perform per-frame facial reconstruction with the generated blendshapes. As show in Figure 7.11, our method generates blendshapes with more faithful facial geometry compared to [2]. Although our method does not produce fine-scale details like [2], we can further extend our method to include such details using displacement maps or similar techniques.

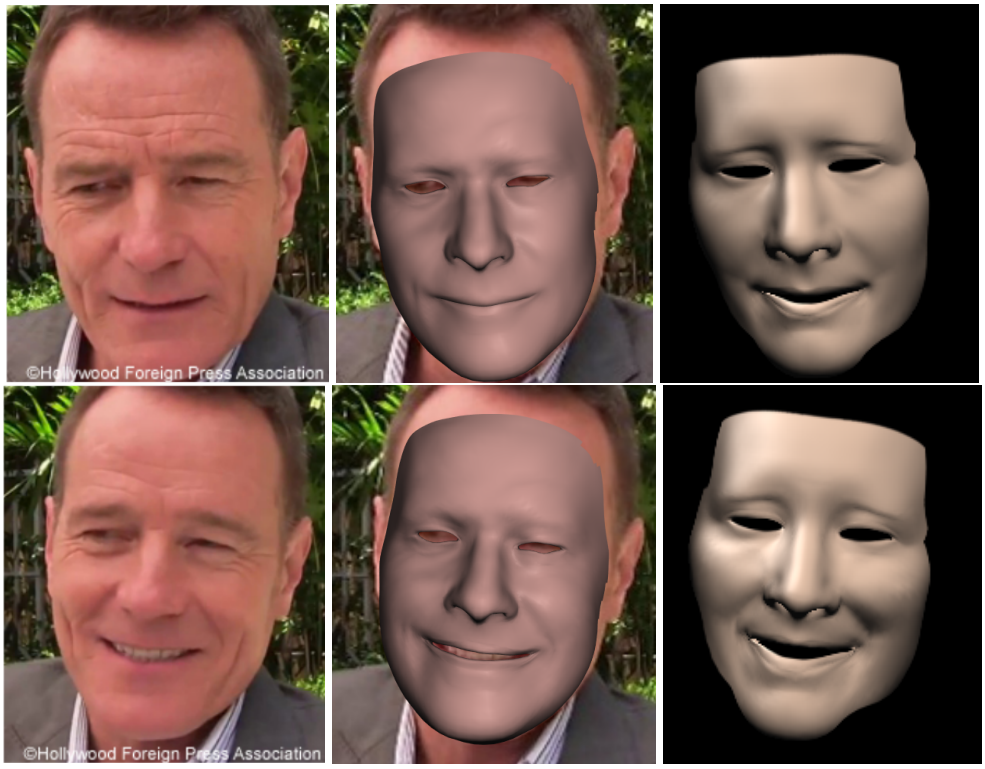


Figure 7.11: Comparing the results by our system (column 2) and Shi et al.[2](last column). Our facial reconstructions shows more facial details compared to [2]. Portraits of Bryan Cranston are reprinted from a video downloaded from http://students.cse.tamu.edu/fuhaoshi/FacefromVideo/index.htm.

7.4.3.3 Compare with Roth et al. and Kemelmacher-Shilizerman et al.

Our system generates a set of high-quality blendshapes, while [3] and [4] focus on creating facial reconstructions. Nonetheless, the personalized blendshapes generated by our system can be used to perform facial reconstruction for new images of the same person. Note that their methods could not be used to create blendshapes like ours.

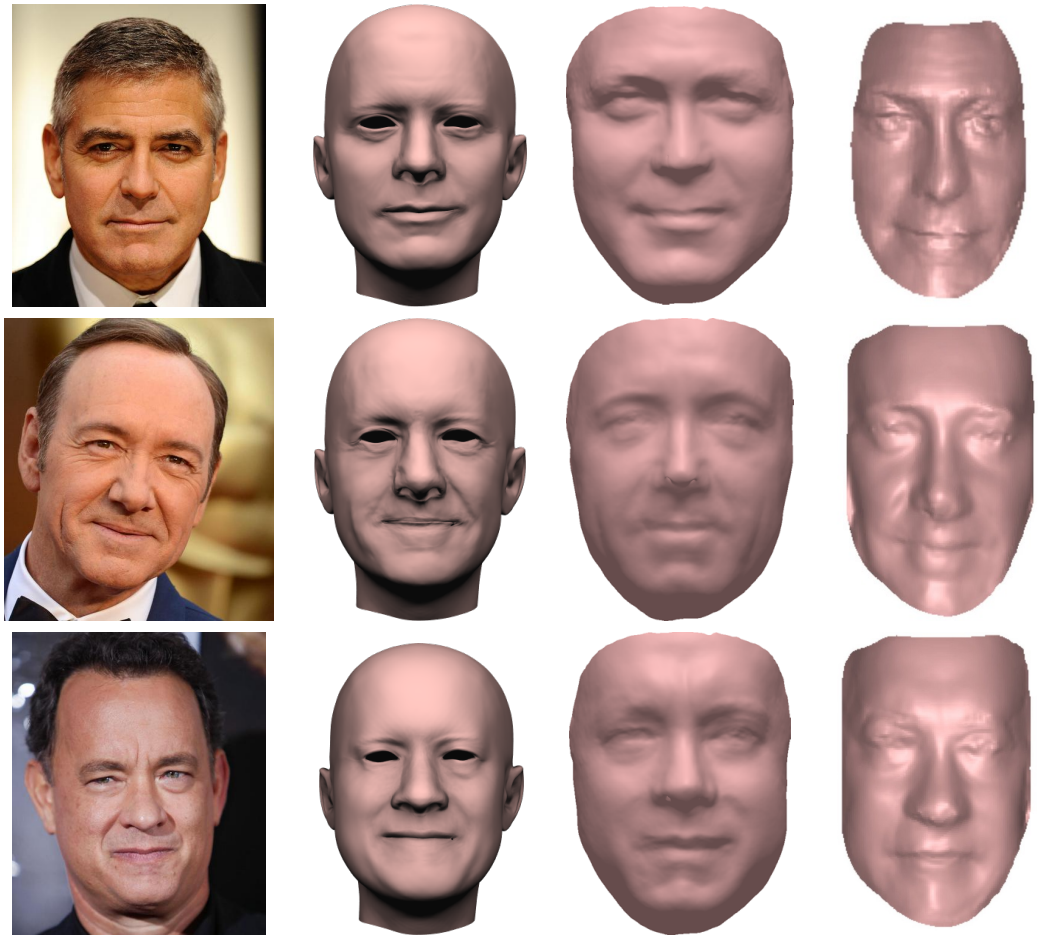


Figure 7.12: Comparing the results by our system (column 2), Roth et al.[3] (column 3) and Kemelmacher-Shilizerman et al.[4](last column). Note that both [3] and [4] perform shape-from-shading for the input image to produce facial reconstruction with fine details. Our system reconstructs 3D faces through linear interpolation of the personalized blendshapes and obtains comparable results. Photographs reprinted from [OP8], [KS18] and [OP9].

We compare the facial reconstruction results by our generated blendshapes against [3] and [4] here. Facial reconstruction with our generated blendshapes can be achieved by solving an optimization problem similar to multi-linear reconstruction, and in this case we only need to estimate head poses and expression weights. As shown in Figure 7.12, our system produces comparable results, with the additional benefit of generating a set of reusable blendshapes at the same time. Note that we can simply use the generated blendshapes to perform facial reconstruction for any new input images without the expensive computation cost required in the other two system. In contrast, [3] and [4] only generate a static facial reconstruction, which could not be directly used for animation purpose. In addition, their methods require expensive shape-from-shading computation for each new images to obtain good reconstruction result.

8. MORE BLENDSHAPE AND RECONSTRUCTION RESULTS

We include 8 persons' blendshapes generated by our system in this section (Figure 8.1 - 8.8). These results clearly show that our system generates highly personalized blendshapes with faithful facial geometry. The blendshapes also succesfully captures some medium and fine-scale facial detail. For example, the charateristic wrinkles on both sides the cheeks and the nasolabial wrinkles in Figure 8.1 are well captured by the generated blendshapes. Similarly, the charateristic wrinkle on both sides of the cheeks and the under-eye bags are captured by the blendshapes in Figure 8.2. The nasolabial wrinkles and the bump below the lower lip are modeled by the generated blendshapes in Figure 8.5, and the rising cheek bones are captured by the blendshapes in Figure 8.7. The selection of different people across different races and different also shows that our system generalize well for a wide range of inputs.

We additionally include facial reconstructions of 14 people to demonstrate how the facial reconstructions improve as a result of the improved blendshapes quality by our system(Figure 8.9 and Figure 8.10). As shown in Figure 8.9 and Figure 8.10, the initial facial reconstruction obtained by multi-linear model (the second column) only captures the large-scale facial deformation to certain degree. In most cases, such reconstructions do not reflect the true facial geometry of the person being modeled. With the optimized blendshapes generated by our system, we are able to create facial reconstructions(the last column) that are much more faithful to the input images.

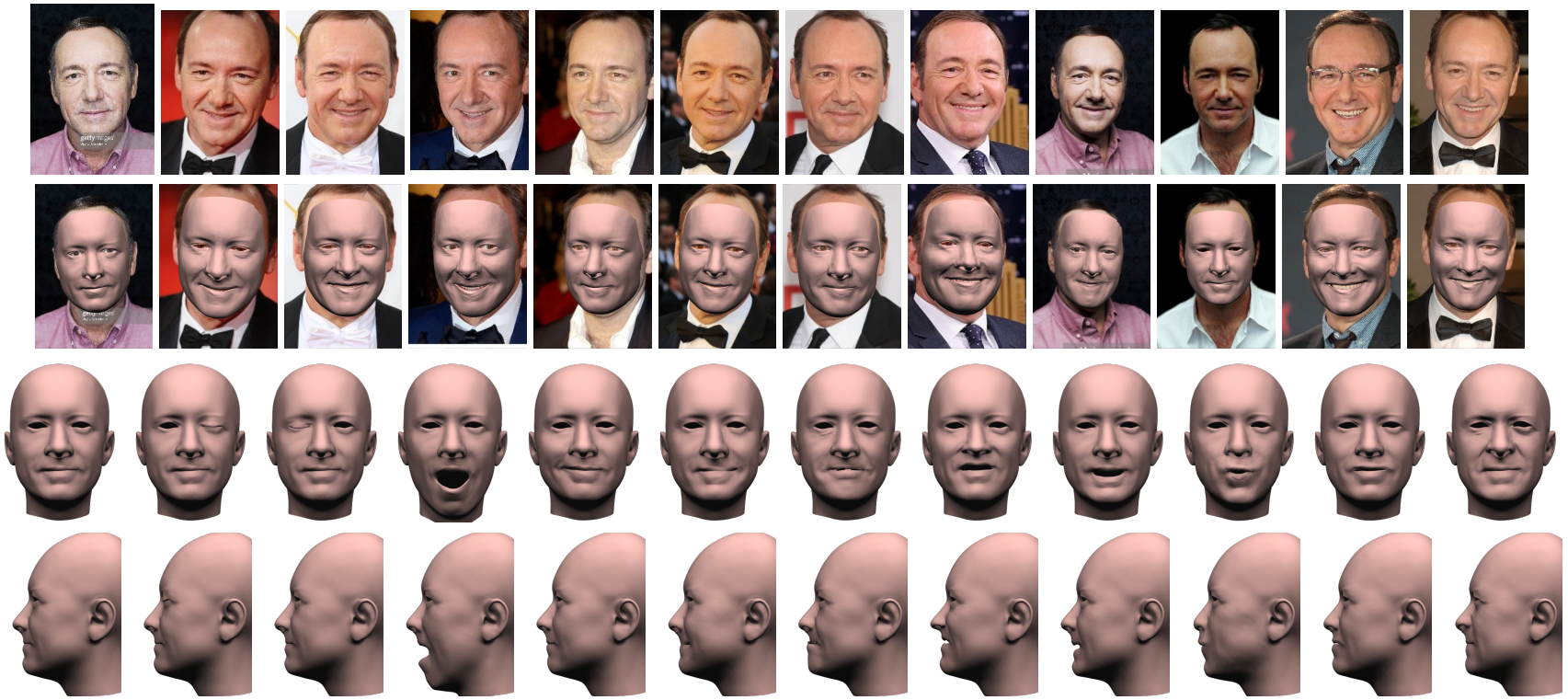


Figure 8.1: Automatic blendshapes generation - subject 1. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [KS7], [KS9], [KS16], [KS2], [KS14], [KS10], [KS4], [KS5], [KS6], [KS8], [KS11], [KS12].

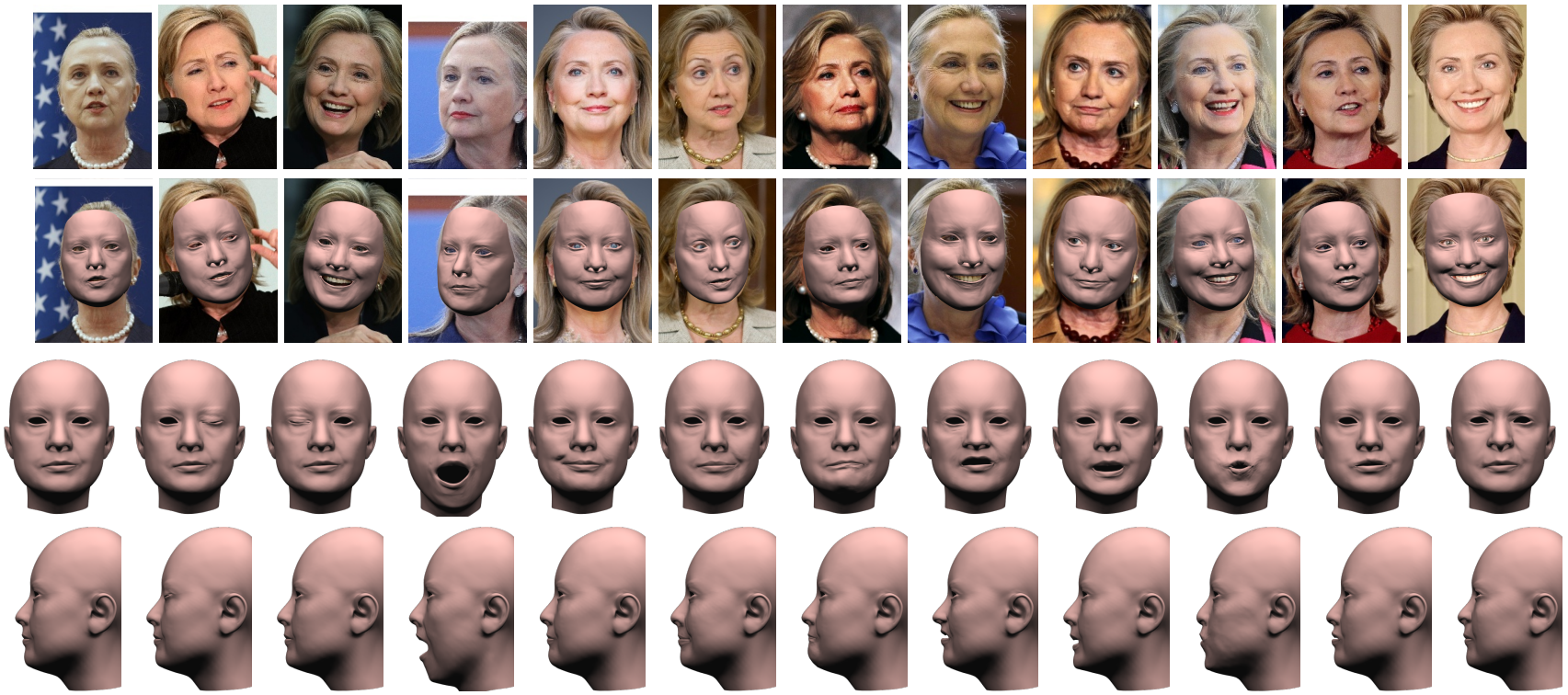


Figure 8.2: Automatic blendshapes generation - subject 2. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [HC2], [HC3], [HC15], [HC4], [HC6], [HC7], [HC10], [HC16], [HC9], [HC11], [HC13], [HC14].

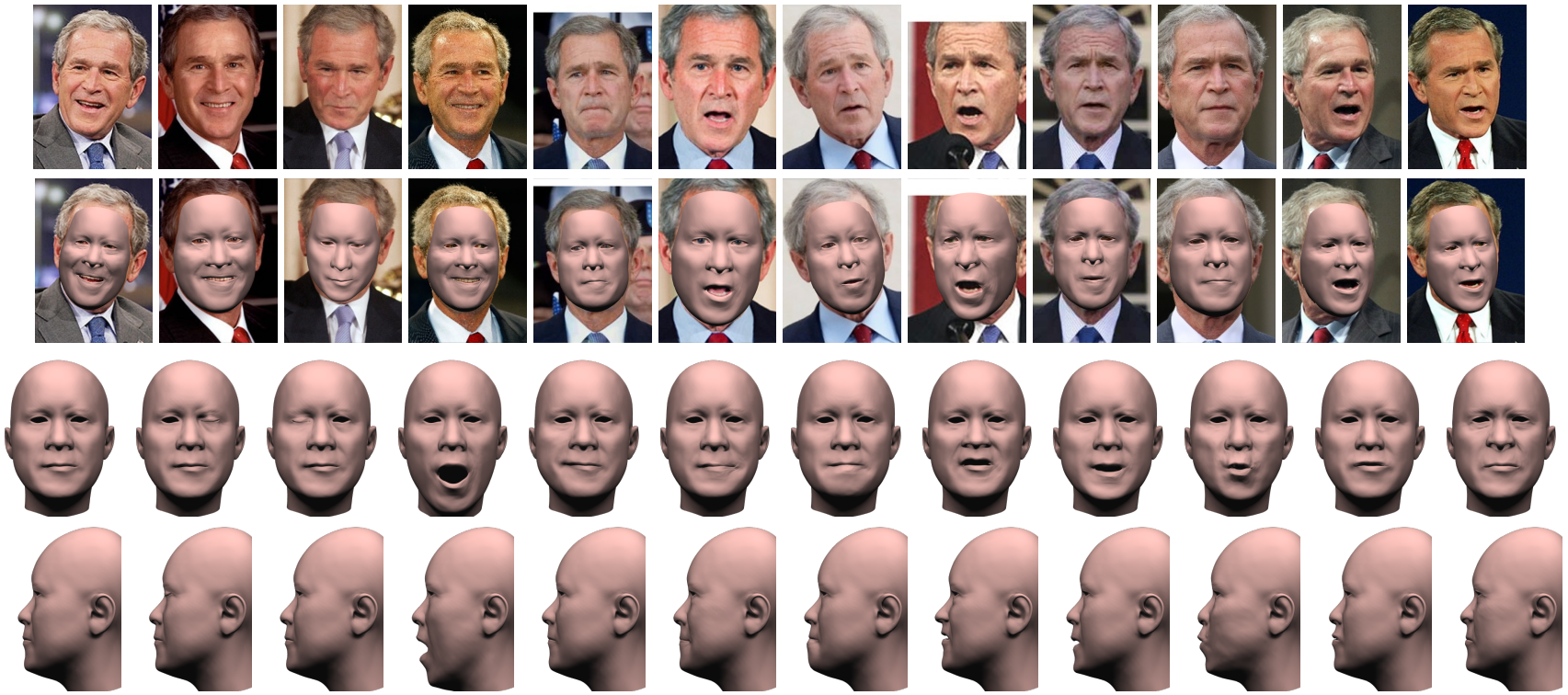


Figure 8.3: Automatic blendshapes generation - subject 3. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [GB13], [GB3], [GB4], [GB5], [GB14], [GB6], [GB7], [GB8], [GB9], [GB10], [GB11], [GB12].

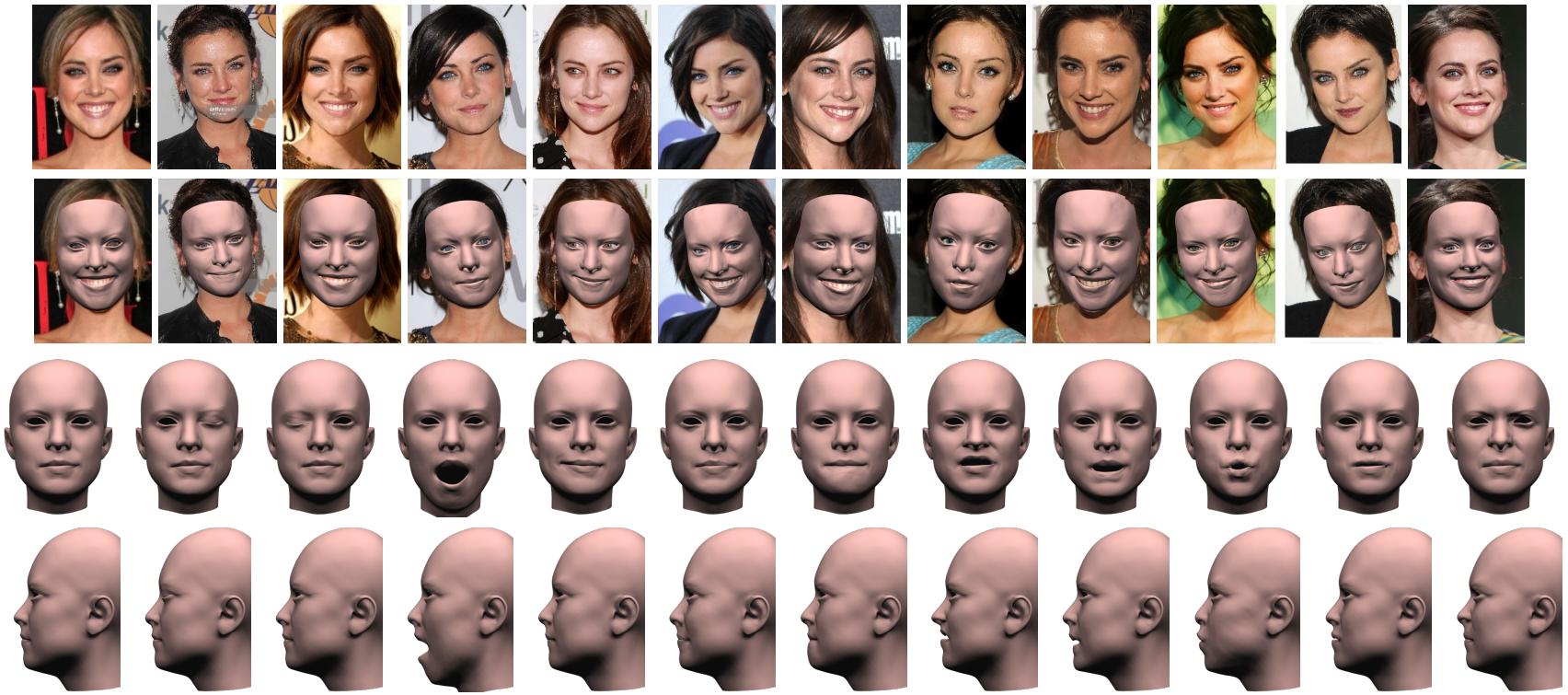


Figure 8.4: Automatic blendshapes generation - subject 4. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [JS2], [JS1], [JS7], [JS3], [JS9], [JS4], [JS11], [JS12], [JS8], [JS6], [JS5], [JS10].

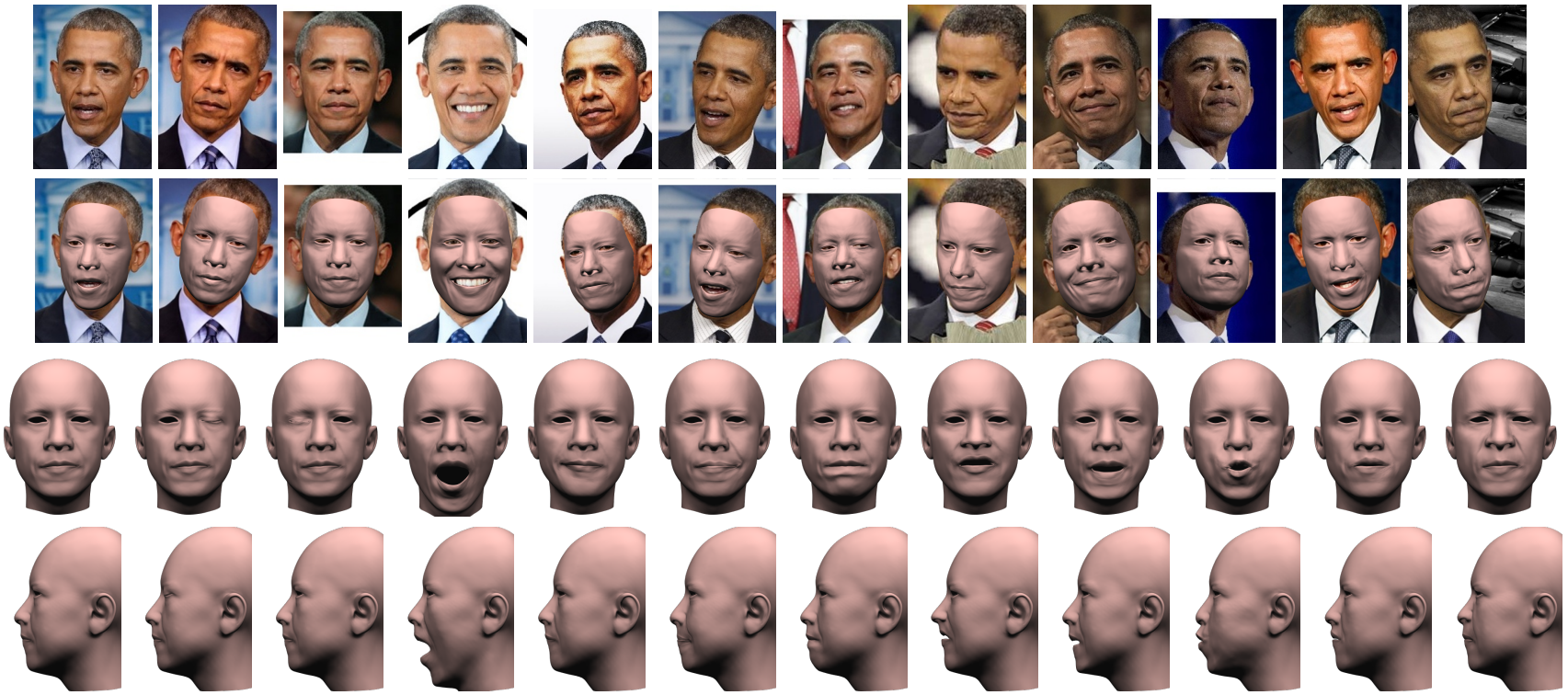


Figure 8.5: Automatic blendshapes generation - subject 5. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [BO1], [BO5], [BO2], [BO3], [BO6], [BO4], [BO12], [BO7], [BO8], [BO9], [BO11], [BO10].

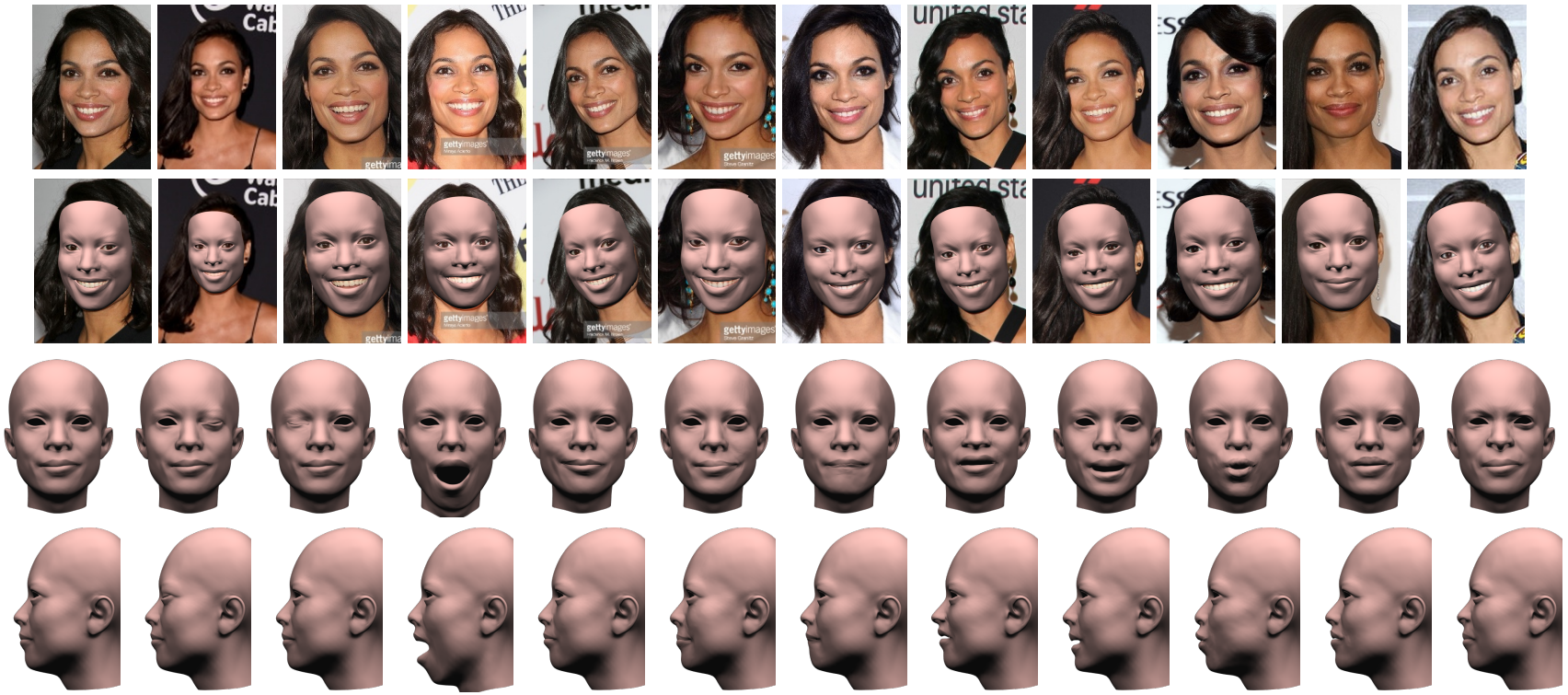


Figure 8.6: Automatic blendshapes generation - subject 6. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [RD1], [RD2], [RD3], [RD4], [RD5], [RD6], [RD7], [RD8], [RD9], [RD10], [RD11], [RD12].

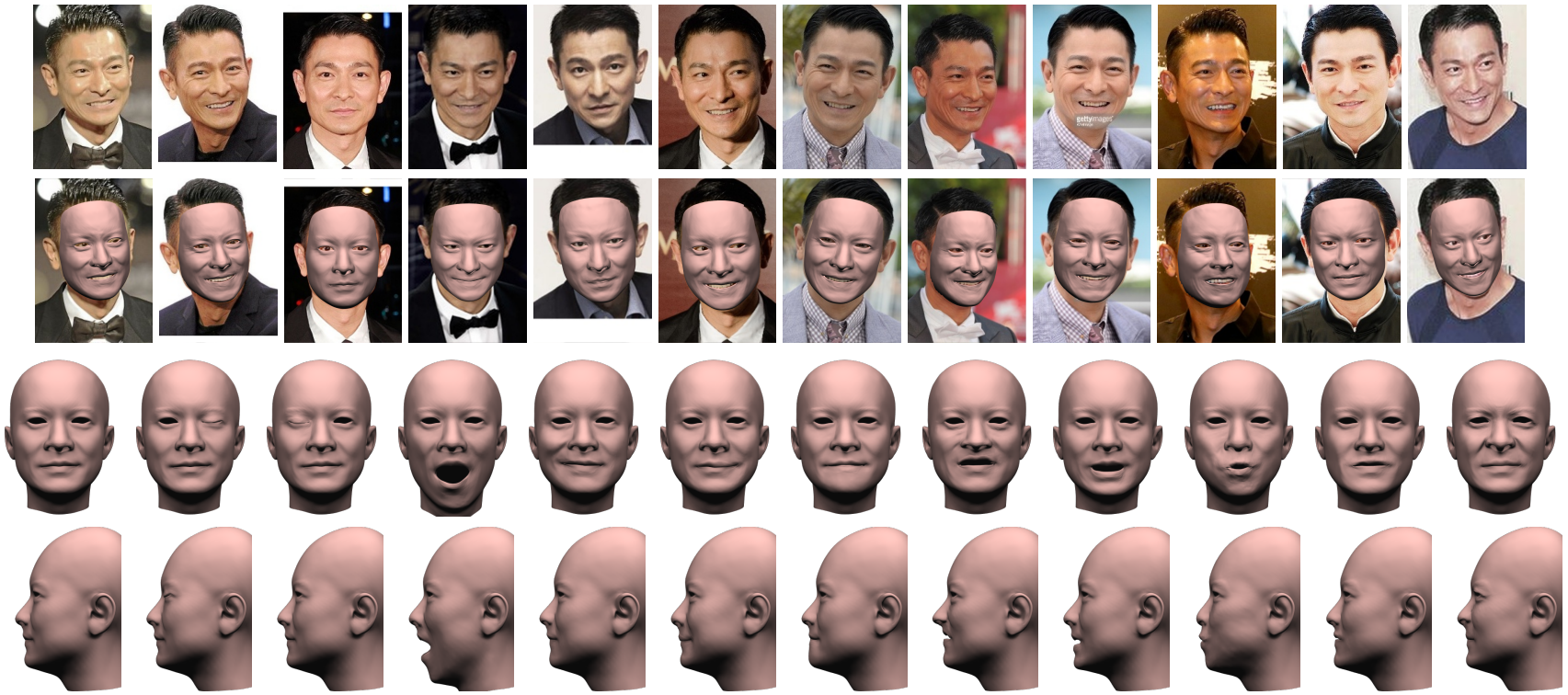


Figure 8.7: Automatic blendshapes generation - subject 7. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [AL1], [AL2], [AL5], [AL21], [AL15], [AL17], [AL26], [AL40], [AL42], [AL44], [AL72], [AL78].

Figure 8.8: Automatic blendshapes generation - subject 8. From top to bottom: input images, face reconstruction with the generated blendshapes, blendshapes generated by our system. We show only 12 out of about 100 input images and 12 out of 46 blendshapes here. Photographs reprinted from [ZZ9], [ZZ4], [ZZ10], [ZZ3], [ZZ6], [ZZ8], [ZZ14], [ZZ15], [ZZ16], [ZZ13], [ZZ11], [ZZ17].

Figure 8.9: Facial reconstructions produced in each step of our progressive reconstruction and after each iteration of blendshapes retargeting - part 1. From left to right: input image, multi-linear reconstruction, after 1 iteration of blendshapes retargeting, after 2 iteration of blendshapes retargeting, after 3 iteration of blendshapes retargeting, final reconstruction with the output blendshapes. Photographs reprinted from [AL26], [BW13], [GB6], [OP11], [HC9], [BO8], [RD3].

Figure 8.10: Facial reconstructions produced in each step of our progressive reconstruction and after each iteration of blendshapes retargeting - part 2. From left to right: input image, multi-linear reconstruction, after 1 iteration of blendshapes retargeting, after 2 iteration of blendshapes retargeting, after 3 iteration of blendshapes retargeting, final reconstruction with the output blendshapes. Photographs reprinted from [OP12], [OP13], [OP14], [OP15], [OP16], [KS9], [ZZ8].

9. CONCLUSION AND FUTURE WORK

Animating virtual characters with realistic expressions is a long-standing challenge in computer graphics, and facial rigs is the core component in generating high quality facial animation. Enabling simple and low-cost creation of facial rigs for both professional artists and common users has potentally broad impacts in many fields including video games, movies, social and security applications. The research presented in this thesis addresses the problem of generating high quality personalized blendshapes using images in the wild.

We proposed an end-to-end system that generates high fidelity blendshapes from a collection of unstructured, unconstrained images. The key idea of our method is to enable robust generation of high quality blendshapes from images in the wild by incorporating crucial failure detection and quality control in our pipeline. The blendshape generation system utilizes a multi-linear model to provide prior knowledge about 3D face shapes for recovering large-scale deformation with the 2D facial feature points detected in input images. However, the 2D facial feature points detected from the challenging input images are usually unreliable and could cause failure in facial reconstruction. Our system includes a robust PCA based outlier detection module to handle such failure cases, and further utilizes a subset selection method to make the facial reconstruction process more robust.

The large-scale facial deformation reconstructed using sparse 2D facial features does not capture all the pixel information in the input images, so the large-scale facial reconstructions do not reflect the true facial geometry of the person being modeled. To address this issue, our system recovers per-pixel point clouds from the large-scale facial reconstruction and input images using shape-from-shading method, which provide per-pixel measurement of the facial geometry. Our system retargets the point clouds to a template set of blendshapes to generate personalized blendshapes. Even with the dense point clouds, the generated blendshapes are still limited by the generalization ability of the multi-linear model. To overcome this limitation, we proposed a point clouds based blendshapes retarting method that combines the deformation gradients from a template set of blendshapes and the per-pixel measurements of the point clouds to generate personalized blend-

shapes.

Fine-scale facial detail is crucial in the perception of certain facial expressions. Due to the limitation by computational cost, we generate blendshapes using medium resolution meshes, which do not capture all fine-scale geometric detail. To improve the realism of the final blendshapes, our system further recovers fine-scale facial detail using a corrective normal maps regression method to estimates proper corrective normals for a given expression. Our results show that the the system is able to generate highly personalized blendshapes with plausible visual appearance.

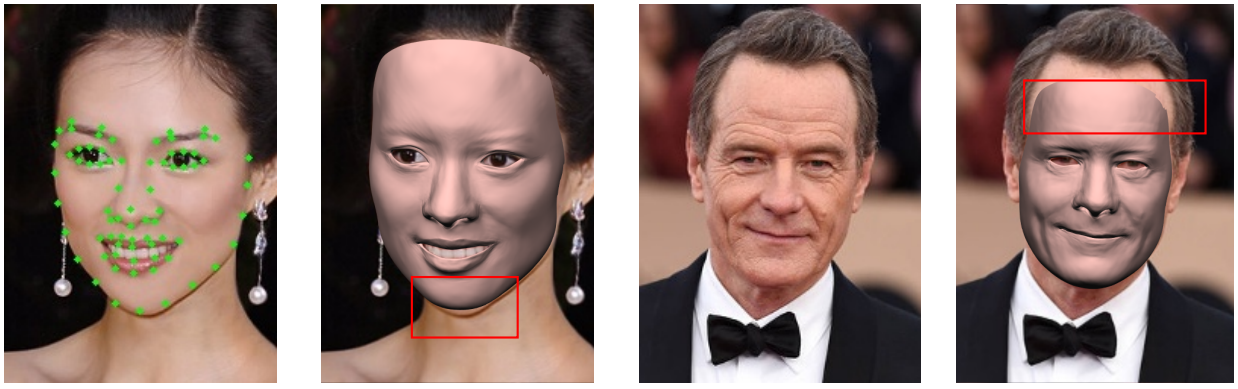
Our system is appealing because it is fully automatic and generates blendshapes with faithful large-scale deformation and reasonable medium and fine-scale detail. We have tested our system on images downloaded from the Internet, demonstrating its accuracy and robustness under challenging combinations of head poses, facial expressions, and illumination.

Creating vivid facial animations requires a complete facial rig including several other components in addition to the facial geometry part. For example, the gaze of the virtual character plays an important part in conveying the emotion of the character, while the tongue and teeth are also indispensable components for realistic motion of the lower face. Similarly, the motion of ears is necessary for presenting certain expressions. Finally, hair and facial hair are crucial in creating distint styles for the virtual characters, though they are not considered as part of a facial rig in general. This thesis focused on the core part of a facial rig, i.e. the face geometry and its deformation model, and ignored all other components. This is a major limitation of this work, and we will investigate how to integrate different components into a unified facial rig in the future.

Texture is another important aspect of a facial rig that directly affects the realism of the synthesized facial animation. Good texture could help improve the visual appearance of a moderate quality facial rig, while poor texture could lead to unpleasant perception of the generated facial animation. In this thesis, we focused on modeling the geometric aspect of the blendshapes and did not generate texture for the final blendshapes. In the future, we will investigate how to create high quality texture and generate photo-realistic blendshapes. Since good texture could also help produce more accurate point clouds in the shape-from-shading algorithm, it is this would improve

both the accuracy and visual appearance of the generated blendshapes.

Our system consists of several key modules to generate the blendshapes, and the performance of our system depends on the performance of these modules. For example, the quality of the 2D facial feature points detected in the input images directly affect the accuracy of the large-scale facial reconstructions. Since the 2D feature points detector inevitably failed on challenging inputs, we included robust PCA based outlier detection and subset selection method to make the reconstruction process more robust. Nonetheless, improvements on the 2D facial feature points detection quality could still improve the large-scale reconstruction quality, which further leads to more accurate final blendshapes.



(a) (b)

Figure 9.1: Limitations of our system. Our system may neglect slight mismatch of the large-scale face shape if it is not critical to the quality of the final blendshapes (a); and some fine-scale detail that is typically inconsistent across input images may be over-smoothed by our fine detail recovery algorithm. Photographs reprinted from [ZZ7] and [OP10].

The large-scale deformation result obtained with the multi-linear model may be sub-optimal in some challenging cases and cannot be detected by our system, which may lead to poor face rigging results. Figure. 9.1 (a) illustrates a typical example of such cases. The slight mismatch of the jaw is considered a non-critical issue by our system and thus is not excluded in our blendshapes generation process. Our system uses a dynamic Gaussian mixture model based method to detect skin region for recovering point clouds. This heuristics method works well for most cases but could fail

on challenging input images. Recent advances in image segmentation using deep learning based approaches may significantly improve the performance of this module and we will experiment with these methods in the future. The shape-from-shading (SfS) method used in our system is a global method that recovers point cloud for the entire face region at once, which could produce some undesired artifacts for certain regions. For example, regions with soft shadow could be interpreted by geometric change in the SfS algorithm, leading to inaccurate geometry. A possible solution to this problem is to use a region-based local SfS method, which may produce higher quality point clouds in such cases. The fine-scale detail recovery method in our system cannot recover fine-scale details that do not consistently appear in different images. We would also like to experiment with a region-based point cloud recovery approach to better handle non-critical occlusions to maximize usable images and image regions. Figure. 9.1(b) shows an example of over-smoothed wrinkles on the forehead. We will explore other methods for recovering fine-scale detail in the future.

The processing time of our system can also be significantly reduced by using a faster C++ implementation instead of the current MATLAB version for point cloud recovery. We plan to speed up the current system with GPU acceleration and exploit a higher degree of parallelism.

We also plan to explore potential applications of our system, such as generating a personal avatar directly from a personal or family album. The personalized blendshapes generated by our system could also be used for facial reenactment similar to [44]. Image and video editing such as expression cloning and facial geometry manipulation are also possible with our system.

REFERENCES

[1] 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 Trans. Graph.*, vol. 35, pp. 28:1–28:15, May 2016.

[2] F. Shi, H.-T. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM Trans. Graph.*, vol. 33, pp. 222:1–222:13, Nov. 2014.

[3] J. Roth, Y. Tong, and X. Liu, "Adaptive 3d face reconstruction from unconstrained photo collections," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4197–4206, June 2016.

[4] I. Kemelmacher-Shlizerman and S. M. Seitz, "Face reconstruction in the wild," in *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, (Washington, DC, USA), pp. 1746–1753, IEEE Computer Society, 2011.

[5] P. Bergeron and P. Lachapelle, "Controlling facial expressions and body movements in the computer generated short 'tony de peltrie'," in *SIGGRAPH 1985 Tutorial Notes, Advanced Computer Animation Course*, 1985.

[6] H. Huang, J. Chai, X. Tong, and H.-T. Wu, "Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition," in *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, (New York, NY, USA), pp. 74:1–74:10, ACM, 2011.

[7] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, "The digital emily project: Photoreal facial modeling and animation," in *ACM SIGGRAPH 2009 Courses*, SIGGRAPH '09, (New York, NY, USA), pp. 12:1–12:15, ACM, 2009.

[8] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging," *ACM Trans. Graph.*, vol. 29, pp. 32:1–32:6, July 2010.

[9] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *ACM Trans. Graph.*, vol. 30, pp. 77:1–77:10, July 2011.

[10] S. Bouaziz, Y. Wang, and M. Pauly, "Online modeling for realtime facial animation," *ACM Trans. Graph.*, vol. 32, pp. 40:1–40:10, July 2013.

[11] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3d shape regression for real-time facial animation," *ACM Trans. Graph.*, vol. 32, pp. 41:1–41:10, July 2013.

[12] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, pp. 158:1–158:10, Nov. 2013.

[13] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives," *ACM Trans. Graph.*, vol. 32, pp. 42:1–42:10, July 2013.

[14] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, pp. 43:1–43:10, July 2014.

[15] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, pp. 46:1–46:9, July 2015.

[16] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3d avatar creation from hand-held video input," *ACM Trans. Graph.*, vol. 34, pp. 45:1–45:14, July 2015.

[17] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making faces," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, (New York, NY, USA), pp. 55–66, ACM, 1998.

[18] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," in *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, (New York, NY, USA), pp. 41:1–41:10, ACM, 2010.

[19] F. I. Parke, *A Parametric Model for Human Faces.* PhD thesis, University of Utah, 1974. AAI7508697.

[20] K. Waters, "A muscle model for animation three-dimensional facial expression," *SIGGRAPH Comput. Graph.*, vol. 21, pp. 17–24, Aug. 1987.

[21] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, (New York, NY, USA), pp. 55–62, ACM, 1995.

[22] E. Sifakis, I. Neverov, and R. Fedkiw, "Automatic determination of facial muscle activations from sparse motion capture marker data," in *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, (New York, NY, USA), pp. 417–425, ACM, 2005.

[23] 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*, SIGGRAPH '99, (New York, NY, USA), pp. 187–194, ACM Press/Addison-Wesley Publishing Co., 1999.

[24] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," *ACM Trans. Graph.*, vol. 24, pp. 426–433, July 2005.

[25] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and X. Alvarez, "A Facial Rigging Survey," in *Eurographics 2012 - State of the Art Reports* (M.-P. Cani and F. Ganovelli, eds.), The Eurographics Association, 2012.

[26] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng, "Practice and Theory of Blendshape Facial Models," in *Eurographics 2014 - State of the Art Reports* (S. Lefebvre and M. Spagnuolo, eds.), The Eurographics Association, 2014.

[27] B. Raitt, "The making of gollum." Presentation at U. Southern California Institute for Creative TechnologiesâĂŹs Frontiers of Facial Animation Workshop, August 2004.

[28] M. Sagar1, "Facial performance capture and expressive translation for king kong," in *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, (New York, NY, USA), ACM, 2006.

[29] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross, "Multi-scale capture of facial geometry and motion," in *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, (New York, NY, USA), ACM, 2007.

[30] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime faces: High resolution capture for modeling and animation," in *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, (New York, NY, USA), pp. 548–558, ACM, 2004.

[31] T. Weise, H. Li, L. Van Gool, and M. Pauly, "Face/off: Live facial puppetry," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, (New York, NY, USA), pp. 7–16, ACM, 2009.

[32] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," in *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, (New York, NY, USA), pp. 40:1–40:9, ACM, 2010.

[33] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt, "Lightweight binocular facial performance capture under uncontrolled lighting," *ACM Trans. Graph.*, vol. 31, pp. 187:1–187:11, Nov. 2012.

[34] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," in *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, (New York, NY, USA), pp. 77:1–77:10, ACM, 2011.

[35] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3d facial expression database for visual computing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 413–425, March 2014.

[36] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, ch. Total Moving Face Reconstruction, pp. 796–812. Cham: Springer International Publishing, 2014.

[37] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, (New York, NY, USA), pp. 75–84, ACM, 1998.

[38] B. Choe and H.-S. Ko, "Analysis and synthesis of facial expressions with hand-generated muscle actuation basis," in *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, (New York, NY, USA), ACM, 2005.

[39] X. Liu, T. Mao, S. Xia, Y. Yu, and Z. Wang, "Facial animation by optimized blendshapes from motion capture data," *Computer Animation and Virtual Worlds*, vol. 19, no. 3-4, pp. 235–245, 2008.

[40] J.-y. Noh and U. Neumann, "Expression cloning," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, (New York, NY, USA), pp. 277–288, ACM, 2001.

[41] V. C. Orvalho, E. Zacur, and A. Susin, "Transferring the rig and animations from a character to different face models," *Computer Graphics Forum*, vol. 27, no. 8, pp. 1997–2012, 2008.

[42] F. Xu, J. Chai, Y. Liu, and X. Tong, "Controllable high-fidelity facial performance transfer," *ACM Trans. Graph.*, vol. 33, pp. 42:1–42:11, July 2014.

[43] T. J. Cashman and A. W. Fitzgibbon, "What shape are dolphins? building 3d morphable models from 2d images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 232–244, Jan 2013.

[44] J. Thies, M. ZollhÃűfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2387–2395, June 2016.

[45] A.-E. Ichim, P. Kadleček, L. Kavan, and M. Pauly, "Phace: Physics-based face modeling and animation," *ACM Trans. Graph.*, vol. 36, pp. 153:1–153:14, July 2017.

[46] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics*, vol. 36, pp. 194:1–194:17, Nov. 2017. Two first authors contributed equally.

[47] P. Debevec, "The Light Stages and Their Applications to Photoreal Digital Actors," in *SIGGRAPH Asia*, (Singapore), Nov. 2012.

[48] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, “Creating a photoreal digital actor: The digital emily project,” in *Visual Media Production, 2009. CVMP ’09. Conference for*, pp. 176–187, Nov 2009.

[49] A. Ghosh, G. Fyffe, B. Tunwattanapong, J. Busch, X. Yu, and P. Debevec, “Multiview face capture using polarized spherical gradient illumination,” in *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA ’11, (New York, NY, USA), pp. 129:1–129:10, ACM, 2011.

[50] P. Dou, S. K. Shah, and I. A. Kakadiaris, “End-to-end 3d face reconstruction with deep neural networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[51] E. Richardson, M. Sela, R. Or-El, and R. Kimmel, “Learning detailed face reconstruction from a single image,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[52] G. Trigeorgis, P. Snape, I. Kokkinos, and S. Zafeiriou, “Face normals "in-the-wild" using fully convolutional networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[53] S. Ren, X. Cao, Y. Wei, and J. Sun, “Face alignment at 3000 fps via regressing local binary features,” in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1685–1692, June 2014.

[54] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*. Katholieke Universiteit Leuven Faculteit der Toegepaste Wetenschappen Department Elektrotechniek, 1997.

[55] P. DollÃąr, P. Welinder, and P. Perona, “Cascaded pose regression,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1078–1085, June 2010.

[56] X. Xiong and F. D. la Torre, “Supervised descent method and its applications to face alignment,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 532–539, June 2013.

[57] I. Matthews and S. Baker, "Active Appearance Models Revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[58] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds.), pp. 612–620, Curran Associates, Inc., 2011.

[59] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591, Jun 1991.

[60] I. Kemelmacher-Shlizerman and R. Basri, "3d face reconstruction from a single image using a single reference face shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 394–405, Feb 2011.

[61] B. K. Horn, *Obtaining shape from shading information*. MIT press, 1989.

[62] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 218–233, Feb 2003.

[63] B. K. Horn and M. J. Brooks, "The variational approach to shape from shading," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 2, pp. 174 – 208, 1986.

[64] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. SÃijsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2274–2282, Nov 2012.

[65] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34–41, Sep 2001.

[66] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," in *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, (New York, NY, USA), pp. 399–405, ACM, 2004.

[67] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, (New York, NY, USA), pp. 175–184, ACM, 2004.

[68] H. Li, R. W. Sumner, and M. Pauly, "Global correspondence optimization for non-rigid registration of depth scans," in *Proceedings of the Symposium on Geometry Processing*, SGP '08, (Aire-la-Ville, Switzerland, Switzerland), pp. 1421–1430, Eurographics Association, 2008.

APPENDIX A

EXTERNAL IMAGE REFERENCES

A.1 Andy Lau

[AL1] https://www.star2.com/entertainment/tv/2014/01/28/andy-lau-to-get-rm166mil-as-judge-of-the-voice-of-china/

[AL2] http://www.cnbinggui.net/332497-andy-lau.html

[AL3] http://info.51.ca/news/sportent/2018-01/618366.html

[AL4] https://forums.hardwarezone.com.sg/eat-drink-man-woman-16/gpgt-zehzeh-stare-me-like-i-am-andy-lau-5775443.html

[AL5] http://www.xiaoguaiguai.com/tags/mingxingliudehua.html

[AL6] http://www.awc618.com/movielava-bear-to-produce-chinese-action-thriller-the-bodyguard-starring-andy-lau/

[AL7] http://www.campaignasia.com/article/osim-launches-udivine-massage-chair-with-andy-lau/238902

[AL8] https://www.thestar.com.my/lifestyle/entertainment/movies/news/2015/03/14/andy-lau-matt-damon-to-star-in-the-great-wall/

[AL9] http://www.n63.com/n_chinam/ldh

[AL10] http://top10famous.com/top-10-famous-chinese-movie-actors/

[AL11] https://www.thestar.com.my/news/nation/2014/10/03/andy-lau-to-set-up-production-company-in-msia/

[AL12] https://hype.my/2013/14368/andy-lau-apologizes-for-his-role-in-film/

[AL13] http://pinthisstar.com/andy-lau.html

[AL14] http://www.madisonboom.com/2013/09/27/mbtv-andy-liu-stared-cartier-micro-movie/andy-liu-cartier-inarticle/

[AL15] https://mubi.com/cast/andy-lau

[AL16] http://movie.info/andy-lau

[AL17] http://www.scmp.com/photos/1193895/37th-hong-kong-international-film-festival/7th-asian-film-awards

[AL18] http://www.chinadaily.com.cn/celebrity/2014-09/29/content_18680418.htm

[AL19] https://reelgood.com/person/andy-lau-1961

[AL20] https://gossipstar.mthai.com/hollywood/inter/50714

[AL21] http://www.zimbio.com/photos/Andy+Lau/Arrivals+Beijing+International+Film+Festival/qM8PvNRHFXv

[AL22] http://www.craveonline.com/site/191049-andy-lau-leaves-the-cast-of-iron-man-3

[AL23] http://www.kklee105.com/april2013toAugust2013.html

[AL24] http://www.czeshop.info/tendency/andy-lau.html

[AL25] http://www.chinadaily.com.cn/celebrity/2014-09/29/content_18680418_4.htm

[AL26] https://www.theepochtimes.com/china-media-authorities-censors-andy-lau-and-chow-yun-fat_1023425.html

[AL27] http://www.chinadaily.com.cn/celebrity/2014-09/29/content_18680418_3.htm

[AL28] http://www.zimbio.com/Andy+Lau/pictures/pro/2005

[AL29] http://www.straitstimes.com/lifestyle/entertainment/andy-lau-goes-back-to-work-on-aug-8

[AL30] http://www.tnp.sg/entertainment/andy-lau-says-wife-not-pregnant-just-fat

[AL31] http://www.czeshop.info/tendency/andy-lau.html

[AL32] https://www.elcinema.com/en/person/2028321/

[AL33] https://www.mgientertainment.com/tag/andy-lau/

[AL34] http://movie.info/andy-lau

[AL35] https://www.brilio.net/selebritis/30-tahun-lebih-berkakrier-ini-12-transformasi-penampilan-andy-lau-170125l.html

[AL36] http://www.contactmusic.com/andy-lau/pictures/1463909#slider-1

[AL37] http://www.boomsbeat.com/articles/10051/20141016/great-photos-of-the-andy-lau.htm

[AL38] http://www.contactmusic.com/andy-lau/pictures/1463909#slider-2

[AL39] http://www.asiaone.com/showbiz/andy-lau-get-63m-voice-judge

[AL40] http://www.boomsbeat.com/articles/10051/20141016/great-photos-of-the-andy-lau.htm

[AL41] http://www.zimbio.com/photos/Andy+Lau/Blind+Detective+Photo+Call+Cannes/BZD3INCoB6j

[AL42] https://www.gettyimages.com/event/blind-detective-photocall-the-66th-annual-cannes-film-festival-169201688#/johnnie-to-attends-the-photocall-for-blind-detective-during-the-66th-picture-id169100923

[AL43] https://www.screendaily.com/news/filmart-fox-working-with-andy-lau-patrick-kong-on-miniseries/5115814.article

[AL44] http://www.asiaone.com/showbiz/andy-laus-wife-rumoured-have-secretly-given-birth-boy

[AL45] https://www.23yy.com/3940000/3936664.shtml

[AL46] Photograph by Chung Sung-Jun/Getty Images (Octoboer 14, 2006) reprinted from http://www.boomsbeat.com/articles/10051/20141016/great-photos-of-the-andy-lau.htm

[AL47] http://www.contactmusic.com/andy-lau/pictures/1463909#slider-8

[AL48] http://picture-xtreme-nice.blogspot.com/2012/05/andy-lau-photos-hot.html

[AL49] http://chinesemov.com/actors/Andy%20Lau.html

[AL50] http://www.jpopasia.com/feed/11489/andy-lau-confesses-past-potential-girlfriends/

[AL51] http://best1-hairstyle.blogspot.com/2015/12/popular-hongkong-artist-andy-lau-hair.html

[AL52] https://ascendcomblog.com/tag/screensingapore/

[AL53] http://chinesemov.com/actors/Andy%20Lau.html

[AL54] http://www.zimbio.com/photos/Andy+Lau/16th+Shanghai+International+Film+Festival/1TLbhT9iC5t

[AL55] http://www.pinsdaddy.com/andy-lau_m9oFnCUjId9CysE7uE2RR*7iuF6DJlL0H7%7CMye3KPlo/10

[AL56] https://lifeactor.ru/8065-endi-lau.html

[AL57] https://www.wowkeren.com/seleb/andy_lau/berita.html

[AL58] http://www.chinaentertainmentnews.com/2016/03/filmart-andy-lau-boards-action-thriller.html

[AL59] http://movie.info/andy-lau

[AL60] https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%BC:LDH061229.jpg

[AL61] http://www.biendao24h.vn/products/Top-20-my-nam-%C4%91ep-nhat-%C4%91ien-anh-Trung-Quoc..html

[AL62] http://www.contactmusic.com/andy-lau/pictures/1463909#slider-6

[AL63] https://www.pinterest.co.uk/pin/474426141969902619/

[AL64] https://vincentloy.wordpress.com/tag/male/

[AL65] http://www.chinadaily.com.cn/entertainment/2008-04/14/content_6614585.htm

[AL66] http://chinesemov.com/actors2/Andy-Lau-Photos.html

[AL67] http://www.contactmusic.com/andy-lau/pictures/1463909#slider-7

[AL68] http://style.tribunnews.com/2017/01/19/andy-lau-cedera-karena-jatuh-dari-kuda-saat-syuting-di-thailand-begini-kondisinya-sekarang

[AL69] http://www.myalt.net/andy-lau.html

[AL70] http://www.epochtimes.com/b5/3/12/13/n428824.htm

[AL71] https://www.gettyimages.co.uk/event/lost-and-love-beijing-press-conference-533899263#/actor-andy-lau-attends-director-sanyuan-pengs-film-lost-and-love-on-picture-id461949626

[AL72] http://www.epochtimes.com/gb/4/5/26/n550119.htm

[AL73] http://www.contactmusic.com/andy-lau/pictures/1463909#slider-3

[AL74] http://wwwbuild.net/girlnba/31359.html

[AL75] http://www.brns.com/pages4/andylau4.html

[AL76] http://hktopten.blogspot.com/2015/09/20150930-andy-lau-shares-his-happiness.html

[AL77] https://andylausounds.blogspot.com/2006/08/andys-new-photographs-with-charlene.html

[AL78] http://www.kklee105.com/july2010.html

[AL79] http://www.laineygossip.com/Chow-Yun-Fat-and-Andy-Lau-promote-The-Man-From-Macau-III/41512

[AL80] http://yellowcranestower.blogspot.com/2011/01/andy-lau-concert-photos-4.html

[AL81] http://yellowcranestower.blogspot.com/2010/05/andy-lau-shanghai-world-expo.html

[AL82] http://www.chinadaily.com.cn/english/doc/2005-02/24/content_419158.htm

[AL83] https://kknews.cc/zh-hk/entertainment/gj9ap8.html

[AL84] https://www.arah.com/article/41431/keren-aktor-film-laga-andy-lau-dapat-gelar-doktor-kehormatan.html

[AL85] https://www.wethepeoplewiki.com/andy-lau-haircut/

[AL86] http://pinthisstar.com/image-post/38-andy-lau-movies-19.jpg.html#gal_post_38_andy-lau-movies-19.jpg

[AL87] http://yummycelebrities.com/2006/10/15/andy-lau-honored-at-busan-international-film-festival/

[AL88] https://www.ctvnews.ca/lau-hopes-sci-fi-film-breaks-new-ground-1.500233

[AL89] Photograph by Junko Kimura/Getty Images reprinted from http://www.zimbio.com/photos/Andy+Lau/Entertainment+Pictures+Week+2006+July+27/4vUvH_QWsCE

[AL90] http://english.cri.cn/3086/2008/07/02/1221s375765.htm

[AL91] http://best1-hairstyle.blogspot.com/2015/12/popular-hongkong-artist-andy-lau-hair.html

A.2 Barack Obama

[BO1] Photograph by Pablo Martinez Monsivais/AP reprinted from https://www.usatoday.com/story/news/politics/2017/01/19/fact-check-eight-years-trolling-obama/96771984/

[BO2] https://imgur.com/gallery/8Z0tf1C

[BO3] https://www.telegraph.co.uk/news/2016/04/21/as-your-friend-let-me-tell-you-that-the-eu-makes-britain-even-gr/

[BO4] Photograph by Charles Dharapak/AP reprinted from https://face2faceafrica.com/article/forbes-most-powerful-people-2016

[BO5] Photograph by Chip Somodevilla (December 16, 2016) by Getty Images reprinted from https://www.usatoday.com/story/news/politics/2016/12/16/obama-always-feel-responsible-aleppo/95523850/

[BO6] http://www.wccbcharlotte.com/tag/obama/page/2/

[BO7] http://freedomessenger.com/en/mullah%60s-nuclear-ambitions/1637-obama-sends-pallets-of-cash-to-iran-as-ransom

[BO8] https://www.theblaze.com/contributions/latest-gdp-report-proves-that-obama-is-purposefully-destroying-america

[BO9] https://www.huffingtonpost.com/entry/obama-nsa-reform_n_6580702.html

[BO10] http://www.armradio.am/en/2016/09/23/anca-disappointed-with-obamas-letter-on-armenias-independence/

[BO11] http://www.dailymail.co.uk/news/article-2151996/Michelle-Obamas-drug-use-He-realised-young-age-life-smoke-pot.html

[BO12] Photograph by Alex Brandon/AP reprinted from https://www.theguardian.com/media/2016/apr/05/daily-mail-barack-obama-peace-sign-president

A.3 Bruce Willis

[BW1] http://www.imdb.com/name/nm0000246/

[BW2] https://en.wikipedia.org/wiki/Bruce_Willis

[BW3] http://www.filmofilia.com/a-good-day-to-die-hard-gets-r-rating-from-mpaa-132321/

[BW4] https://www.pinterest.com/pin/427067977135033698

[BW5] http://planetterror.wikia.com/wiki/File:Bruce_Willis.jpg

[BW6] https://www.bunte.de/starprofile/bruce-willis.html

[BW7] https://theidleman.com/manual/health/bruce-willis-hair-loss/

[BW8] http://celebritiesmovie.com/celebrities-detail/bruce-willis-phone-number-email/

[BW9] https://www.usmagazine.com/celebrity-news/news/bruce-willis-wears-donald-trump-wig-on-jimmy-fallon-20151910/

[BW10] https://www.pinterest.com/pin/11962755244817671/

[BW11] http://wvxu.org/post/bruce-willis-shoot-another-film-here-august#stream/0

[BW12] http://entertainment.ie/trending/news/Pic-Bruce-Willis-wins-Halloween-with-this-amazingly-creepy-costume/398713.htm

[BW13] http://www.ktvb.com/article/news/local/idaho/documents-show-bruce-williss-airport-will-be-large-busy/277-458503461

A.4 George Bush

[GB1] http://babailov.homestead.com/Bushfront.html

[GB2] https://antifascistnews.net/2017/0/page/5/

[GB3] http://www.nndb.com/people/360/000022294/

[GB4] http://thehill.com/blogs/ballot-box/gop-primaries/255801-george-w-bush-to-fundraise-for-jeb-in-washington

[GB5] http://www.patheos.com/blogs/philosophicalfragments/2011/10/20/reconsidering-bushs-compassionate-conservatism/

[GB6] http://www.cnn.com/2009/POLITICS/09/04/mann.george.w.bush/index.html

[GB7] http://thehill.com/blogs/blog-briefing-room/news/312525-george-w-bush-to-attend-trumps-inauguration

[GB8] https://nkreligiousviolations.wordpress.com/

[GB9] Photograph by Jason Reed/Reuters reprinted from https://www.theguardian.com/world/2010/oct/29/george-bush-thought-9-11-plane-shot-down

[GB10] http://fox2now.com/2018/02/08/george-w-bush-says-russia-meddled-in-2016-us-election

[GB11] http://picmeme.bid/george-bush-speech/george-w-bush-gives-his-post-presidency-speech-zimbio.html

[GB12] http://www.ohmynews.com/NWS_Web/View/at_pg.aspx?CNTN_CD=A0000729761

[GB13] Photograph by Paul Drinkwater/NBC reprinted from http://people.com/celebrity/president-george-w-bush-to-have-first-gallery-showing/

[GB14] https://www.telegraph.co.uk/news/picturegalleries/worldnews/4223984/My-Way-by-George-W-Bush.html?image=20

A.5 Hillary Clinton

[HC1] https://history.state.gov/departmenthistory/people/clinton-hillary-rodham

[HC2] https://www.salon.com/2012/08/16/hillary_clinton_does_not_have_time_for_your_games/

[HC3] https://mightyfee.wordpress.com/2016/02/08/why-i-will-never-ever-vote-for-hillary-clinton/us-secretary-of-state-hillary-clinton-ge/

[HC4] https://saintpetersblog.com/hillary-clinton-releases-tax-health-records-on-a-busy-friday

[HC5] https://www.huffingtonpost.com/2012/05/24/hillary-clinton-pops-collar-senate-treaty_n_1542796.html

[HC6] http://www.businessinsider.com/rnc-nbc-cnn-hillary-clinton-debate-resolution-2013-8

[HC7] http://www.businessinsider.com/hillary-clinton-private-server-company-platte-2015-8

[HC8] http://freebeacon.com/blog/rove-hillary-may-have-brain-damage/

[HC9] http://billlawrenceonline.com/larry-flynt-chester-molester-hillary/

[HC10] https://www.newsmax.com/newsfront/ed-klein-hillary-plea-bargain/2017/08/08/id/806488/

[HC11] https://bethechange2012.com/tag/hillary-rodham-clinton/

[HC12] https://loviribolov.biz/11357/hillary-clinton-quote

[HC13] http://www.patheos.com/blogs/themediawitches/2016/11/hillary-clinton-is-not-a-witch-or-is-she/

[HC14] https://artsandculture.google.com/asset/hillary-rodham-clinton-official-portrait-u-s-senate-historical-office/SgHM7MLNLnitqw

[HC15] Photograph by Justin Sullivan by Getty Images reprinted from http://fortune.com/2016/09/22/hillary-clinton-between-two-ferns-zach-galifianakis/

[HC16] Photograph by Matt Rourke/AP reprinted from https://www.forbes.com/pictures/egeh45fjkej/the-many-faces-of-hillar/#70d6fb2166da

A.6 Jessica Stroup

[JS1] https://www.picsofcelebrities.com/celebrites/jessica-stroup.html

[JS2] http://www.listal.com/viewimage/6341171h

[JS3] http://www.vettri.net/gallery/celeb/jessica_stroup/90210-Season-Wrap-Party/JessicaStroup_90210-Season-Wrap-Party-05.html

[JS4] https://www.pkbaseline.com/jessica-stroup-height-weight

[JS5] https://www.themoviedb.org/person/55463-jessica-stroup/images/profiles

[JS6] http://www.taaz.com/hairstyles/jessica-stroup-hair-style-317.html

[JS7] http://koench.com/jessica-stroup-short-hairstyles/jessica-stroup-short-hairstyles-trends-looks-and-jessica-stroup-short-hairstyles-hairstyles-on-pinterest/

[JS8] https://movietimes.jp/celebrities/jessica-stroup

[JS9] http://3.bp.blogspot.com/-JncBnpObPoU/UjjoxahMcMI/AAAAAAAG9kk/UXmpr_D67wM/s1600/jessica_stroup_1270006.jpg

[JS10] http://www.justjaredjr.com/2014/01/14/jessica-stroup-the-following-tca-panel/

[JS11] https://www.celebritysizes.com/jessica-stroup-workout-routine/

[JS12] https://www.themoviedb.org/person/55463-jessica-stroup/images/profiles

A.7 Kevin Spacey

[KS1] http://www.imdb.com/name/nm0116232/mediaviewer/rm889225472

[KS2] https://www.mirror.co.uk/3am/celebrity-news/kevin-spacey-hints-could-more-11433513

[KS3] https://www.theplace2.ru/photos/Kevin-Spacey-md668/

[KS4] http://chasingspacey.tumblr.com/post/102594275147/by-request-kevins-trying-hard-not-to-laugh

[KS5] Photograph by Douglas Gorenstein/NBC/NBCU Photo Bank (August 14, 2015) via Getty Images reprinted from https://www.gettyimages.com/event/s-the-tonight-show-starring-jimmy-fallon-with-guests-kevin-spacey-keegan-michael-key-monroe-martin-569765033#episode-0313-pictured-actor-kevin-spacey-on-august-14-2015-picture-id484047574

[KS6] Photograph by Jay L. Clendenin / Los Angeles Times reprinted from http://www.latimes.com/entertainment/movies/la-et-mn-kevin-spacey-20140507-story.html

[KS7] Photograph by Jay L. Clendenin (April 29, 2014) by Getty Images reprinted from http://www.gettyimages.ie/event/kevin-spacey-los-angeles-times-may-7-2014-489293513#actor-kevin-spacey-is-photographed-for-los-angeles-times-on-april-29-picture-id488794199

[KS8] https://www.britannica.com/biography/Kevin-Spacey

[KS9] http://www.fansshare.com/gallery/photos/19099319/spacey-kevin-spacey-kevin-spacey/

[KS10] Photograph by Photo by Frazer Harrison (2011) by Getty Images reprinted from http://www.imdb.com/name/nm0000228/mediaviewer/rm3121592832

[KS11] http://www.justjared.com/photo-gallery/3314915/robin-wright-kevin-spacey-house-of-cards-premiere-02/

[KS12] http://blog.livedoor.jp/shipshapesavour85/archives/3982745.html

[KS13] https://natsimonemua.wordpress.com/2014/03/06/men-of-the-oscars-2014/

[KS14] https://persons-info.com/persons/SPEISI_Kevin

[KS15] https://www.huffingtonpost.com/2014/03/03/kevin-spacey-frank-underwood-oscars_n_4889624.html

[KS16] https://www.aceshowbiz.com/news/view/00075562.html

[KS17] http://collider.com/kevin-spacey-will-play-lobbyist-jack-abramoff-in-casino-jack/

[KS18] http://zeenews.india.com/entertainment/hollywood/kevin-spacey-denies-james-bond-villain-rumours_156414.html

[KS19] https://www.irishexaminer.com/lifestyle/artsfilmtv/ridley-scotts-quick-fix-replacement-of-kevin-spacey-one-of-his-best-ever-decisions-465476.html

A.8 Rosario Dawson

[RD1] http://www.sensacine.com/actores/actor-32976/fotos/

[RD2] http://www.puretrend.com/media/rosario-dawson-devient-mere-a-35-ans_m1131224

[RD3] https://shadowandact.com/rosario-dawson-attached-to-play-womens-rights-activist-donna-hylton-in-biopic/

[RD4] Photograph by Mireya Acierto/FilmMagic reprinted from https://www.gettyimages.com/event/kids-20th-anniversary-screening-bamcinemafest-2015-561472499#actress-rosario-dawson-attends-the-kids-20th-anniversary-screening-picture-id478526410

[RD5] Photograph by Frederick M. Brown/Getty Images reprinted from https://www.gettyimages.com/event/celebrates-the-champions-of-our-planets-future-arrivals-616354311#actress-rosario-dawson-attends-ucla-ioes-celebration-of-the-champions-picture-id517360704

[RD6] http://unapix.com/articles/rosario-dawson-married-husband-boyfriend-and-dating-or-affair.html

[RD7] http://entora.com/artist/4231/Rosario-Dawson

[RD8] https://woman.infoseek.co.jp/news/celebrity/hollywood_03Dec2014_51690

[RD9] https://www.huffingtonpost.com/2014/08/24/chloe-moretz-makeup-best-worst-beauty_n_5699294.html

[RD10] http://www.belezaextraordinaria.com.br/noticia/melhores-da-semana-penteados-sofisticados-marcam-os-looks-das-famosas_a1807/1#9

[RD11] http://go.seoul.co.kr/news/newsView.php?id=20140523500104

[RD12] http://www.eonline.com/news/602697/rosario-dawson-adopts-a-12-year-old-girl-she-has-always-wanted-this

A.9 Ziyi Zhang

[ZZ1] http://www.chinadaily.com.cn/dfpd/2013-04/10/content_16390703_3.htm

[ZZ2] http://chinafilminsider.com/chinese-korean-co-pro-halted-thaad-fallout/

[ZZ3] http://cdn23.us1.fansshare.com/photos/ziyizhang/zhang-ziyi-toronto-international-film-festival-portraits-september-463259156.jpg

[ZZ4] http://www.justjared.com/photo-gallery/2929875/ziyi-zhang-wealthiest-actress-in-greater-china-16/

[ZZ5] http://www.tracking-board.com/chinese-superstar-zhang-ziyi-to-topline-comedy-eastwest/

[ZZ6] http://www.justjared.com/photo-gallery/2914925/ziyi-zhang-the-grandmaster-screening-11/

[ZZ7] http://geniusbeauty.com/celebrity-gossip/actress-zhang-ziyi-accused-prostitution/

[ZZ8] https://asianmoviepulse.com/2012/11/zhang-ziyi-woman-of-many-talents/

[ZZ9] http://www.fanpop.com/clubs/zhang-ziyi/images/3034022/title/zhang-ziyi-photo

[ZZ10] https://www.usmagazine.com/celebrity-moms/news/zhang-ziyi-gives-birth-to-first-child-welcomes-baby-girl-with-wang-feng-w160442/

[ZZ11] http://hollywoodneuz.us/zhang-ziyi/

[ZZ12] http://ethnicelebs.com/zhang-ziyi

[ZZ13] http://www.listal.com/viewimage/1039187

[ZZ14] http://www.justjared.com/2009/05/21/ziyi-zhangs-inglourious-earrings/

[ZZ15] http://www.justjared.com/photo-gallery/2903637/ziyi-zhang-valentino-fashion-show-in-paris-08/

[ZZ16] http://www.sweetandtalented.com/images/zhang/zhang40.jpg

[ZZ17] Photograph by Lia Toby/WENN reprinted from https://www.aceshowbiz.com/events/zhang+ziyi/zhang-ziyi-66th-cannes-film-festival-07.html

A.10 Others

[OP1] http://www.chinahush.com/2009/07/15/yao-ming-bought-his-former-club-shanghai-sharks/

[OP2] http://gl.ict.usc.edu/LightStages/

[OP3] http://indianexpress.com/article/entertainment/hollywood/jennifer-anistons-first-look-photo-in-war-drama-the-yellow-birds-revealed/

[OP4] http://www.justjared.com/photo-gallery/3017225/jennifer-aniston-did-not-shave-her-hair-real-photos-here-03/

[OP5] http://www.fanpop.com/clubs/robert-downey-jr/images/31831924/title/robert-downey-jr-photo

[OP6] http://www.celebritypix.us/celebrities/old-gracefully-celebrities-f1733.html

[OP7] Photograph by Stephen Lam (November 9, 2014) reprinted from http://www.businessinsider.com/mark-zuckerbergs-2015-new-years-resolution-2015-1

[OP8] http://blog.amsterdamfashiontv.com/archives/28656

[OP9] http://feelgrafix.com/956437-tom-hanks.html

[OP10] Photograph by Jordan Strauss reprinted from https://www.newsday.com/entertainment/books/bryan-cranston-talks-new-autobiography-a-life-in-parts-1.12408265

[OP11] Photograph by Hubert Boesl/dpa/Corbis reprinted from http://www.foodandwine.com/fwx/life-sized-chocolate-benedict-cumberbatch-unveiled-england

[OP12] Photograph by Pascal Le Segretain/Getty Images reprinted from https://www.hollywoodreporter.com/news/george-clooney-armenia-as-commemoration-887163

[OP13] Photograph by Michael Buckner/Getty Images reprinted from http://www.zimbio.com/photos/Gong+Li/Grace+Monaco+Premieres+Cannes/_FgLd1Zt9ME

[OP14] http://www.worldstopmost.com/2017-2018-2019-2020/celebrities/richest-entrepreneurs-world-2016-top-10-list/#6_Mark_Zuckerberg

[OP15] http://www.purepeople.com/article/oprah-winfrey-la-femme-la-plus-puissante-battue-par-un-membre-du-ncis_a72155/1

[OP16] Photograph by Jordan Strauss/AP reprinted from http://www.huffingtonpost.ca/2015/02/22/scarlett-johansson-oscars-2015_n_6723492.html