Riggable 3D Face Reconstruction via In-Network Optimization Supplementary Material

Ziqian Bai[1] Zhaopeng Cui[2*] Xiaoming Liu[3] Ping Tan[1*]
[1] Simon Fraser University [2] State Key Lab of CAD & CG, Zhejiang University
[3] Michigan State University
{ziqianb, pingtan}@sfu.ca, zhpcui@zju.edu.cn, liuxm@cse.msu.edu

To make our paper self-contained, more information is provided in this supplementary material, including more method details (Sec. A), training/testing data consistency (Sec. B.1), additional qualitative results (Sec. B.2), details of the 3D face verification (Sec. B.3) and the regression baselines (Sec. B.4), quantitative photometric errors (Sec. B.5) and limitation analysis (Sec. B.6).

A. Method Details

In this section, we provide additional details of various components in our method. It is better to read together with the corresponding sections of the main paper.

A.1. Image Feature Extraction (Main Paper Sec. 3.1.1)

We use the following strategy to compute the UV space feature F_{uv} from the input images $\{\mathbf{I}_i\}_{i=1}^N$ and initial/intermediate reconstructions $\{\hat{\mathbf{V}}_i^{old}\}_{i=1}^N$. For each image $\mathbf{I}_i$, a feature map is firstly computed by a *Feature Pyramid Network* (FPN) [7]. Then, we unwrap the feature map into UV space based on the initial/intermediate reconstruction $\hat{\mathbf{V}}_i^{old}$. This feature map is then concatenated with the 3-channel UV image that stores the xyz coordinates of $\hat{\mathbf{V}}_i^{old}$, and goes through several ResBlocks [6]. Finally, we forward the N resulting feature maps into max pooling and ResBlocks to get the desired UV feature F_{uv}. Note that we extract 3 different features F_{uv} for neutral shape, expression deformation, and albedo separately.

A.2. Neutral Shape (Main Paper Sec. 3.1.2)

The neural network $\mathcal{F}_{ns}$ is used to compute the neutral shape $\mathbf{V}_{ns}$ from the UV feature map F_{uv} and the neutral shape code $\boldsymbol{\alpha}_{ns}$. More specifically, we first decode the vector $\boldsymbol{\alpha}_{ns}$ into a feature map via a FC-layer and several ResBlocks interleaved nearest upsampling, then the resulting feature map is concatenated with F_{uv} to be further decoded into the neutral shape $\mathbf{V}_{ns}$ via ResBlocks.

*Corresponding authors

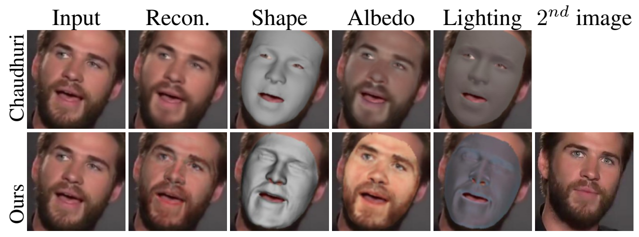

Figure A. Qualitative comparison with Chaudhuri *et al.* [3].

A.3. Expression Deformation (Main Paper Sec. 3.1.3)

As described in Sec. 3.1.3 of the main paper, 3 subnetworks ($\mathcal{F}_{exp}$, $\mathcal{F}_{exp\_mlp}$, and $\mathcal{F}_{exp\_cnn}$) are used to compute the expression deformation $\mathbf{D}_{exp}$ from the UV feature map F_{uv}, the expression code $\boldsymbol{\alpha}_{exp}$, and the expression parameter $\boldsymbol{\beta}$. More specifically, $\mathcal{F}_{exp}$ is a CNN structure similar to $\mathcal{F}_{ns}$ that decodes $\boldsymbol{\alpha}_{exp}$ and F_{uv} into a tensor $\theta_{mlp} \in \mathbb{R}^{H\times W\times(C_{\boldsymbol{\beta}}\times C_0+C_0\times C_1)}$ (*i.e.* the spatially variant weights of the 2-layer MLP $\mathcal{F}_{exp\_mlp}$), where H and W are the spatial dimensions of the UV space while $\{C_{\boldsymbol{\beta}}, C_0, C_1\}$ are the channel sizes of the expression parameter $\boldsymbol{\beta}$, the hidden layer of $\mathcal{F}_{exp\_mlp}$, and the output of $\mathcal{F}_{exp\_mlp}$ respectively. Then the 2-layer MLP $\mathcal{F}_{exp\_mlp}$ with spatially variant weights θ_{mlp} decodes the expression parameter $\boldsymbol{\beta} \in \mathbb{R}^{C_{\boldsymbol{\beta}}}$ to a feature map with size $H\times W\times C_1$, which is further decoded by the CNN $\mathcal{F}_{exp\_cnn}$ to the final expression deformation $\mathbf{D}_{exp}$. Note that we do not personalize the weights of $\mathcal{F}_{exp\_cnn}$ in order not to exceed the memory limitation.

For level 1 in the 3-level scheme, we first convert the UV feature F_{uv}^1 into a vector with several convolution blocks, then concatenate the vector with the expression code $\boldsymbol{\alpha}_{exp}^1$, and feed the concatenated vector into a MLP to obtain the conventional (*i.e.* spatially invariant) weights $\theta_{mlp}^1 \in \mathbb{R}^{C_{\boldsymbol{\beta}}\times C_0+C_0\times C_1}$.

A.4. Detailed Loss Definitions (Main Paper Sec. 3.3)

L_{pose} is a pose-aware loss that supervises the per-image reconstruction, where we have two terms $L_{pose} = L_{dep\_v} +$

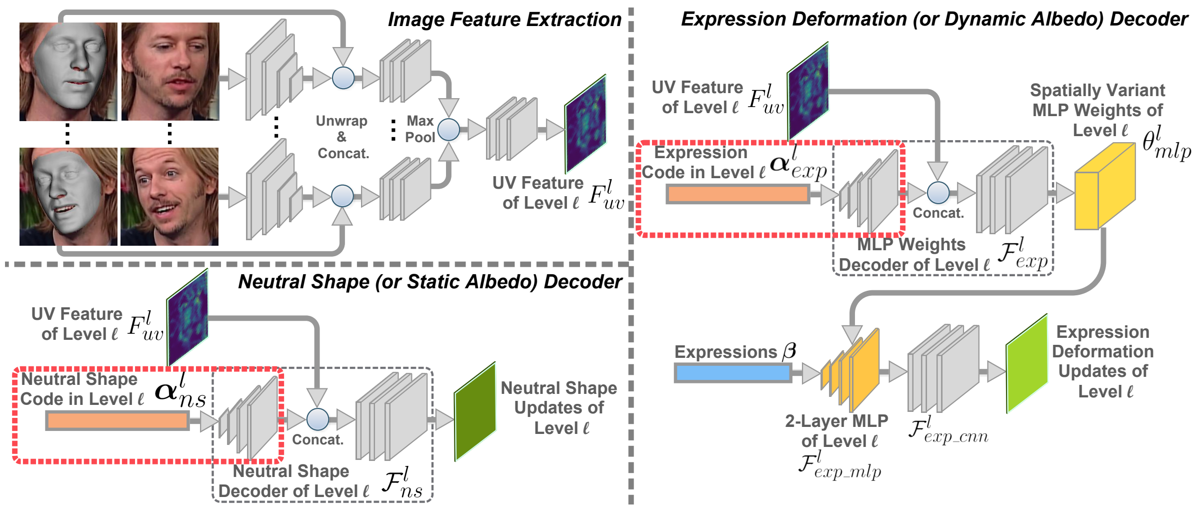
Image Feature Extraction
Unwrap & Concat.
Max Pool
UV Feature of Level ℓ F^l_{uv}
Expression Deformation (or Dynamic Albedo) Decoder
UV Feature of Level ℓ F^l_{uv}
Expression Code in Level ℓ α^l_{exp}
Concat.
MLP Weights Decoder of Level ℓ $\mathcal{F}^l_{exp}$
Spatially Variant MLP Weights of Level ℓ θ^l_{mlp}
Expressions β
2-Layer MLP of Level ℓ $\mathcal{F}^l_{exp\_mlp}$
$\mathcal{F}^l_{exp\_cnn}$
Expression Deformation Updates of Level ℓ
Neutral Shape (or Static Albedo) Decoder
UV Feature of Level ℓ F^l_{uv}
Neutral Shape Code in Level ℓ α^l_{ns}
Concat.
Neutral Shape Decoder of Level ℓ $\mathcal{F}^l_{ns}$
Neutral Shape Updates of Level ℓ

Figure B. The illustration of modifications in the regression baselines (Sec. B.4). Components inside the Red boxes are removed.

$0.025L_{lan}$. In the depth-aligned vertex loss L_{dep_v}, we first align the ground truth scan to the prediction in depth dimension since we do not estimate depth translation in our weak perspective camera model. Following [2], we perform the depth alignment by adding the mean depth difference to the ground truth, then compute the L_2 distances of corresponding points between the prediction and the ground truth $L_{dep\_v} = \sum_i \sum_k \|\mathbf{v}_i^{gt,k} - \hat{\mathbf{v}}_i^k\|_2^2$ for all iterations and all levels. For the landmark loss L_{lan}, we adopt the same formulation as in [2], which is a standard re-projection error. We use the 2D locations of all landmarks from a 3D detector (*i.e.* first 2 dimensions), and dynamic landmarks from a 2D detector, as supervisions. We use different weights for different landmarks. For the landmarks of contour, eyebrow, and mouth, we use weight 10, while for others (*i.e.* eye, nose, and dynamic landmarks) we use weight 1.

L_{recon_geo} is a geometry loss supervising the per-image reconstruction with two terms $L_{recon\_geo} = L_{den\_v} + 1000L_{norm}$. We first rigidly align the prediction to the ground truth using dense correspondences. Then, the dense-aligned vertex loss L_{den_v} with the same form as L_{dep_v} and the normal loss $L_{norm} = \sum_i \sum_k (1 - \cos(\mathbf{n}_i^{gt,k}, \hat{\mathbf{n}}_i^k))$ are computed.

A.5. Implementation Details

Due to the incorrectness of the oversimplified image formulation and the memory limitation, we prevent the appearance relate energy and loss from influencing the shape estimation. More specifically, the photo-metric reconstruction energy $\mathbf{E}_{pho}$ only updates the albedo code $\boldsymbol{\alpha}_{alb}$, and the photo-metric loss L_{recon_pho} only trains the albedo related networks.

B. Experiments

B.1. Training/Testing Data Consistency

All testing data is the same for all methods (Bosphorus from [2]; BU3DFE from [9]; NoW from [8]). [3, 8, 9] are self- or 2D-supervise methods trained on in-the-wild images. [10] is a 3DMM regression method thus trained with pre-fitted 3DMM data. The differences on training data against us are due to the differences in the *methodology*. [4] is trained on pre-fitted 3DMM data, while ours is trained with scans—a trade-off between data size and quality.

B.2. Additional Qualitative Results

We provide more qualitative results for per-image and video reconstructions as well as video retargeting.

Per-image Reconstruction. Fig. A shows the comparison with Chaudhuri *et al.* [3], where we get better geometries with more medium level details while having comparable textures. In Fig. D and Fig. E, we show comparisons with Tewari *et al.* [9] and Bai *et al.* [2]. Our method produces more faithful shapes than Tewari *et al.* [9] and Bai *et al.* [2] and higher resolution textures than Tewari *et al.* [9], though Tewari *et al.* [9] achieves better albedo-illumination disentanglement.

Video Reconstruction and Retargeting. For video reconstructions, we adopt the following strategy. Initially, we uniformly select 5 frames from the video sequence and cache them. Given an incoming frame, we perform reconstruction using this frame together with the cached 5 frames (*i.e.* 6 frames in total). Finally, the cached 5 frames are updated to cover as large yaw angle range as possible. More specifically, we first sort the 6 frames with estimated yaw angles. Then we discard the frame that has the smallest yaw angle difference with its neighbor (won't discard the first frame or the last one), and treat the rests as the updated 5 cached frames. The estimated per-image parameters (*i.e.* expressions, poses, and illuminations) are used for video retargeting. The supplementary video can be found at https://youtu.be/vs7Kyv5rGas.

Results on YouTube clips and videos from Bai *et al.* [2] and Chaudhuri *et al.* [3] are included. On YouTube clips, our method achieves faithful reconstructions and reasonable retargeting results to various subjects. Compared with Bai *et al.* [2], our method generates more stable reconstructions and additionally supports retargeting. Compared with Chaudhuri *et al.* [3], our method has superior shape quality that better reflects the personal characteristics, such as the round chin instead of the sharp one from [3] and the shape of the month, and achieves reasonable expression transfer results.

Originally, we planed to have a user study to quantitatively compare video retargeting results with Chaudhuri *et al.* [3], but we only have a demo video of [3] that is not enough for a user study. As the code of [3] is not publicly available, we contacted with the authors of [3]. However, we were not able to get additional results at the end.

B.3. 3D Face Verification Details

The 3D face verification network is a ResNet34 [6], which takes in the UV representation of vertex positions and normals of the neutral shape (i.e. 6 channels) and outputs an embedding. The network is trained with the contrastive loss [5] on the LFW training spilt under Restricted Configuration. We only train the first Conv & BatchNorm layer and the last FC layer, while using weights pre-trained on ImageNet for all other layers. We also augment the input neutral shape with a random small rotation (*i.e.* Euler angles sampled from $[-7.5°, 7.5°]$) during training for better robustness.

B.4. Regression Baseline Details

To demonstrate the effectiveness of explicit optimization, we design regression baselines to compare with, where the components of the face rig are directly predicted by the neural network instead of being optimized. More specifically, we remove different parts of the rig code $\boldsymbol{\alpha}_{ns}$, $\boldsymbol{\alpha}_{exp}$ from the decoding process (*i.e.* red boxes in Fig. B). Thus, the neutral shape or the network weights θ_{mlp} of the MLP are directly regressed without explicit optimization. For the neutral shape updates in level 1, we additionally regress the 3DMM coefficients from the UV feature map F_{uv}.

B.5. Photometric Errors

We test photometric errors on 220 images selected by [9] (on its website) from VoxCeleb2. As in Tab. A, although our method is trained on limited rendered images augmented with synthetic degradation, it performs on-par with the SOTA face modeling method [9] that is trained on vast in-the-wild data. More discussions about the limitation in texture quality on in-the-wild data can be found in Sec. B.6.

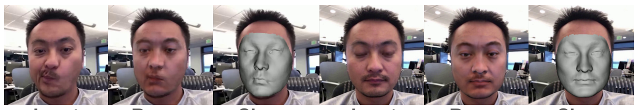

Figure C. Failure cases of our method.

Table A. Photometric errors. Color value range: [0, 255).

	$L_{2,1}$ norm ↓			PSNR ↑		
Methods	Ours	Ours(R)	[38]	Ours	Ours(R)	[38]
Mean	14.81	14.12	14.05	29.66	30.06	30.10
STD	4.58	4.43	3.46	2.53	2.65	2.22

B.6. Limitation

Though our method achieves good reconstruction quality and reasonable retargeting results, we still observe some limitations on unusual expressions, eyelid motions, and the amplitude of transferred expressions. Currently our model cannot capture unusual expressions and eyelid motions well as in Fig. C, which could be due to the lack of training data since we use the Stirling/ESRC 3D face database [1] for training where only 8 expressions are included without unusual expressions and eyelid motions. Also, for some expressions (*e.g.* the "frown" expression in 1:28 of the supplementary video), the amplitude of the transferred expression is slightly smaller than the source video, which could be due to the fact that the current space of the expression parameter $\boldsymbol{\beta}$ is automatically learned and not explicitly defined, such as blendshape coefficients. We leave this issue to future works.

Since our model is trained with rendered images (augmented with synthetic degradation for *Ours(R)* version), its generalization ability is not perfect when applied on in-the-wild images, resulting in artifacts on textures (*e.g.* making the face look dirty), which is also mentioned in Main Paper Sec.4.1. We believe our model could be improved with more realistically rendered training data and/or self-supervise learning directly on in-the-wild images.

References

[1] *Stirling/ESRC 3D face database*. 3

[2] Ziqian Bai, Zhaopeng Cui, Jamal Ahmed Rahim, Xiaoming Liu, and Ping Tan. Deep facial non-rigid multi-view stereo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5850–5860, 2020. 2, 3, 4, 5

[3] Bindita Chaudhuri, Noranart Vesdapunt, Linda Shapiro, and Baoyuan Wang. Personalized face modeling for improved face reconstruction and motion retargeting. In *Eur. Conf. Comput. Vis.*, pages 142–160, 2020. 1, 2, 3

[4] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Eur. Conf. Comput. Vis.*, pages 534–551, 2018. 2

[5] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 2, pages 1735–1742, 2006. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 1, 3

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2117–2125, 2017. 1

[8] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7763–7772, 2019. 2

[9] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Fml: face model learning from videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10812–10822, 2019. 2, 3, 4, 5

[10] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5163–5172, 2017. 2

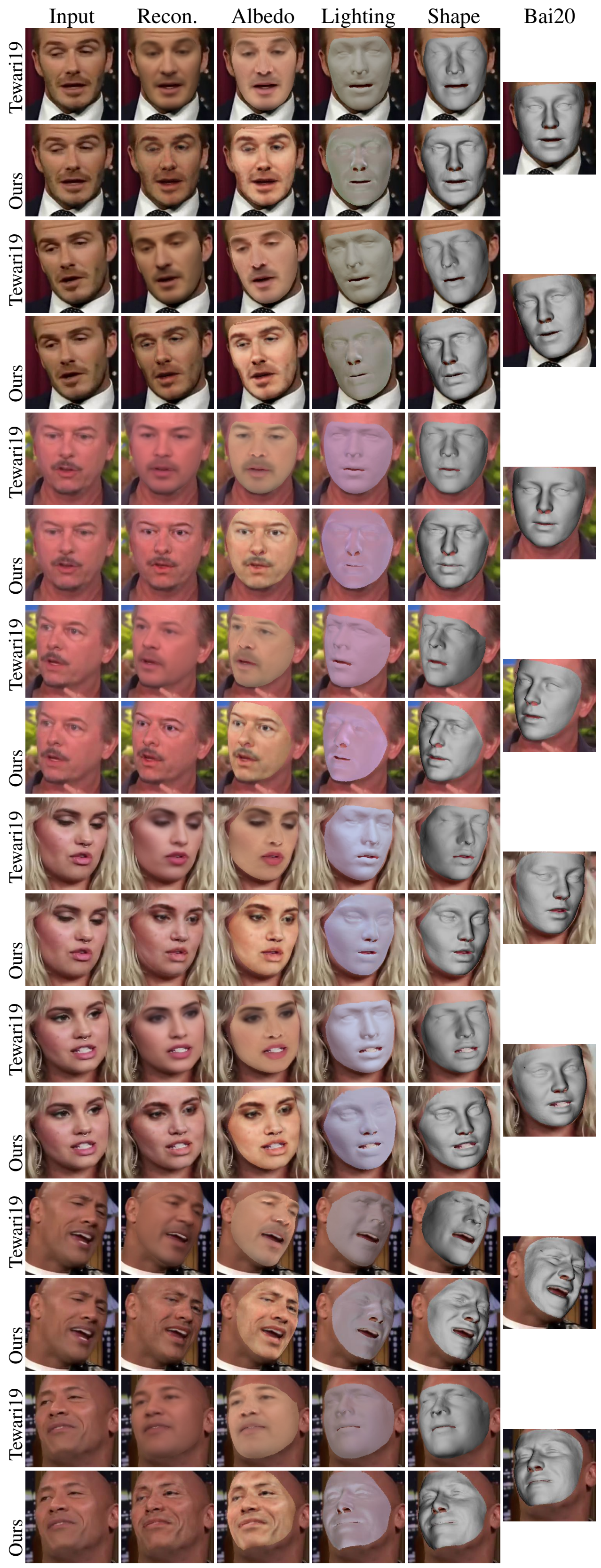

Figure D. Qualitative comparison with Tewari *et al*. [9] and Bai *et al*. [2].

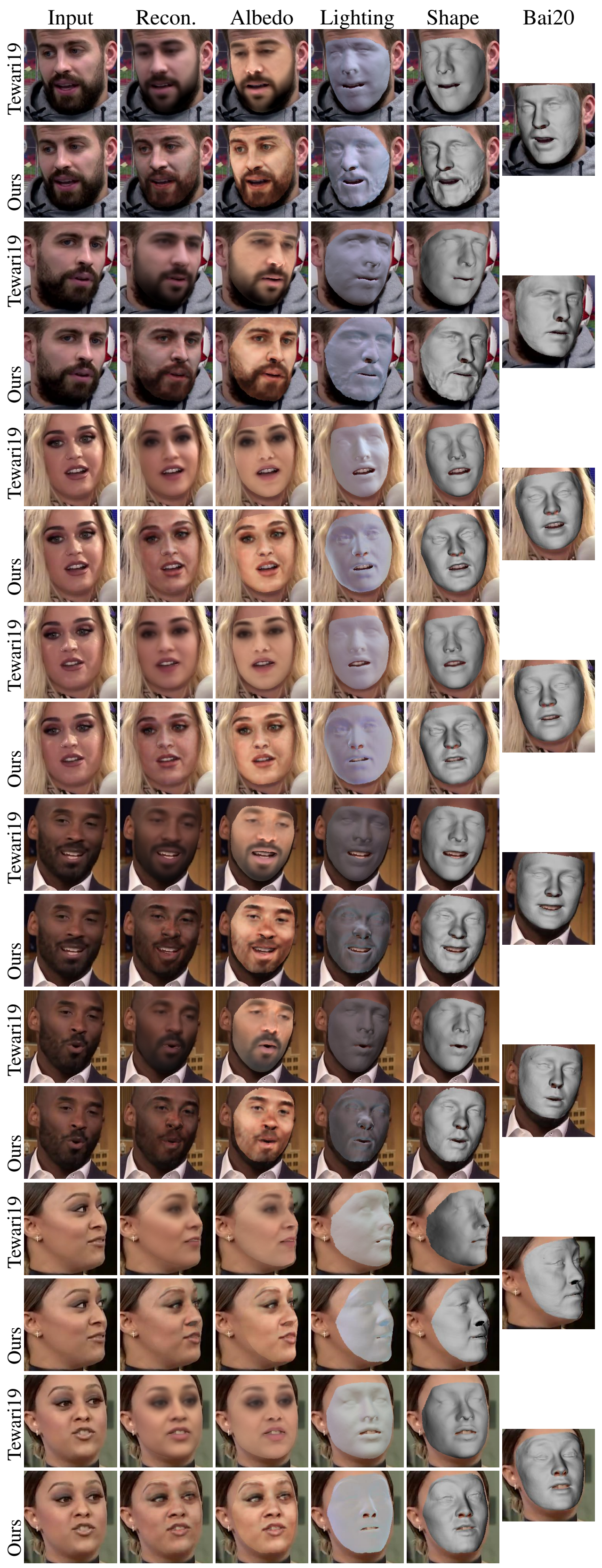

Figure E. Qualitative comparison with Tewari *et al*. [9] and Bai *et al*. [2].