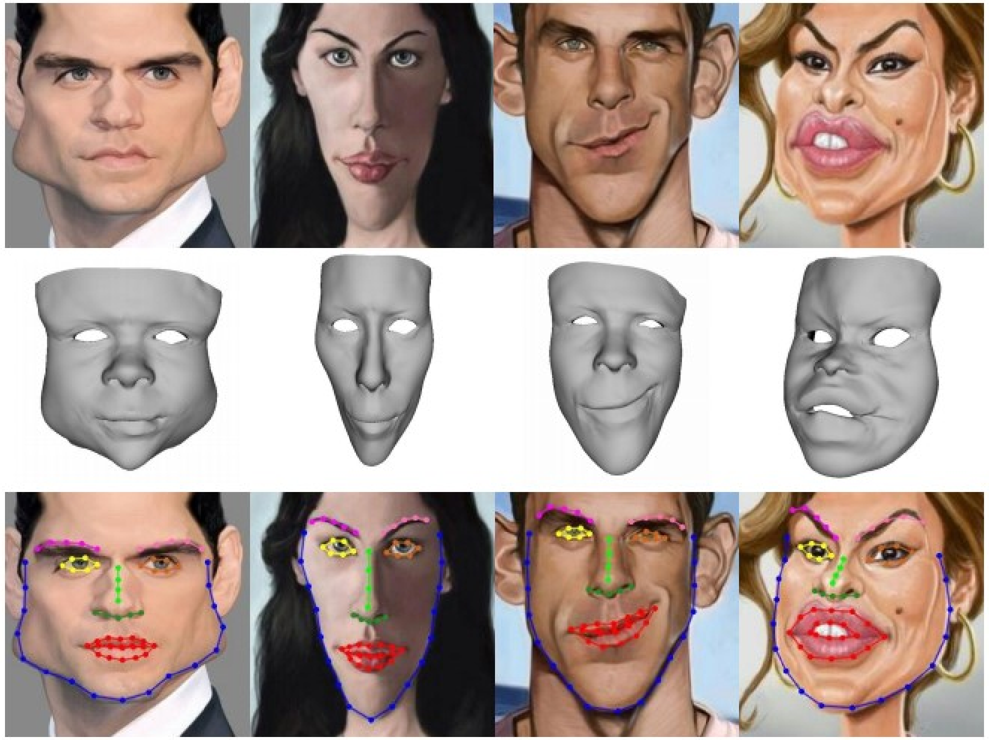
Landmark Detection and 3D Face Reconstruction for Caricature using a Nonlinear Parametric Model

Hongrui Cai[1], Yudong Guo[1], Zhuang Peng[1], Juyong Zhang[1]

[1]University of Science and Technology of China

What is our goal?

- Given a single caricature image (first row), our algorithm generates its **3D model** with orientation (second row) and **corresponding landmarks** (third row).



Why 2D landmarks of caricatures are required?

- Most of caricature related works need facial landmarks to help preprocess the caricatures.

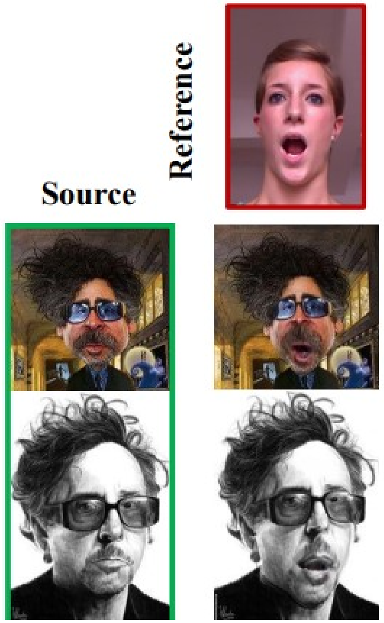
Generation

[Cao et al. 2018]

Reconstruction

[Wu et al. 2018]

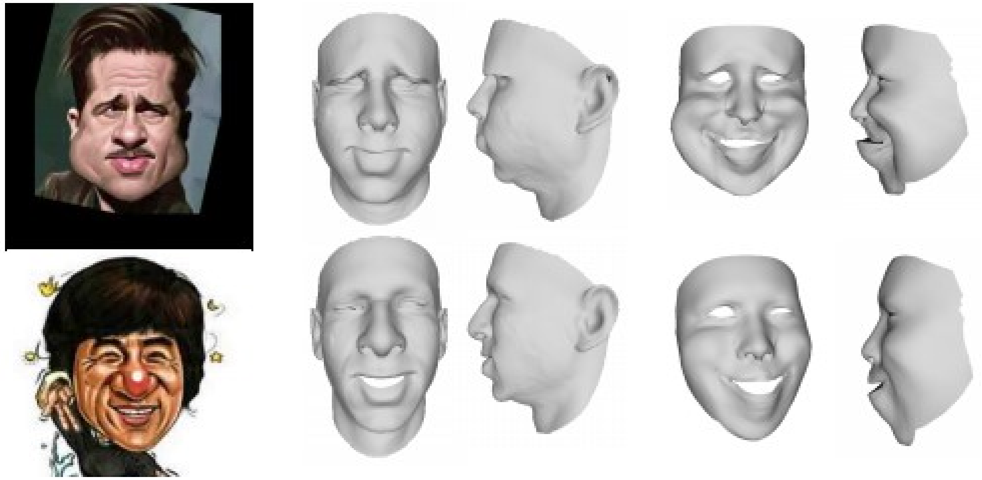
Editing



[Chen et al. 2020]

Challenges

- Abstract and exaggerate patterns
- Large representation varieties
- No real shading information

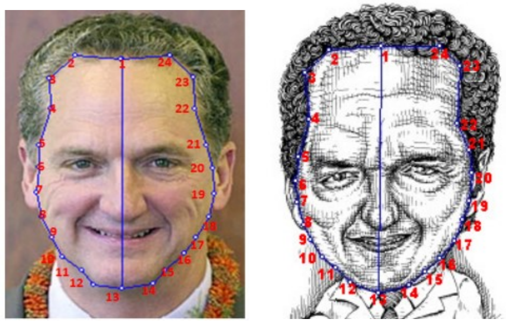


Reconstruction results of traditional methods

Detection results of baselines

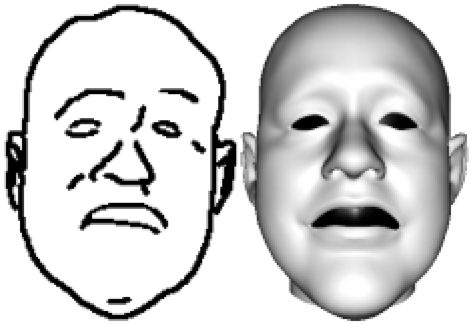
Related work

- Automatic landmark detection for caricatures

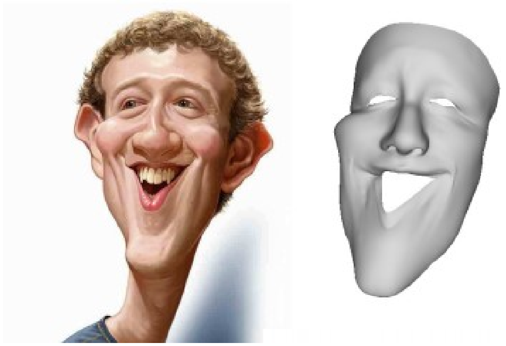


[Sadimon et al. 2015]

- 3D caricature reconstruction



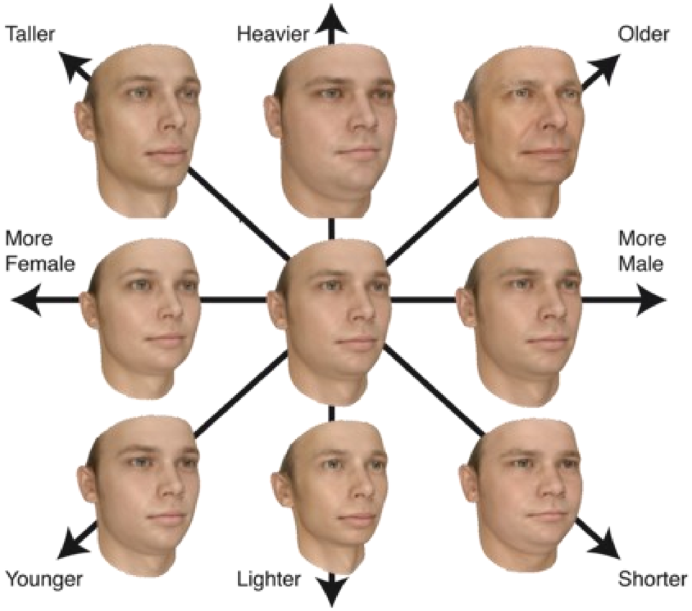
[Han et al. 2017]



[Wu et al. 2018]

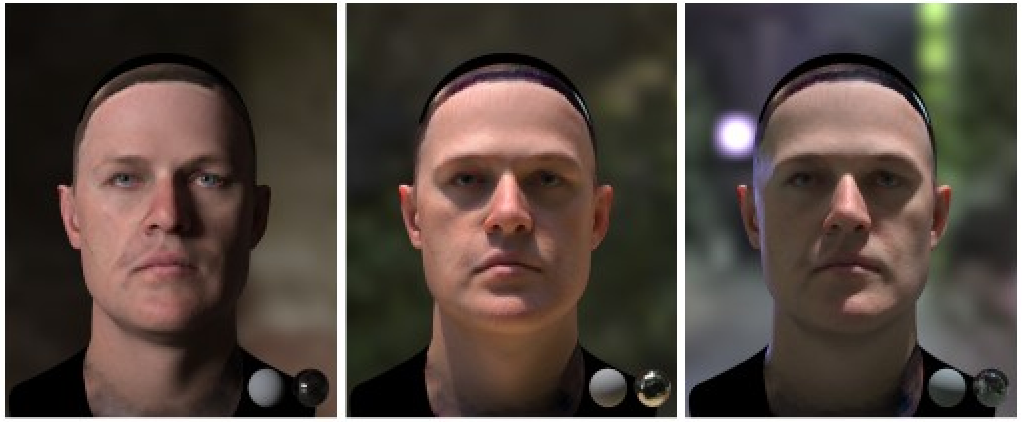
Related work of normal face reconstruction

3DMM



[Blanz et al. 1999]

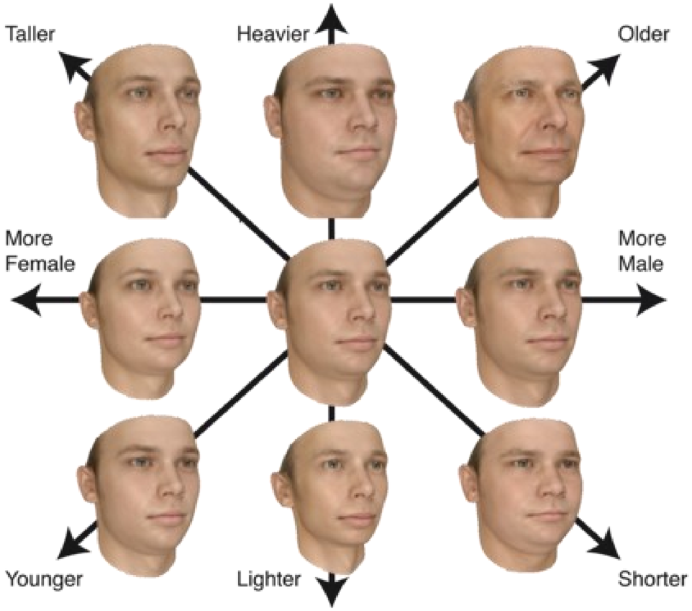
Realistic Modeling



[Li et al. 2020]

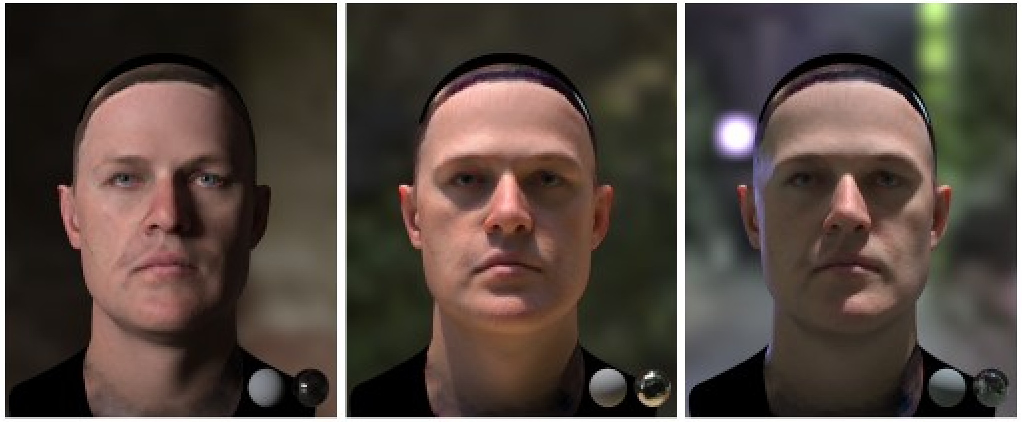
Related work of normal face reconstruction

3DMM



[Blanz et al. 1999]

Realistic Modeling



[Li et al. 2020]

- Could not directly applied to general caricatures

Our solution

- Construct a caricature dataset with 2D images, labeled landmarks, and 3D meshes
- Via a deformation representation, propose a deep learning method to recover 3D shape and weak perspective parameters

Our solution

- Construct a caricature dataset with 2D images, labeled landmarks, and 3D meshes
- Via a deformation representation, propose a deep learning method to recover 3D shape and weak perspective parameters

Existing datasets

- IIIT-CFW [Mishra et al. 2016] Dataset:
  - 8,928 annotated cartoon faces of 100 public figures
  - with additional attributes, such as age group, expression

Existing datasets

- WebCaricature [Huo et al. 2018] Dataset :
  - Images of 6,042 caricatures and 5,974 photos from 252 persons
  - **17 facial landmarks** for each image

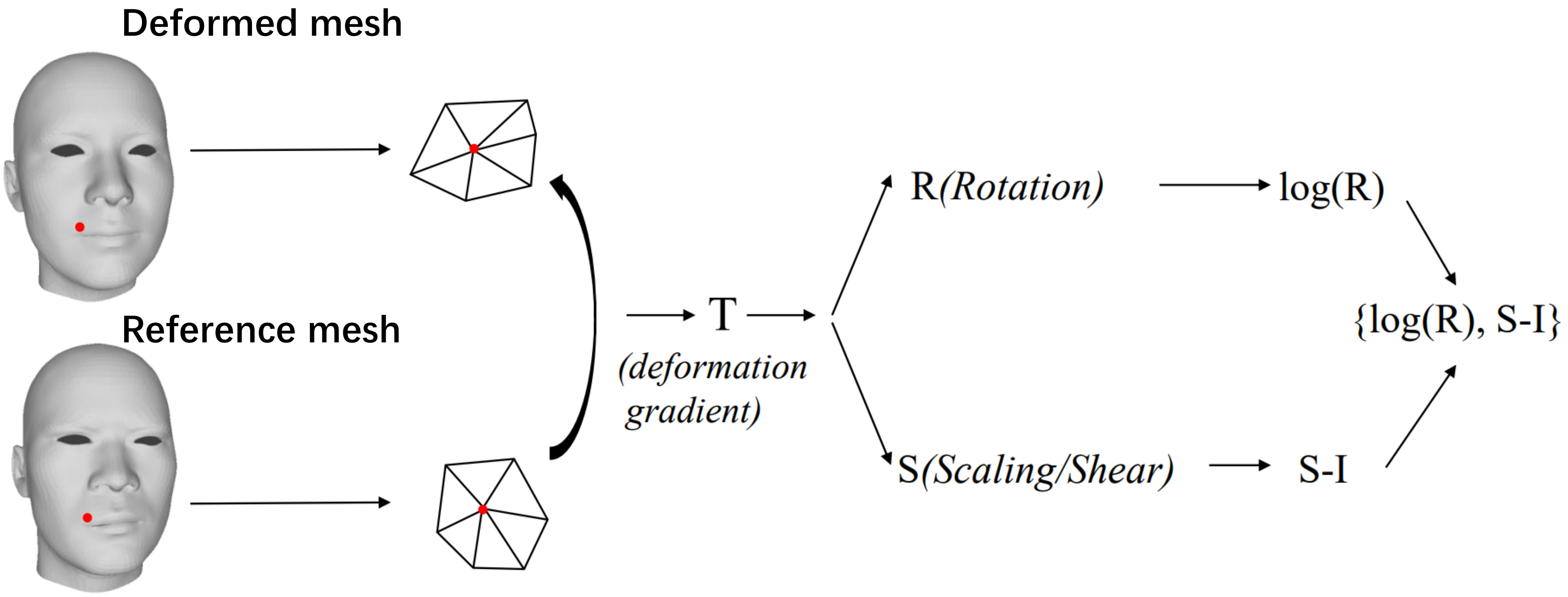
Dataset Construction and Augmentation

- Select nearly 6K caricatures from the Internet, then label 68 landmarks on each image
- Based on CariGANs [Cao et al. 2018] , generate around 2K caricatures and corresponding landmarks
- Adopt an optimization based method [Wu et al. 2018] to recover 3D meshes

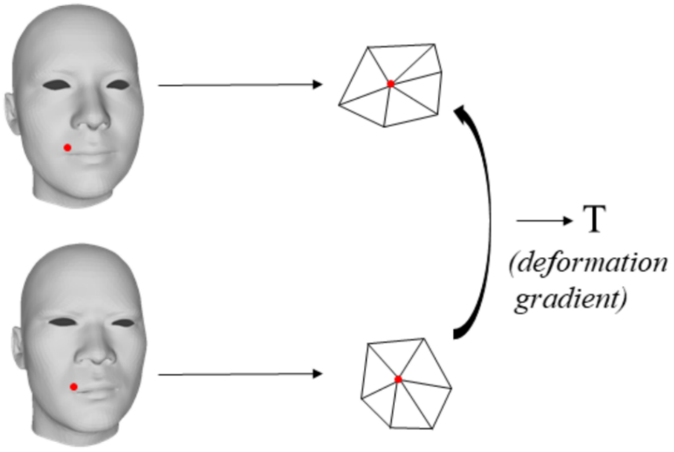
Our solution

- Construct a caricature dataset with 2D images, labeled landmarks, and 3D meshes
- Via a deformation representation, propose a deep learning method to recover 3D shape and weak perspective parameters

Deformation representation [Gao et al. 2019]



Deformation representation [Gao et al. 2019]

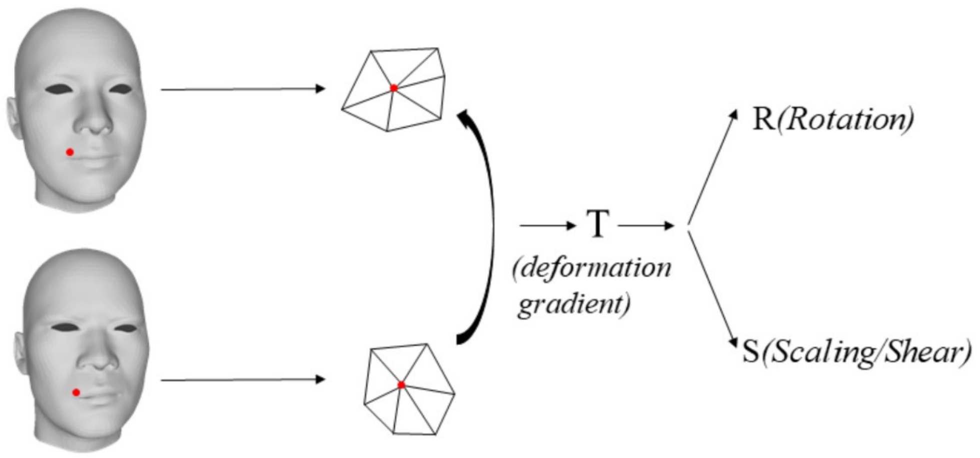


- Compute deformation gradient $\mathbf{T}_i$ of i^{th} vertex with edge weight c_{ij} :

$$\min_{\mathbf{T}_i} \sum_{j \in \mathcal{N}_i} c_{ij} \| (\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j) \|_2^2, \qquad (1)$$

- Polar decomposition of $\mathbf{T}_i$: $\mathbf{T}_i = \mathbf{R}_i \mathbf{S}_i$.
- Logarithm of rotation part $\mathbf{R}_i$. It allow effective linear combination for $\log \mathbf{R}_i$.
- Transformation of scaling / shear part $\mathbf{S}_i$.

Deformation representation [Gao et al. 2019]

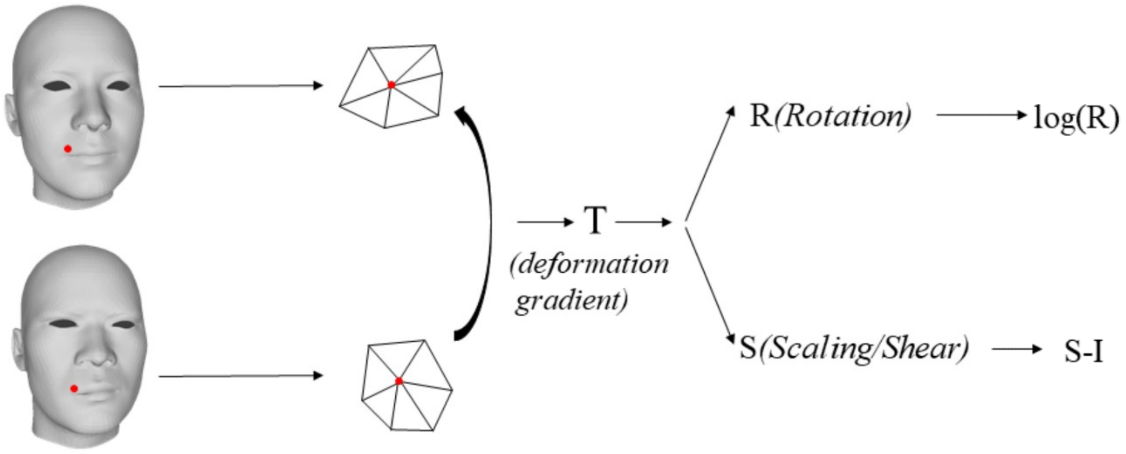


- Compute deformation gradient $\mathbf{T}_i$ of i^{th} vertex with edge weight c_{ij} :

$$\min_{\mathbf{T}_i} \sum_{j \in \mathcal{N}_i} c_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j)\|_2^2, \qquad (1)$$

- Polar decomposition of $\mathbf{T}_i$: $\mathbf{T}_i = \mathbf{R}_i \mathbf{S}_i$.
- Logarithm of rotation part $\mathbf{R}_i$. It allow effective linear combination for $\log \mathbf{R}_i$.
- Transformation of scaling / shear part $\mathbf{S}_i$.

Deformation representation [Gao et al. 2019]

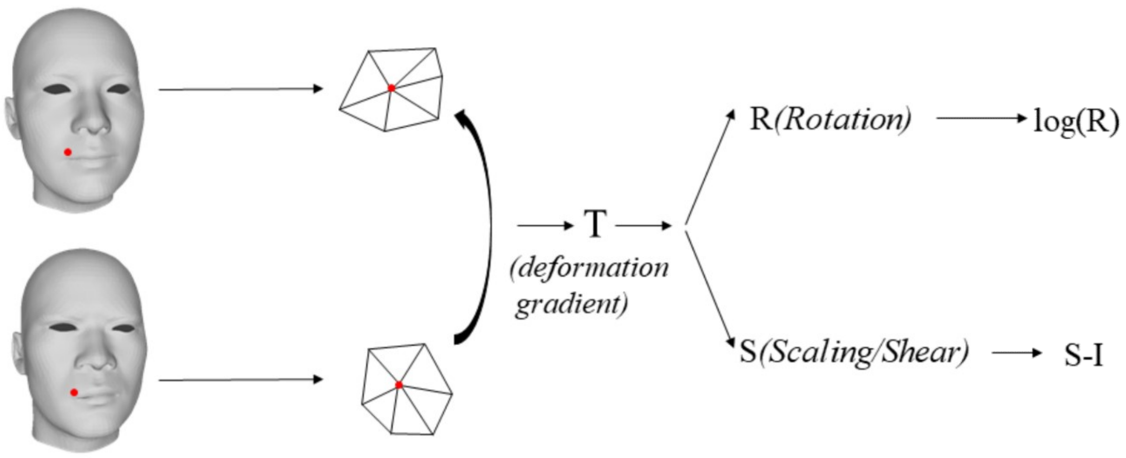


- Compute deformation gradient $\mathbf{T}_i$ of i^{th} vertex with edge weight c_{ij} :

$$\min_{\mathbf{T}_i} \sum_{j \in \mathcal{N}_i} c_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j)\|_2^2, \qquad (1)$$

- Polar decomposition of $\mathbf{T}_i$: $\mathbf{T}_i = \mathbf{R}_i \mathbf{S}_i$.
- Logarithm of rotation part $\mathbf{R}_i$. It allow effective linear combination for $\log \mathbf{R}_i$.
- Transformation of scaling / shear part $\mathbf{S}_i$.

Deformation representation [Gao et al. 2019]



- The deformation representation of i^{th} vertex:
    - matrix form $\{\log \mathbf{R}_i, \mathbf{S}_i - \mathbf{I}\}$
    - vector form $[\mathbf{r}_i, \mathbf{s}_i] \in \mathbb{R}^9$

Deformation base

- Based on a reference mesh and n deformed meshes, build a linear combination of deformation representations:

$$\mathbf{T}_i(\mathbf{w}) = \exp\left(\sum_{l=1}^{n} w_{R,l} \log \mathbf{R}_i^l\right)\left(\mathbf{I} + \sum_{l=1}^{n} w_{S,l}(\mathbf{S}_i^l - \mathbf{I})\right), \quad (4)$$

- Compute the Jacobian matrix $\partial \mathbf{T}_i(\mathbf{w})/\partial \mathbf{w}$, then use the Levenberg-Marquardt algorithm to solve:

$$\min_{\mathbf{w}} \sum_{v_i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} c_{ij} \|(\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{T}_i(\mathbf{w})(\mathbf{p}_i - \mathbf{p}_j)\|^2, \quad (5)$$

Deformation base

- Based on a reference mesh and n deformed meshes, build a linear combination of deformation representations:

$$\mathbf{T}_i(\mathbf{w}) = \exp(\sum_{l=1}^{n} w_{R,l} \log \mathbf{R}_i^l)(\mathbf{I} + \sum_{l=1}^{n} w_{S,l}(\mathbf{S}_i^l - \mathbf{I})), \quad (4)$$

- Compute the Jacobian matrix $\partial \mathbf{T}_i(\mathbf{w})/\partial \mathbf{w}$, then use the Levenberg-Marquardt algorithm to solve:

$$\min_{\mathbf{w}} \sum_{v_i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} c_{ij} \|(\mathbf{p}_i' - \mathbf{p}_j') - \mathbf{T}_i(\mathbf{w})(\mathbf{p}_i - \mathbf{p}_j)\|^2, \quad (5)$$

Deformation base

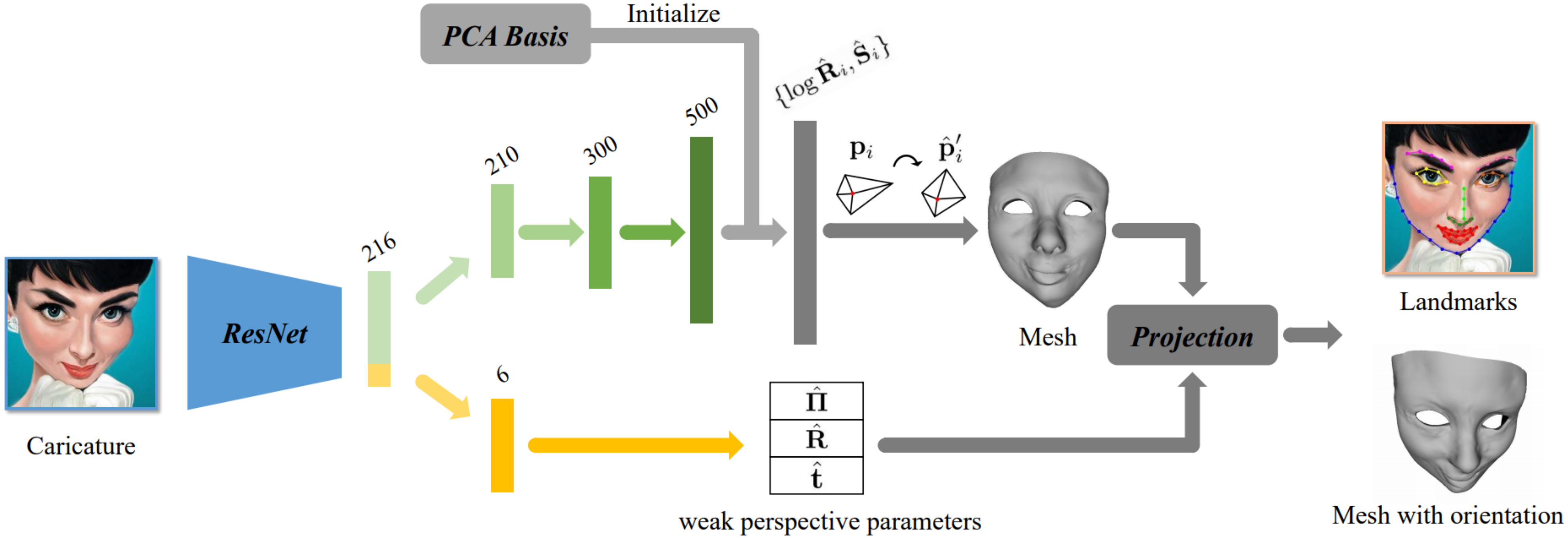
- Based on a reference mesh and n deformed meshes, build a linear combination of deformation representations:

$$\mathbf{T}_i(\mathbf{w}) = \exp(\sum_{l=1}^{n} w_{R,l} \log \mathbf{R}_i^l)(\mathbf{I} + \sum_{l=1}^{n} w_{S,l}(\mathbf{S}_i^l - \mathbf{I})), \quad (4)$$

- Compute the Jacobian matrix $\partial \mathbf{T}_i(\mathbf{w})/\partial \mathbf{w}$, then use the Levenberg-Marquardt algorithm to solve:

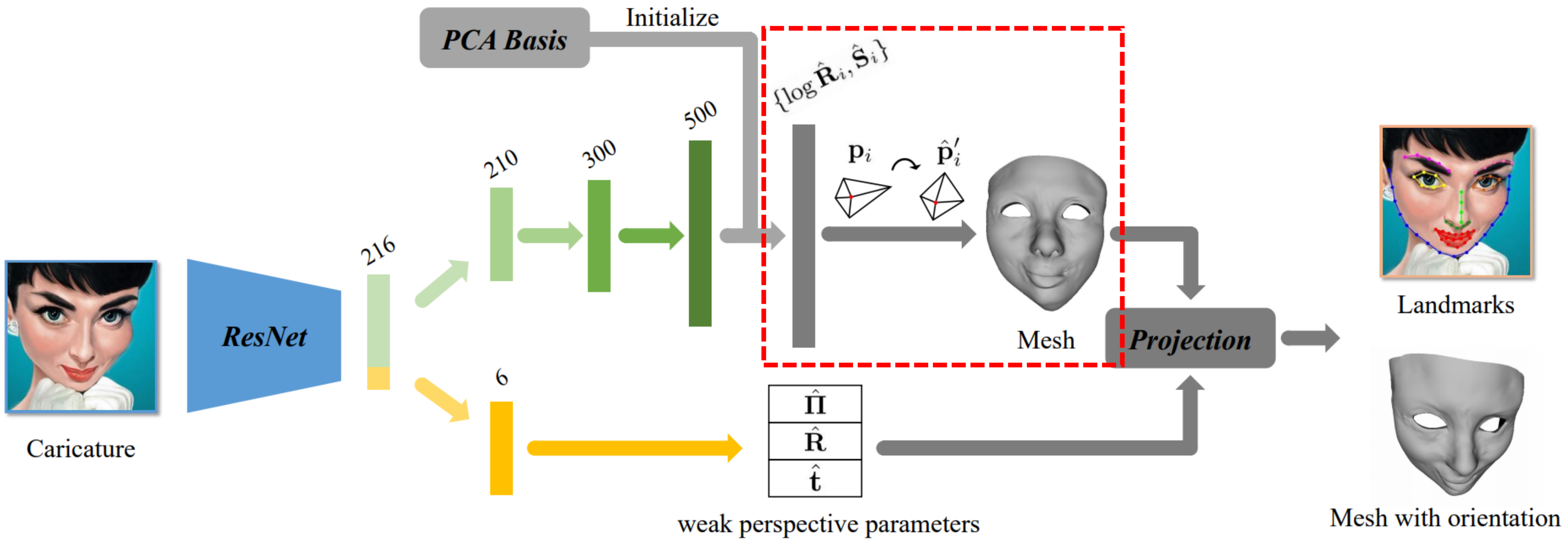
$$\min_{\mathbf{w}} \sum_{v_i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} c_{ij} \| \underbrace{(\mathbf{p}'_i - \mathbf{p}'_j)}_{\text{vertex of caricature}} - \mathbf{T}_i(\mathbf{w}) \underbrace{(\mathbf{p}_i - \mathbf{p}_j)}_{\text{vertex of mean face}} \|^2, \quad (5)$$

Deep learning framework



- use ResNet-34 backbone as the encoder, 3 Fully Connected layers as the decoder
- use the PCA basis of deformation presentation $\{\log \mathbf{R}_i, \mathbf{S}_i\}$ to initialize the last FC layer

Deep learning framework

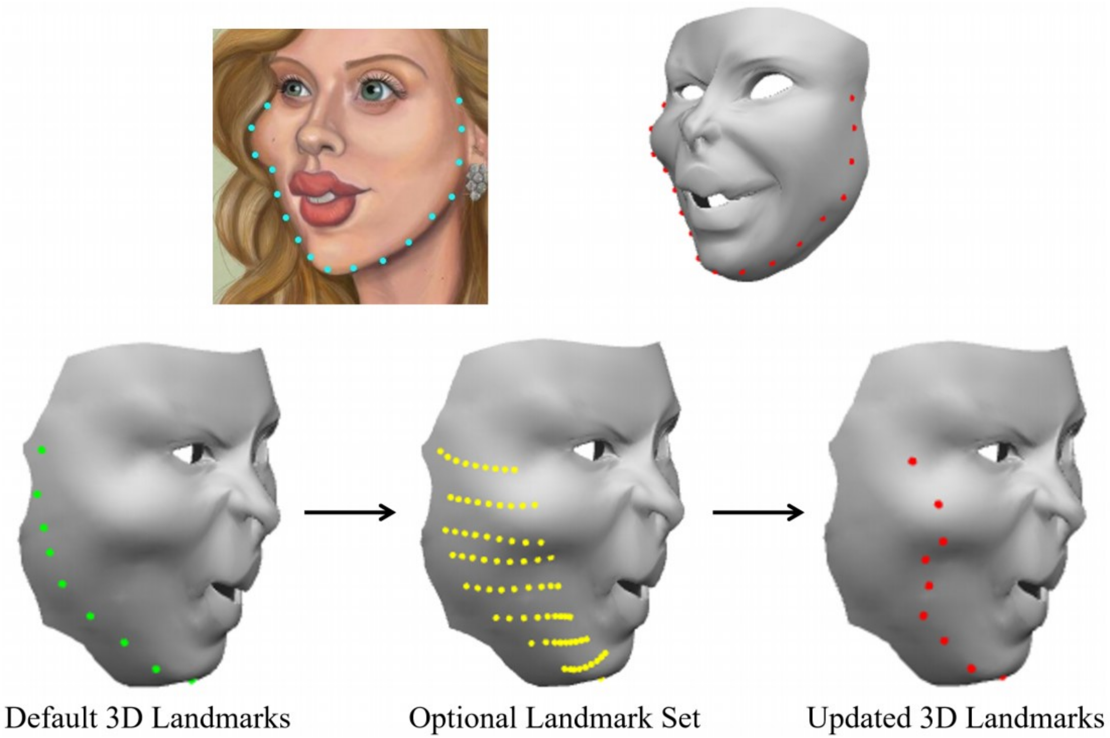


- Recover the vertex coordinate $\{\hat{\mathbf{p}}_i'\}$ from estimated deformation representation $(\log \hat{\mathbf{R}}_i, \hat{\mathbf{S}}_i)$ by solving:

$$\arg\min_{\{\hat{\mathbf{p}}_i'\}} \sum_{j \in \mathcal{N}_i} c_{ij} \|(\hat{\mathbf{p}}_i' - \hat{\mathbf{p}}_j') - \hat{\mathbf{T}}_i(\mathbf{p}_i - \mathbf{p}_j)\|_2^2, \qquad (6)$$

$$\Longleftrightarrow \quad 2 \sum_{j \in \mathcal{N}_i} c_{ij}(\hat{\mathbf{p}}_i' - \hat{\mathbf{p}}_j') = \sum_{j \in \mathcal{N}_i} c_{ij}(\hat{\mathbf{T}}_i + \hat{\mathbf{T}}_j)(\mathbf{p}_i - \mathbf{p}_j). \qquad (7)$$

Silhouette updating strategy



- Construct an optional landmark set from each horizontal line that has a vertex lying on the silhouette
- In each training time, select among them a set of updated silhouette landmarks according to estimated rotation matrix $\hat{\mathbf{R}}$

Loss Function

- Loss for Caricature Shape

$$\mathbf{E}_{ver}(\boldsymbol{\chi}_s) = \sum_{v_i \in \mathcal{V}} \|\hat{\mathbf{p}}'_i - \mathbf{p}'_i\|_2^2, \tag{8}$$

- Loss for Landmarks

$$\mathbf{E}_{lan}(\boldsymbol{\chi}_s, \boldsymbol{\chi}_p) = \sum_{v_i \in \mathcal{L}'} \|\hat{\mathbf{\Pi}}\hat{\mathbf{R}}\hat{\mathbf{p}}'_i + \hat{\mathbf{t}} - \mathbf{q}'_i\|_2^2, \tag{9}$$

- Total loss function

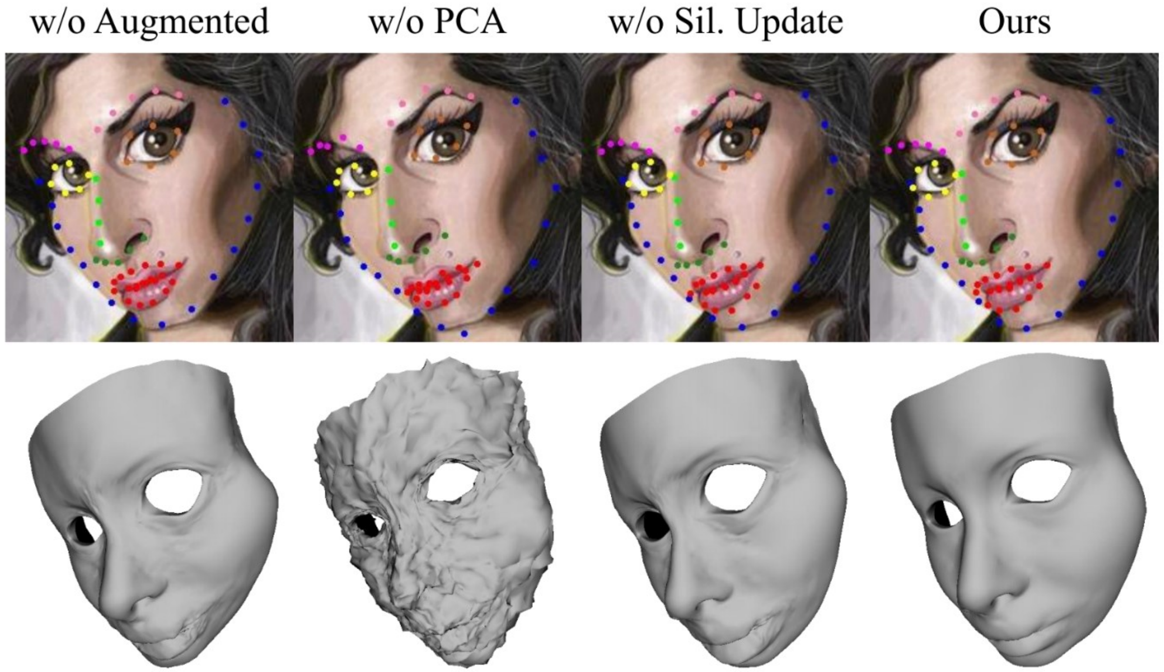
$$\mathbf{E} = \lambda_1 \mathbf{E}_{ver} + \lambda_2 \mathbf{E}_{lan}, \tag{10}$$

λ_1, λ_2 : hyperparameters

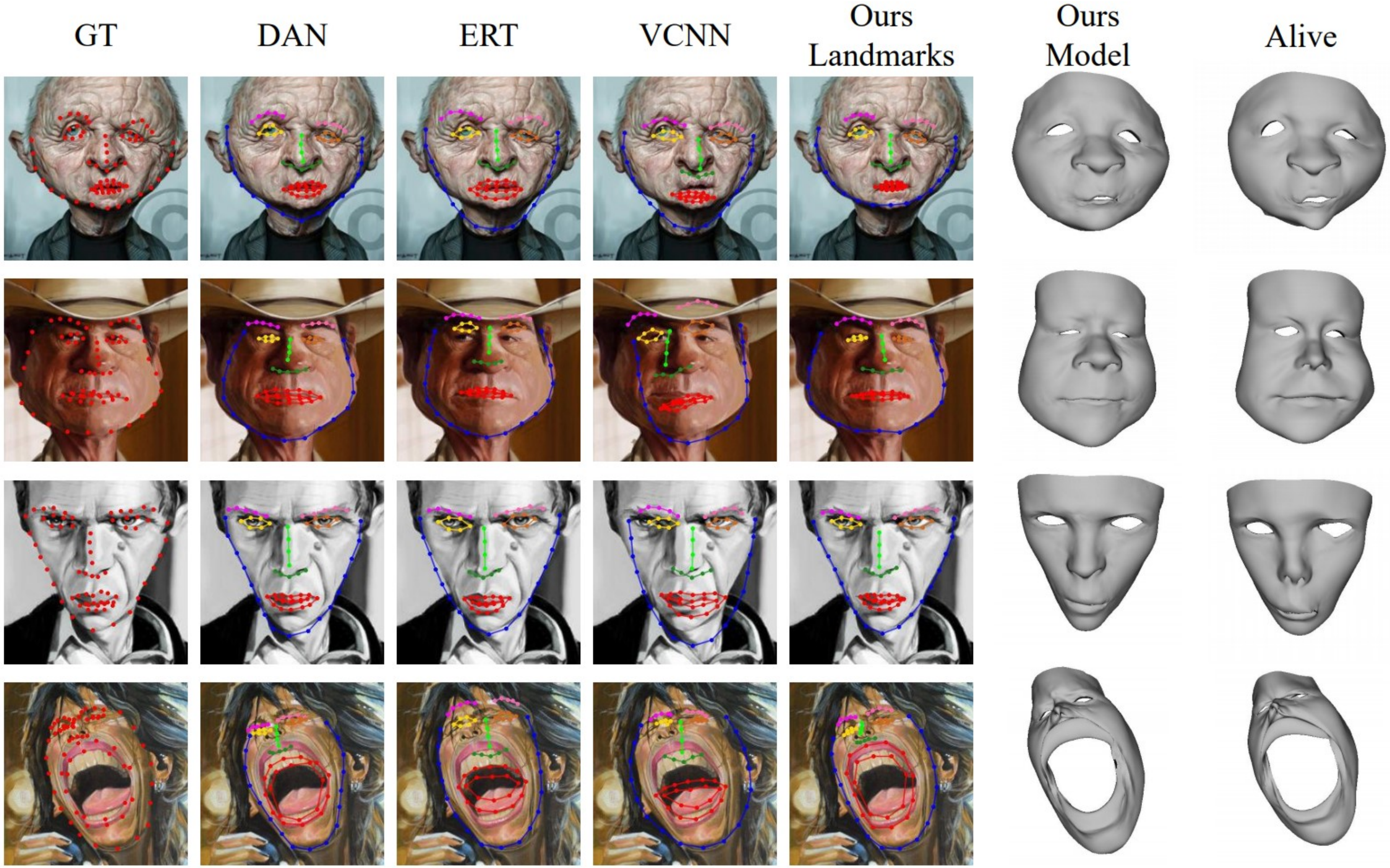
Ablation studies

- Augmented data
- PCA initialization
- Silhouette updating strategy

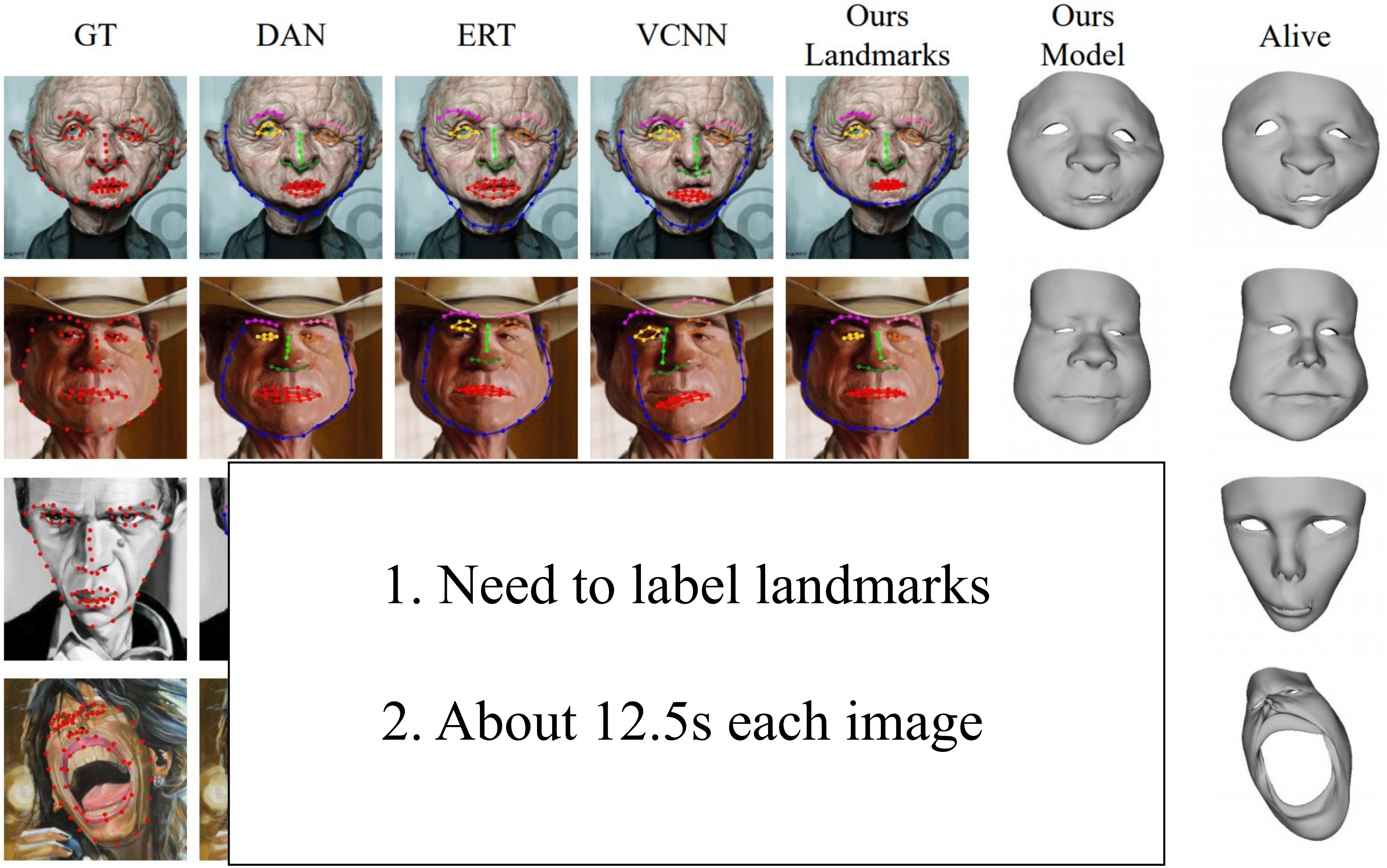
	mean error	inter -pupil	inter -ocular	diagonal
w/o Augmented	5.85	9.29	6.34	2.38
w/o PCA	6.91	11.01	7.52	2.82
w/o Sil. Update	5.99	9.49	6.48	2.44
Ours	**5.64**	**8.93**	**6.10**	**2.30**



Comparison



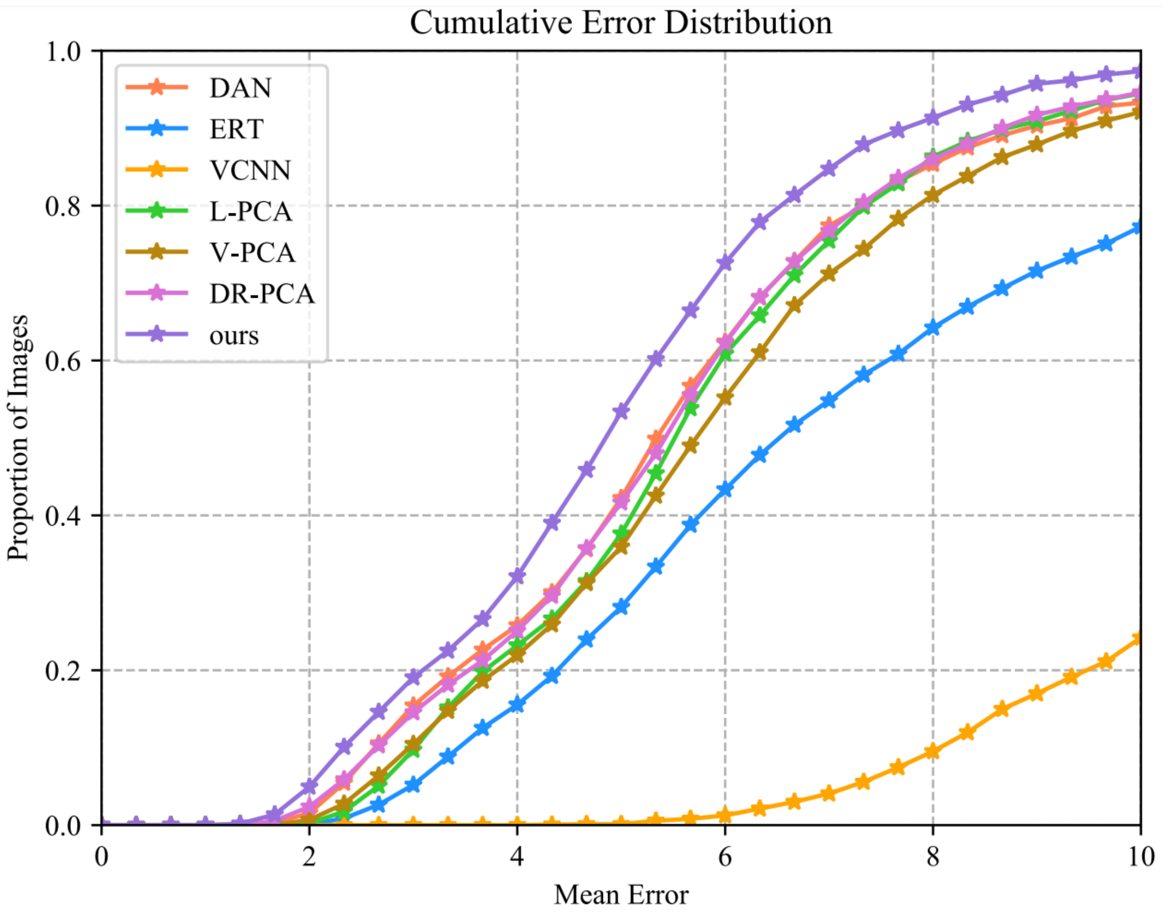
Comparison



1. Need to label landmarks

2. About 12.5s each image

Comparison



	mean error	inter -pupil	inter -ocular	diagonal	time (ms)
DAN	5.78	9.93	6.80	2.59	25.9
ERT	8.24	14.52	9.95	3.71	2.7
VCNN	14.04	24.33	16.67	6.39	**1.6**
L-PCA	5.87	10.08	6.91	2.64	4.8
V-PCA	6.20	10.68	7.32	2.79	6.4
DR-PCA	5.75	9.89	6.77	2.58	9.3
Ours	**4.98**	**8.51**	**5.82**	**2.23**	9.8

Thanks

Q&A

Our constructed dataset, source code, and trained model are available at:
https://github.com/Juyong/CaricatureFace