Tencent
AI Lab

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

FFHQ-UV: Normalized Facial UV-Texture Dataset for 3D Face Reconstruction

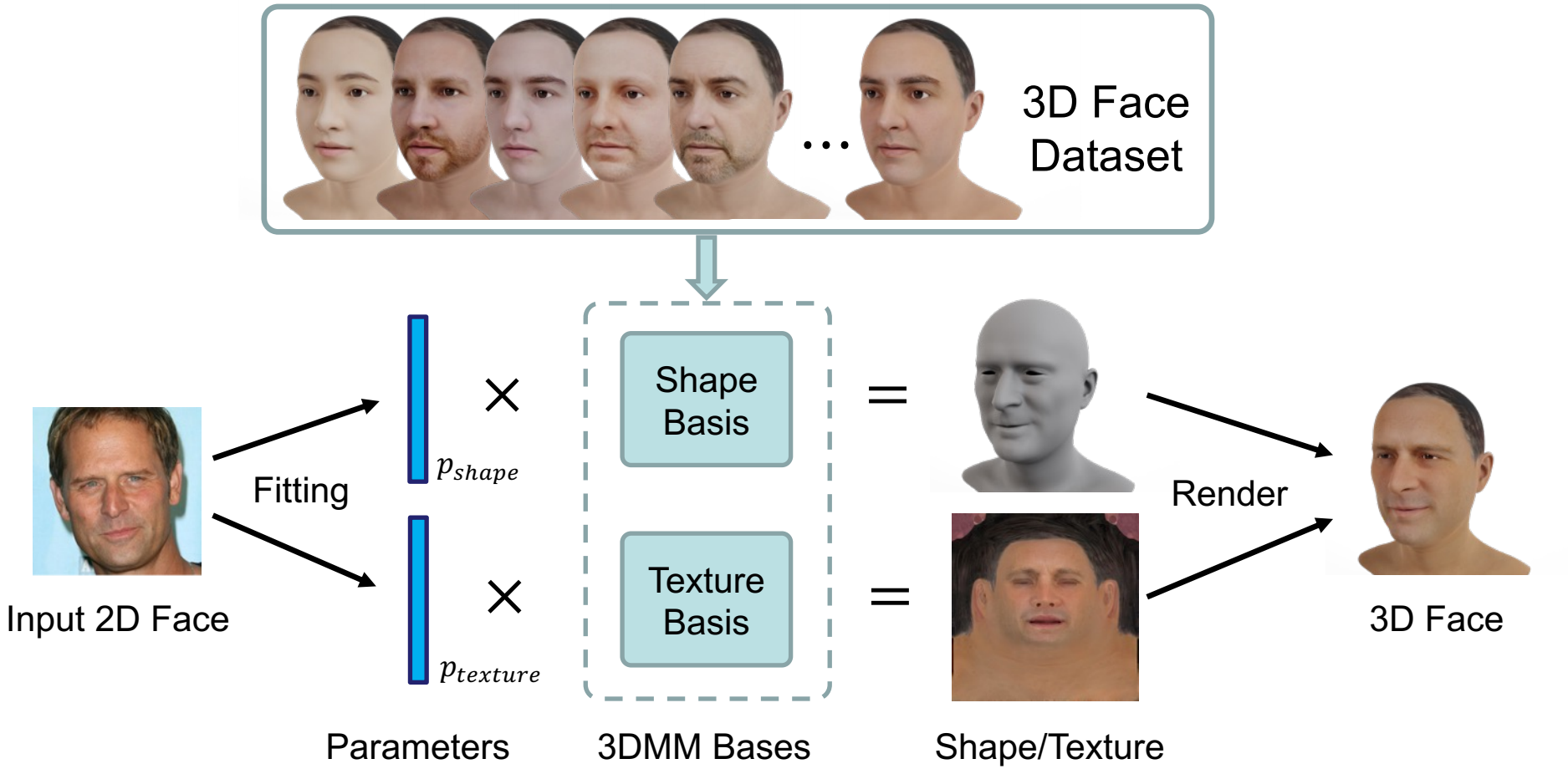
Poster: TUE-AM-035

Haoran Bai[1], Di Kang[2], Haoxian Zhang[2], Jinshan Pan[1], Linchao Bao[2]

[1]Nanjing University of Science and Technology
[2]Tencent AI Lab

3D face reconstruction based on 3DMM

- The reconstruction can be achieved by parametric fitting with the 3DMM.
- 3D face dataset is critical for constructing strong 3DMM bases.



Challenges in texture map recovery

- Two key aspects:
  - *fidelity*: require higher diversity for large expressive capacities
  - *quality*: require even illumination and clean facial region
- Few existing facial texture datasets are of sufficient diversity and high-quality to satisfy both of above aspects.

Datasets	# samples	Resolution	Even illumination	Public available
LSFM*	10,000	512×512	✗	✗
AvatarMe*	200	6144×4096	✓	✗
HiFi3DFace*	200	2048×2048	✓	✗
Facescape*	847	4096×4096	✓	✓
WildUV	5638	377×595	✗	✗
NormAvatar	5601	256×256	✓	✗
FFHQ-UV (Ours)	54,165	1024×1024	✓	✓

"*" denotes the dataset which is captured under controlled conditions.

Our contribution

- We create a normalized UV-texture dataset
  - derive from in-the-wild face images, cheap and diverse
  - even illuminations, neutral expressions, and clean facial regions

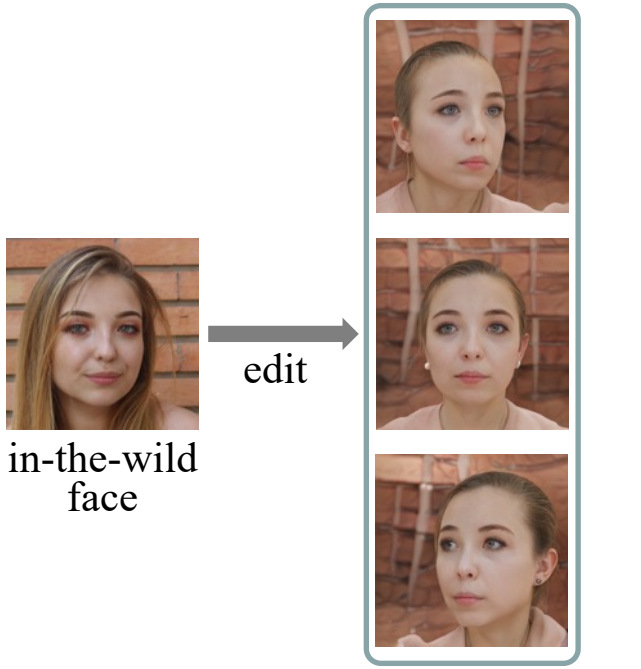
in-the-wild faces

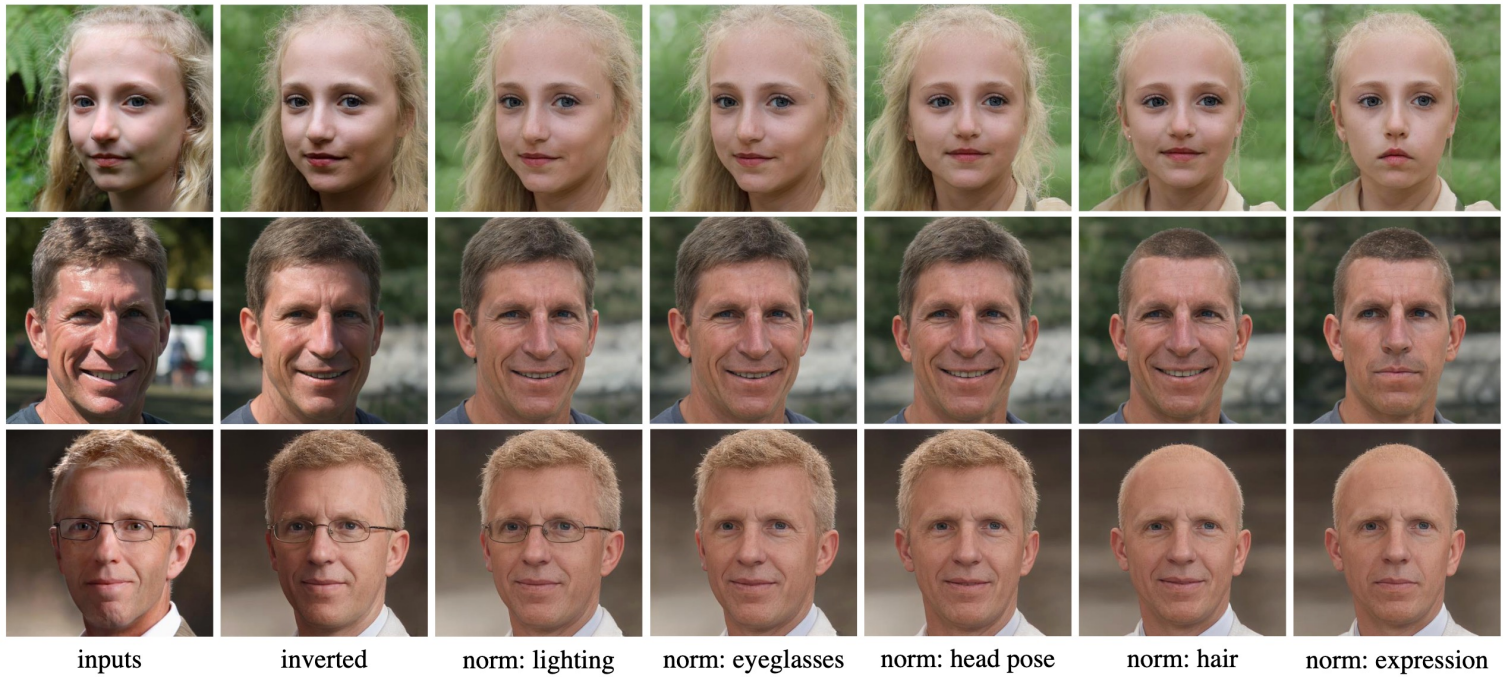
normalized faces

normalized UV-texture maps

Dataset creation pipeline

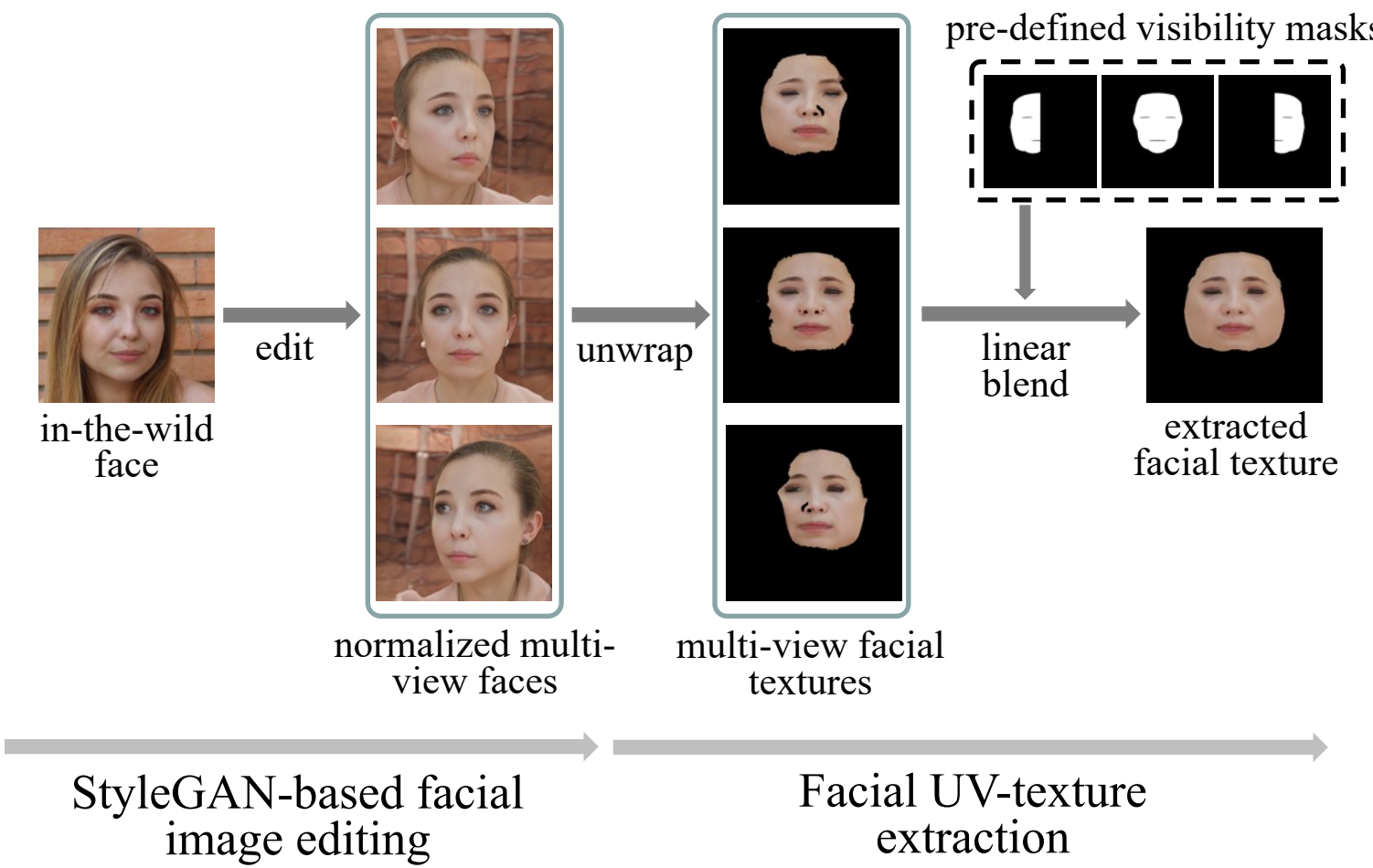
- StyleGAN-based facial image editing
  - generate multi-view normalized faces from single-view in-the-wild faces

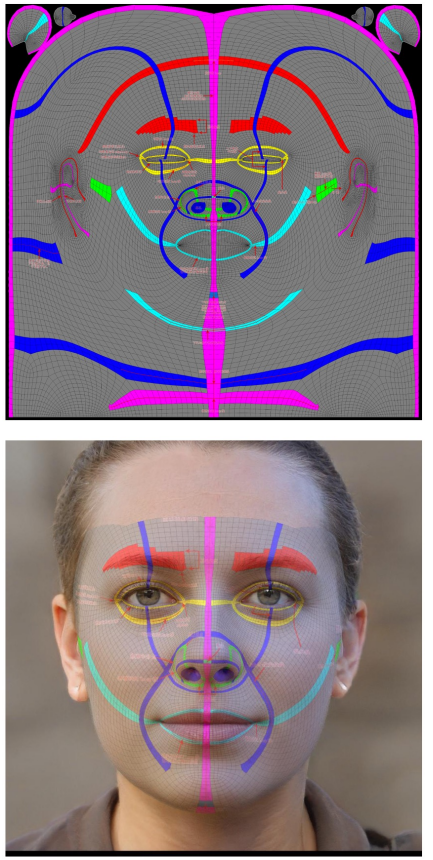




Dataset creation pipeline

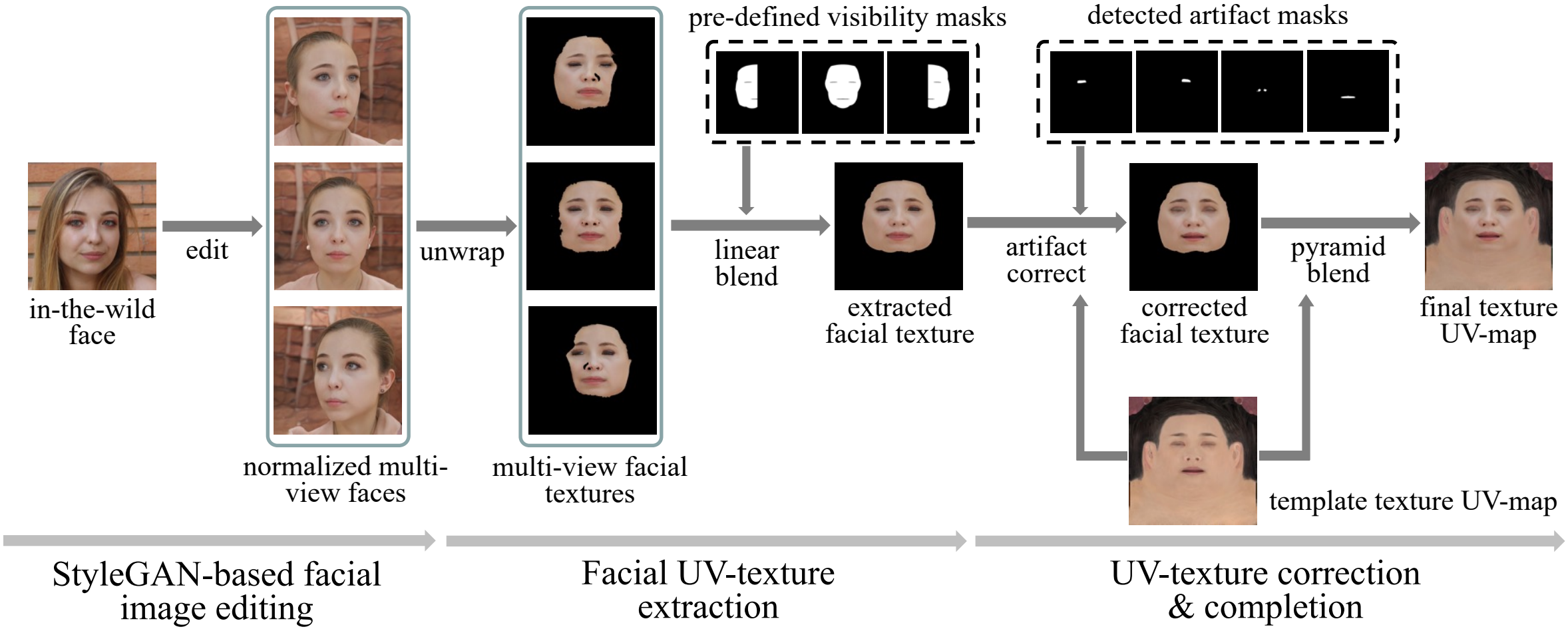
- Facial UV-texture extraction
  - extract multi-view facial textures and blend them together





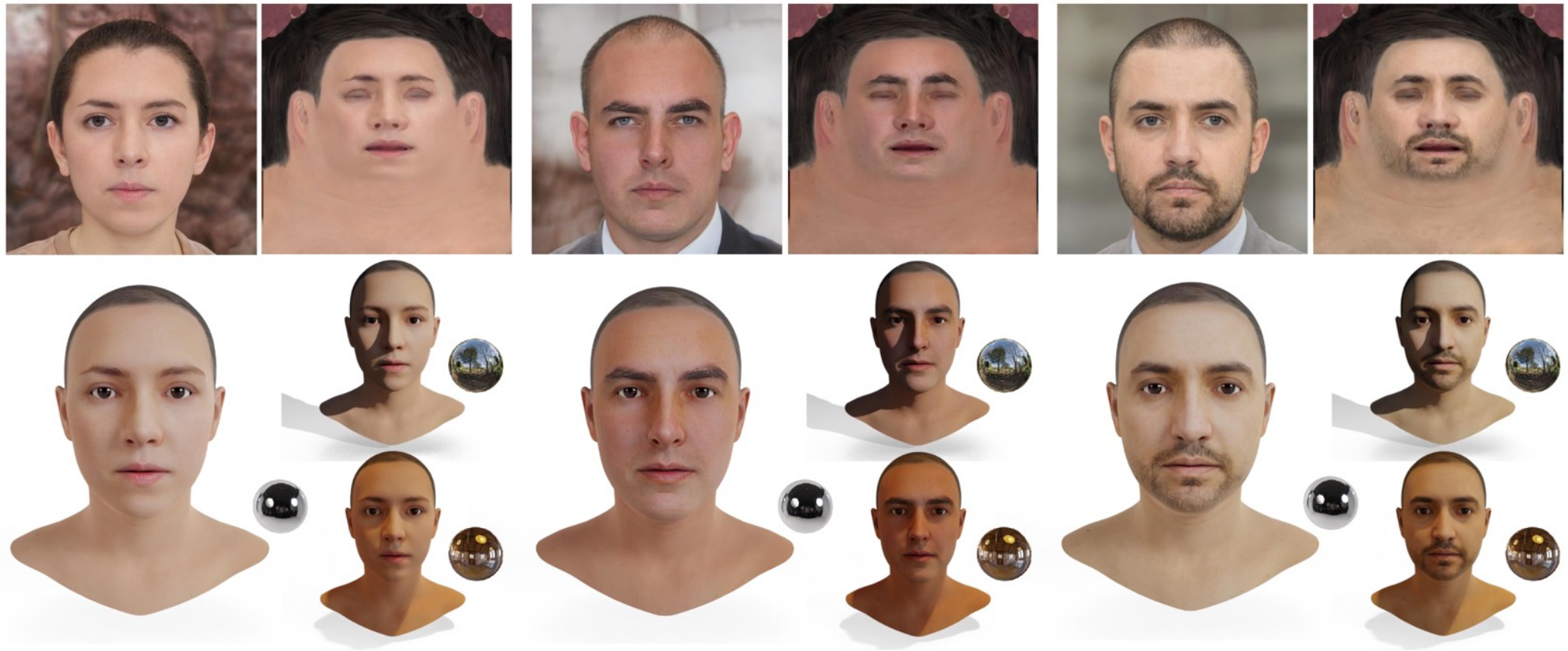
Dataset creation pipeline

- UV-texture correction & completion
  - correct artifacts and generate complete UV-texture map



FFHQ-UV dataset

- Based on FFHQ dataset, which contains diverse in-the-wild faces.
- Over $50{,}000$ UV-texture maps at 1024×1024 resolution.
- High-quality and evenly illuminated, ready for realistic renderings.



Is the FFHQ-UV diverse?

- ID diversity of face datasets
  - FFHQ-UV inherits the most ID variations in FFHQ (over 90%).
  - FFHQ-UV has a higher ID standard deviation value compared to Facescape, indicating that FFHQ-UV is more diverse.

Datasets	**FFHQ**	**FFHQ-Inv**	**FFHQ-Norm**	**FFHQ-UV***	**Facescape***
ID std.	100%	91.86%	90.06%	90.01%	84.24%

"*" indicates that ID features are extracted from rendered face images.

Does the FFHQ-UV evenly illuminated?

- We present Brightness Symmetry Error (BS Error) to measure the illumination of UV maps.

$$BS\ Error = \|\mathcal{B}_\alpha(T^Y) - \mathcal{F}_h(\mathcal{B}_\alpha(T^Y))\|_1$$

T^Y: the Y channel of T in YUV space;
$\mathcal{B}_\alpha$: the Gaussian blurring operation with the kernel size of α;
$\mathcal{F}_h$: the horizontal flip operation

- The StyleGAN-based editing step in our pipeline facilitates even illumination.
- FFHQ-UV is competitive with Facescape (under controlled conditions).

Methods	**Facescape***	**w/o editing**	**FFHQ-UV (Ours)**
BS Error	6.984	11.385	7.293

"*" denotes the dataset which is captured under controlled conditions.

3D face reconstruction with FFHQ-UV

- GAN-based texture decoder
  - we train a GAN-based texture decoder using FFHQ-UV
  - the goal of our UV-texture recovery is to find the latent code z

$$\tilde{T} = \mathcal{G}_{tex}(z)$$

- Three-stage reconstruction algorithm
  1. Linear 3DMM initialization
     - initialize parameters using a trained Deep3D, except texture latent code
  2. Texture latent code z optimization
     - find a latent code z of the texture decoder, while fixing other parameters
  3. Joint parameter optimization
     - jointly optimize all the parameters

Quantitative results on the RELAY benchmark

- Our method outperforms state-of-the-art single-image reconstruction approaches.

Methods	nose	mouth	forehead	cheek	all
3DDFA-v2	1.903	1.597	2.477	1.757	1.926
GANFit	1.928	1.812	2.402	1.329	1.868
MGCNet	1.771	1.417	2.268	1.639	1.774
Deep3D	1.719	1.368	2.015	1.528	1.657
stage 1	1.557	1.661	1.940	1.014	1.543
PCA tex basis	1.904	1.419	1.773	0.982	1.520
w/o multi-view	1.780	1.419	1.711	0.980	1.473
w/ Facescape (scratch)	1.731	1.653	1.711	1.207	1.576
w/ Facescape (finetune)	1.570	1.576	1.581	1.074	1.450
Ours	1.681	1.339	1.631	0.943	**1.399**

Quantitative results on the RELAY benchmark

- Our method outperforms baseline variants: initialization in stage 1, using PCA texture basis, and using dataset without multi-view faces.

Methods	nose	mouth	forehead	cheek	all
3DDFA-v2	1.903	1.597	2.477	1.757	1.926
GANFit	1.928	1.812	2.402	1.329	1.868
MGCNet	1.771	1.417	2.268	1.639	1.774
Deep3D	1.719	1.368	2.015	1.528	1.657
stage 1	1.557	1.661	1.940	1.014	1.543
PCA tex basis	1.904	1.419	1.773	0.982	1.520
w/o multi-view	1.780	1.419	1.711	0.980	1.473
w/ Facescape (scratch)	1.731	1.653	1.711	1.207	1.576
w/ Facescape (finetune)	1.570	1.576	1.581	1.074	1.450
Ours	1.681	1.339	1.631	0.943	**1.399**

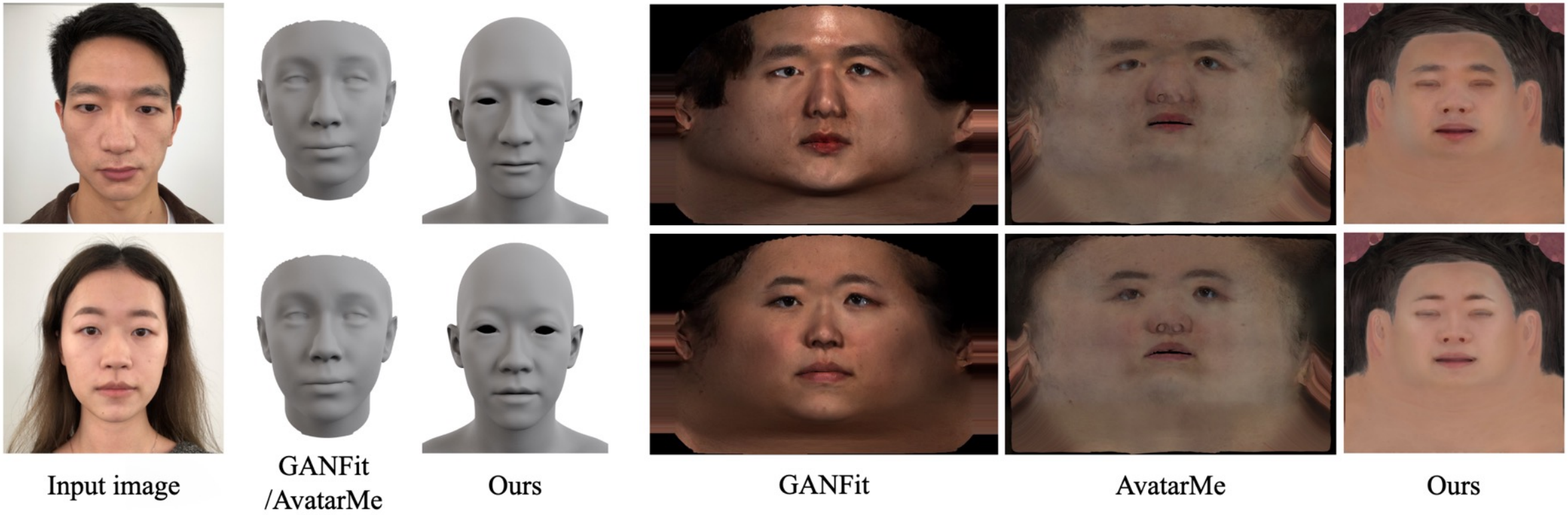
Quantitative results on the RELAY benchmark

- Using the proposed FFHQ-UV dataset outperforms using texture UV-maps from Facescape dataset.

Methods	nose	mouth	forehead	cheek	all
3DDFA-v2	1.903	1.597	2.477	1.757	1.926
GANFit	1.928	1.812	2.402	1.329	1.868
MGCNet	1.771	1.417	2.268	1.639	1.774
Deep3D	1.719	1.368	2.015	1.528	1.657
stage 1	1.557	1.661	1.940	1.014	1.543
PCA tex basis	1.904	1.419	1.773	0.982	1.520
w/o multi-view	1.780	1.419	1.711	0.980	1.473
w/ Facescape (scratch)	1.731	1.653	1.711	1.207	1.576
w/ Facescape (finetune)	1.570	1.576	1.581	1.074	1.450
Ours	1.681	1.339	1.631	0.943	**1.399**

Visual results

- Comparison with the GANFit and AvatarMe methods.
  - our shapes are more faithful to input faces
  - our texture maps are more evenly illuminated and of higher quality



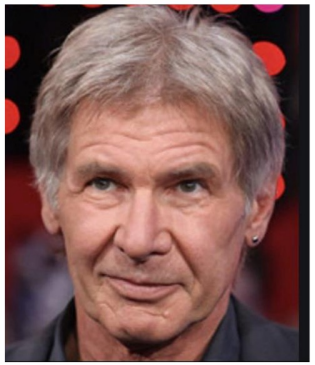
Visual results

- Comparison with the Normalized Avatar method.
  - our results better resemble the input faces


Input image


Normalized Avatar


Ours

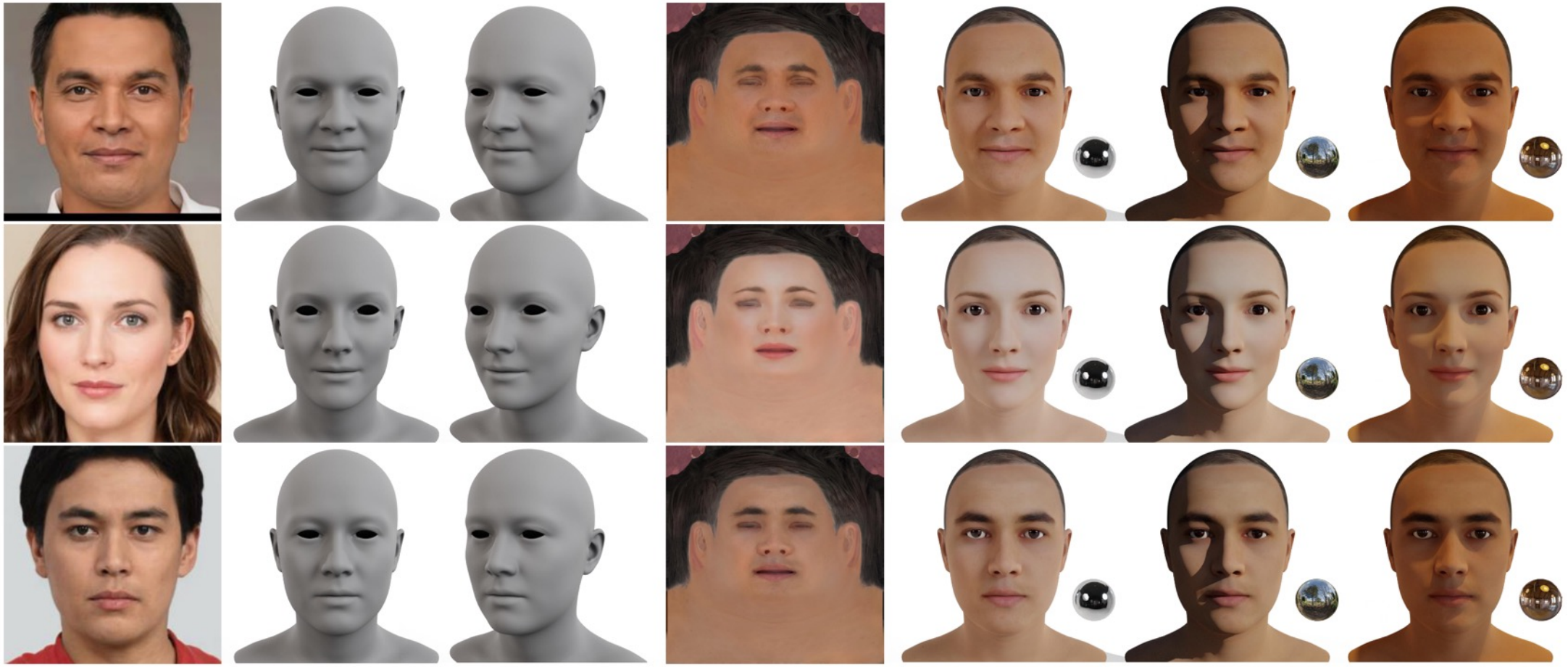

Input image


Normalized Avatar


Ours

Visual results

- Examples of our reconstructed results.
  - our textures can be rendered with different lighting conditions



Thanks

Code Link: https://github.com/csbhr/FFHQ-UV