CVPR 2018
Salt Lake City

Alive Caricature from 2D to 3D

Qianyi Wu Juyong Zhang Yu-Kun Lai Jianmin Zheng Jianfei Cai

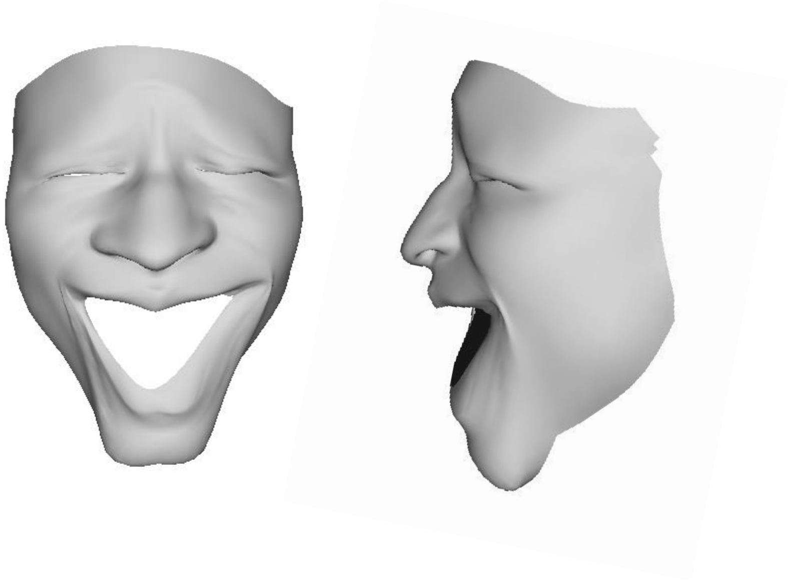
University of Science and Technology of China
CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD
NANYANG TECHNOLOGICAL UNIVERSITY

Goal

- Generate 3D model from 2D caricature image.

Input: 2D caricature image

Output: 3D caricature face model



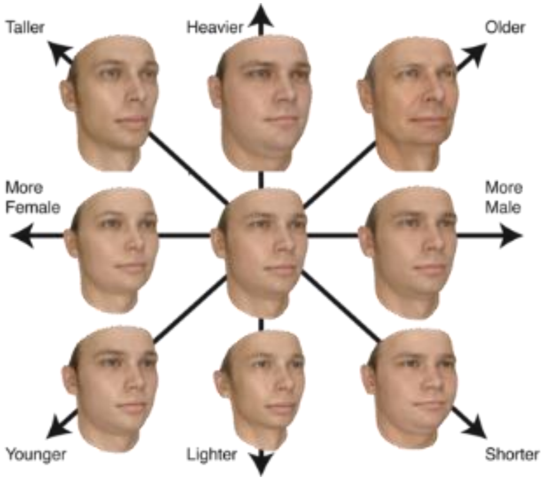
Current solution

- 3D face statistical representation
- Shape from shading/Inverse rendering
- Machine learning algorithm
- Interactive modeling

Current solution

- 3D face statistical representation
- Shape from shading/inverse rendering
- Machine learning algorithm
- Interactive modeling

Current solution



- 3D face statistical representation
  - 3D Morphable Model (3DMM):

$$P = \bar{P} + \sum_i \alpha_i P_i$$

P : a 3D face, $\bar{P}$: mean face, P_i: principal basis, α_i: parameter of each basis

  - Bilinear model:

$$P = C_r \times_2 u_{id} \times_3 u_{exp}$$

C_r: Core tensor; u_{id}, u_{exp}: coefficient of identity and expression

[1] Blanz V et al. *A morphable model for the synthesis of 3D faces*. ACM TOG
[2] C. Cao et al. *Facewarehouse: A 3D facial expression database for visual computing*. IEEE TVCG

Current solution

- 3D face statistical representation
  - 3D morphable model (3DMM):

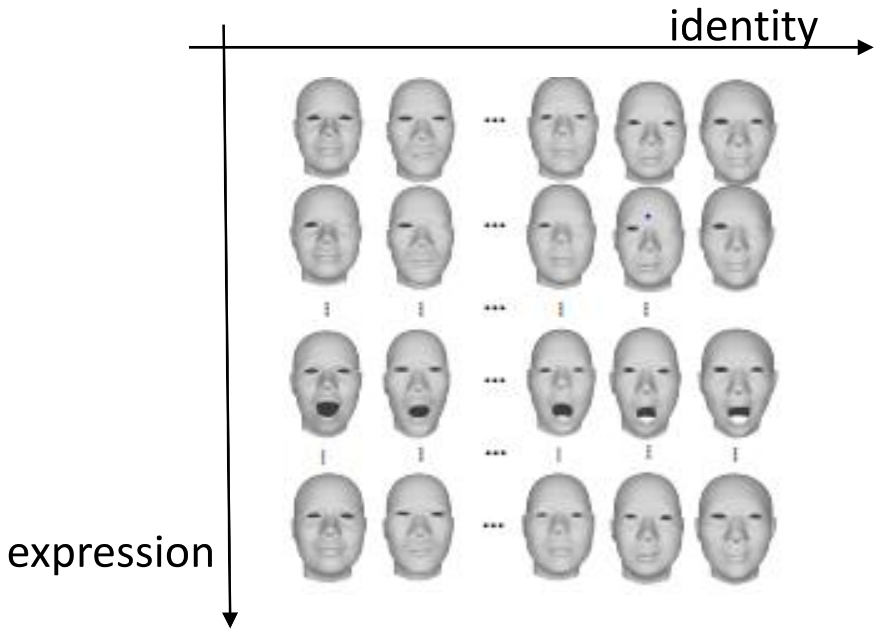
$$P = \bar{P} + \sum_i \alpha_i P_i$$

P : a 3D face, $\bar{P}$: mean face, P_i: principal basis, α_i: parameter of each basis

  - Bilinear model:

$$P = C_r \times_2 u_{id} \times_3 u_{exp}$$

C_r: Core tensor; u_{id}, u_{exp}: coefficient of identity and expression



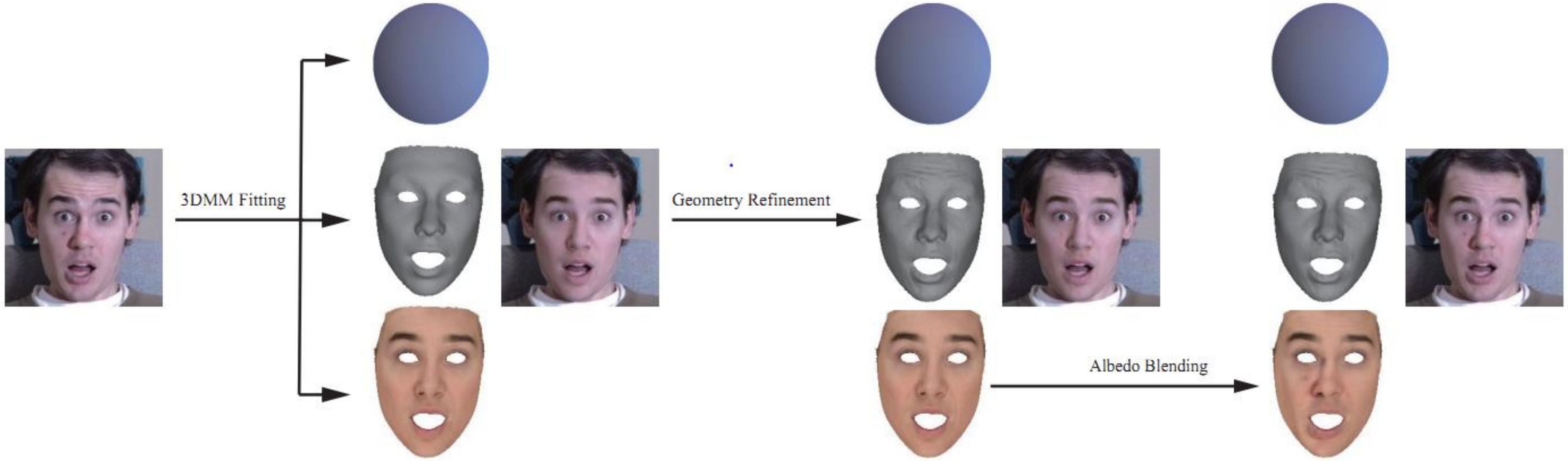
[1] Blanz V et al. *A morphable model for the synthesis of 3D faces*. ACM TOG
[2] C. Cao et al. *Facewarehouse: A 3D facial expression database for visual computing*. IEEE TVCG

Current solution

- 3D face statistical representation
- Shape from shading/Inverse rendering
- Machine learning algorithm
- Interactive modeling

Current solution

- Shape from shading/Inverse rendering
  - Add lighting/reflectance information generating 3D face model



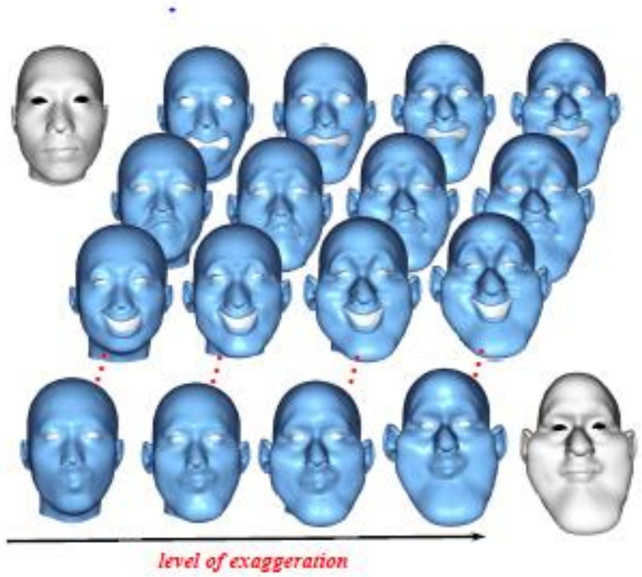
[1] I. Kemelmacher-Shlizerman et al. *3D face reconstruction from a single image using a single reference face shape*. T-PAMI
[2] Y. Guo et al. *CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images.* T-PAMI

Current solution

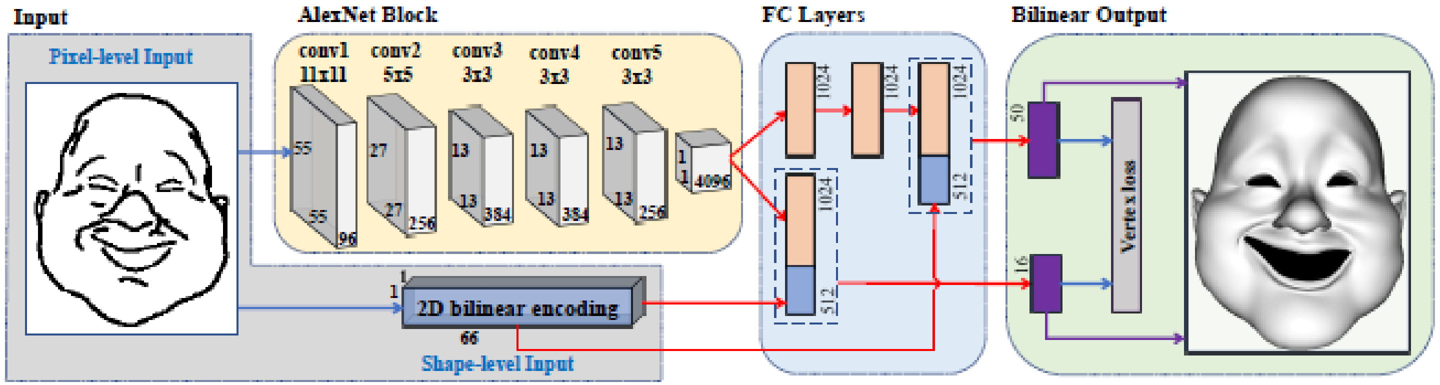
- 3D face statistical representation
- Shape from shading/Inverse rendering
- Machine learning algorithm
- Interactive modeling

Current solution

- Machine learning algorithm
  - Build 3D caricature dataset



  - Then use learning method to generate 3D model



[1] X. Han et al. *DeepSketch2Face: a deep learning based sketching system for 3D face and caricature modelling*. ACM TOG

Current solution

- Interactive modeling
  - Artist use specific software to create 3D model.

  - This is the best method, but need domain knowledge and professional skill. And it is usually a time-consuming process.

Challenges

- A large variety of caricature styles
- Caricature image may not reflect real shading information
- Create a 3D caricature dataset is time-consuming

Observation

- Caricature have two basic characteristics
  - Face constraint. i.e. we can tell they are faces
  - The features of the face have been exaggerated.

Observation

Caricature modeling problem can be treat as deformation problem

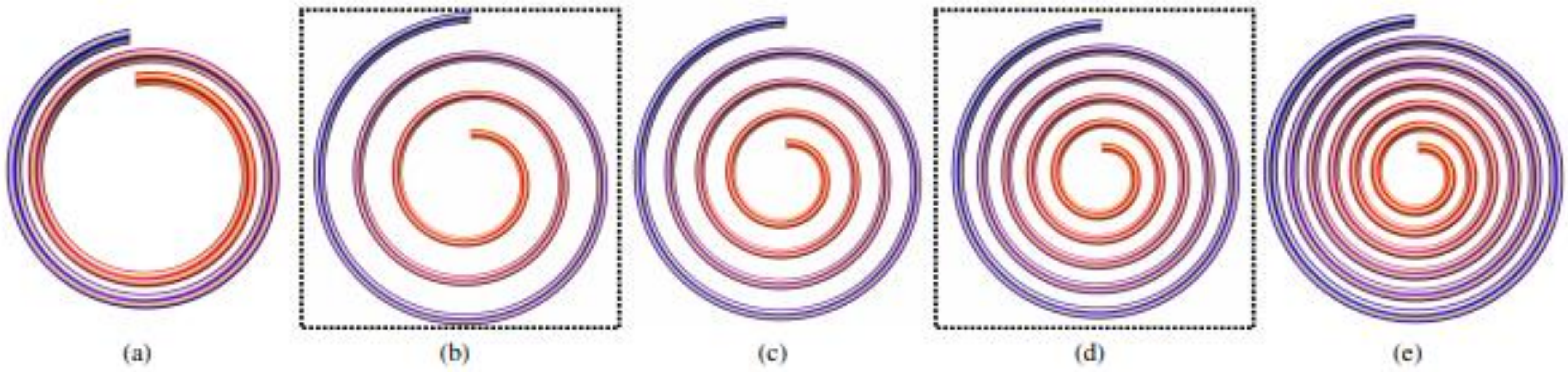


Fig. 3. Shape blending with interpolation and extrapolation. (b) and (d) are the source ($t = 0$) and target ($t = 1$) models. (a),(c),(e) are interpolated/extrapolated models with $t = -0.5, 0.5, 1.5$, respectively.

[1] L. Gao, Y.-K. Lai, D. Liang, S.-Y. Chen, and S. Xia. *Efficient and flexible deformation representation for data-driven surface modelling*. ACM TOG

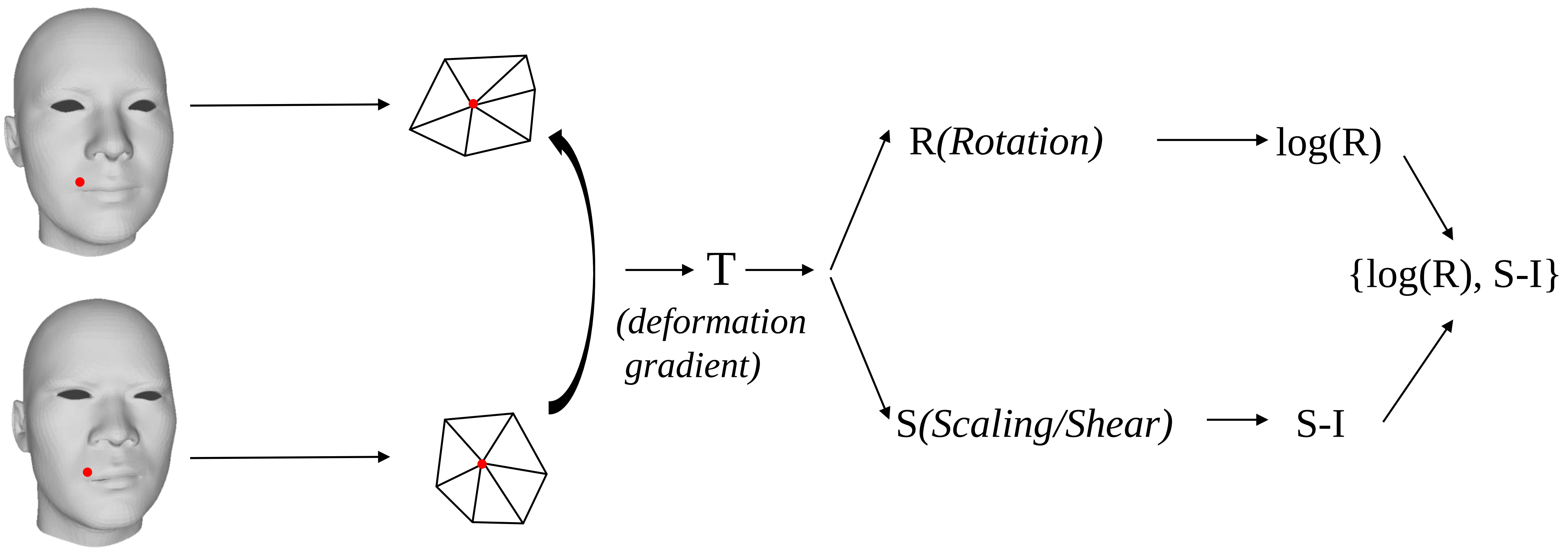
Our solution

- Build deformation base on normal 3D face dataset
- Formulate 3D caricature generation problem as an optimization problem

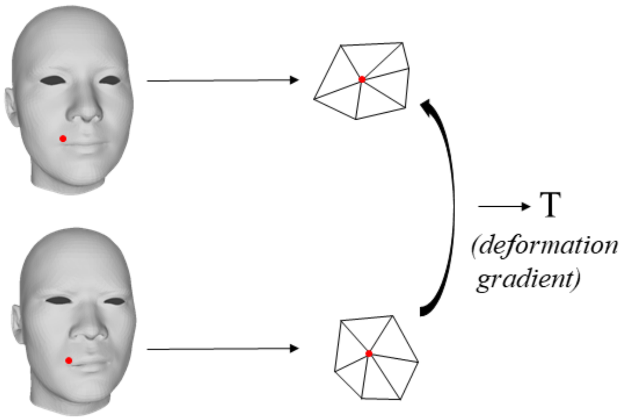
Our solution

- Build deformation base on normal 3D face dataset
- Formulate 3D caricature generation problem as an optimization problem

Deformation representation



Deformation representation



- Compute *deformation gradient* T of i^{th} vertex with edge weight c_{ij}:

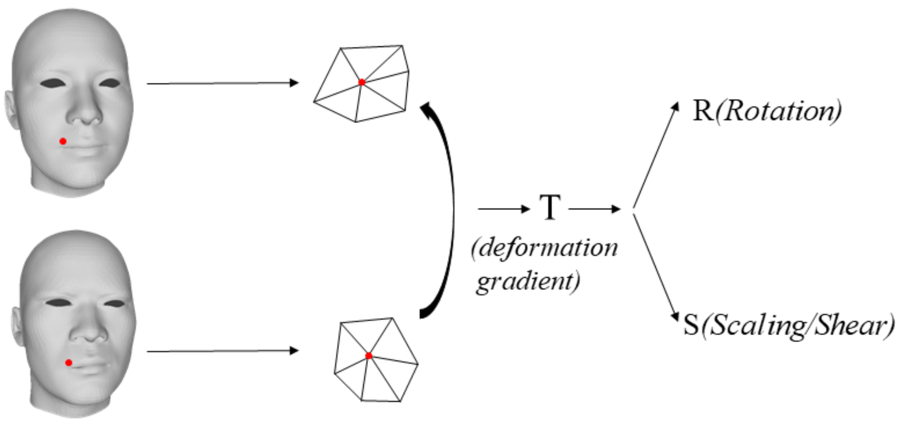
$$E(T_i) = \sum_{j \in N_i} c_{ij} ||(p'_i - p'_j) - T_i(p_i - p_j)||^2)$$

- Polar decomposition of T_i. Decompose T_i into rotation and scaling/shear parts.

$$T_i = R_i S_i$$

- Logarithm of rotation part R_i. It allow effective linear combination for $\log(R)$
- Transformation of scaling/shear part S_i. Using $S_i - I$ instead of S_i

Deformation representation



- Compute *deformation gradient* T of i^{th} vertex with edge weight c_{ij}:

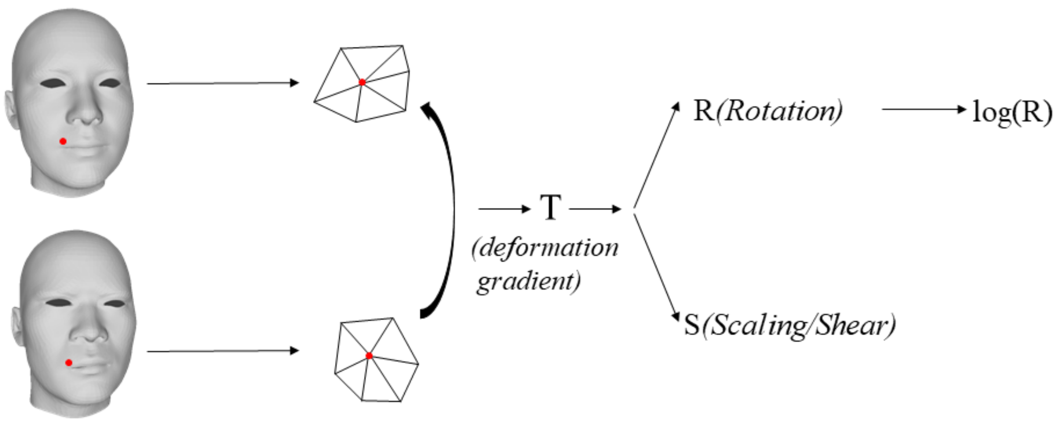
$$E(T_i) = \sum_{j \in N_i} c_{ij}||(p'_i - p'_j) - T_i(p_i - p_j)||^2)$$

- Polar decomposition of T_i. Decompose T_i into rotation and scaling/shear parts.

$$T_i = R_i S_i$$

- Logarithm of rotation part R_i. It allow effective linear combination for $\log(R)$
- Transformation of scaling/shear part S_i. Using $S_i - I$ instead of S_i

Deformation representation



- Compute *deformation gradient* T of i^{th} vertex with edge weight c_{ij}:

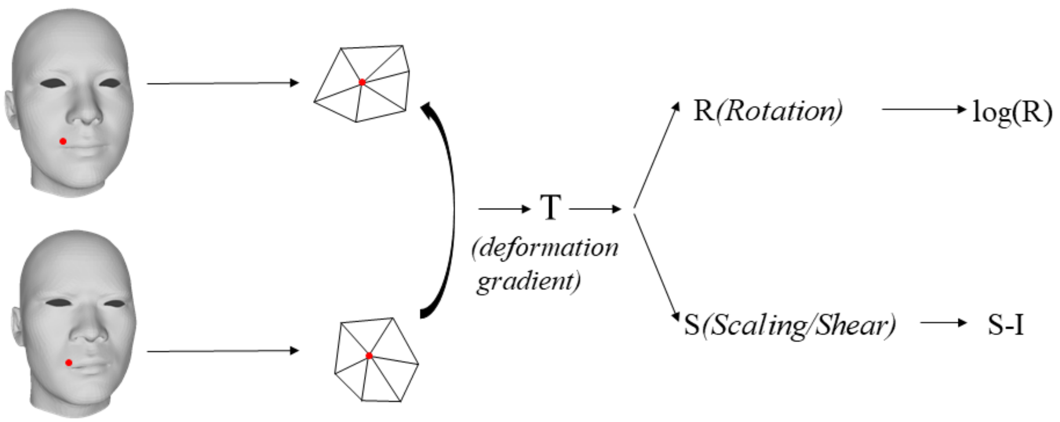
$$E(T_i) = \sum_{j \in N_i} c_{ij} ||(p_i' - p_j') - T_i(p_i - p_j)||^2)$$

- Polar decomposition of T_i. Decompose T_i into rotation and scaling/shear parts.

$$T_i = R_i S_i$$

- Logarithm of rotation part R_i. It allow effective linear combination for $\log(R)$
- Transformation of scaling/shear part S_i. Using $S_i - I$ instead of S_i

Deformation representation



- Compute *deformation gradient* T of i^{th} vertex with edge weight c_{ij}:

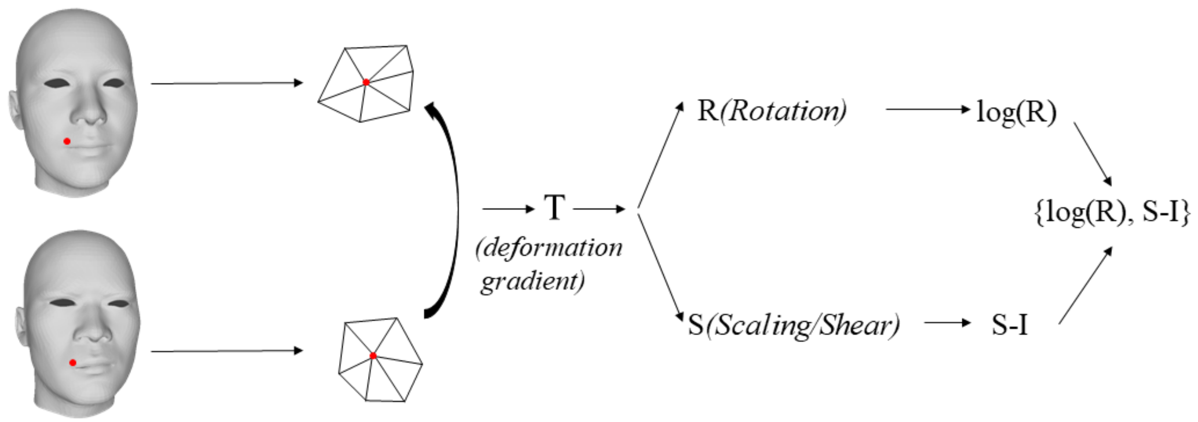
$$E(T_i) = \sum_{j \in N_i} c_{ij} ||(p_i' - p_j') - T_i(p_i - p_j)||^2)$$

- Polar decomposition of T_i. Decompose T_i into rotation and scaling/shear parts.

$$T_i = R_i S_i$$

- Logarithm of rotation part R_i. It allow effective linear combination for $\log(R)$
- Transformation of scaling/shear part S_i. Using $S_i - I$ instead of S_i

Deformation representation



- Compute *deformation gradient* T of i^{th} vertex with edge weight c_{ij}:

$$E(T_i) = \sum_{j \in N_i} c_{ij} ||(p_i' - p_j') - T_i(p_i - p_j)||^2)$$

- Polar decomposition of T_i. Decompose T_i into rotation and scaling/shear parts.

$$T_i = R_i S_i$$

- Logarithm of rotation part R_i. It allow effective linear combination for $\log(R)$
- Transformation of scaling/shear part S_i. Using $S_i - I$ instead of S_i

Apply it to each vertex, and we can represent the deformation from one mesh to another mesh as f:

$$f = \{\log(R_i), S_i - I \mid i = 1, 2, .., \#(vertices)\}$$

Note: T_i can be represent as $\exp(\mathbf{log}(\boldsymbol{R}))\,(I + (\boldsymbol{S_i} - \boldsymbol{I}))$

Deformation base

- Suppose we have N face model, we can obtain N deformation representation $F = \{f_l \mid l = 1, 2, \ldots, N\}$

- To generate a new deformed mesh based on F. The deformation gradient of a new deformed mesh as:

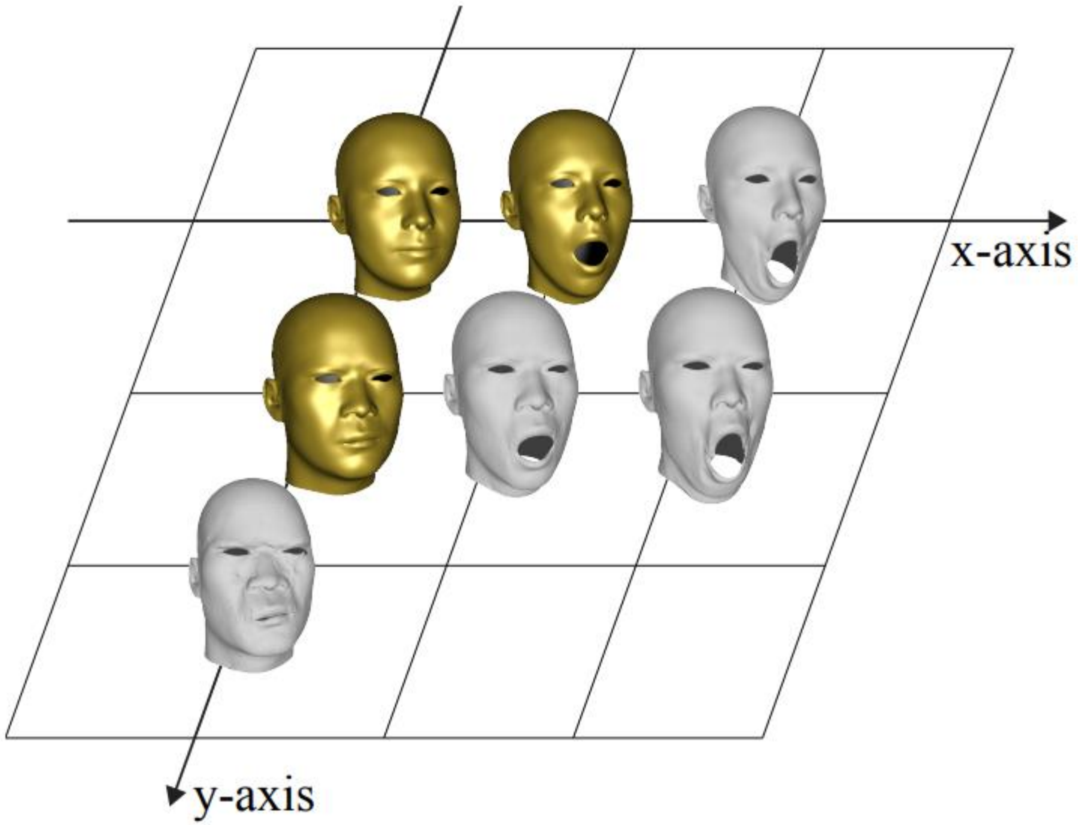
$$T_i(w) = \exp\Big(\sum_{l=1}^{n} w_{R,l}\, \boldsymbol{log}(\boldsymbol{R_{l,i}})\Big)\Big(I + \sum_{l=1}^{n} w_{S,l}(\boldsymbol{S_{l,i}} - \boldsymbol{I})\Big)$$

$\{w_{R,l}, w_{S,l}\}$ are the combination weights.

Simple Example

- Special situation: $w_{R,l} = w_{S,l}$
- Reference model at (0,0), two deformed model at (1,0) and (0,1).
- Using combination of deformation basis,

we can obtain some 3D new face.



Optimization Framework

- Deformation energy E_{def} defined as:

$$E_{def}(P', w) = \sum_{v_i \in V} \left(\sum_{j \in N_i} c_{ij} ||\left(p_i' - p_j'\right) - T_i(w)\left(p_i - p_j\right)||^2 \right)$$

- By minimizing this energy, we are able to determine P' given weights $w = \{w_R, w_{S'}\}$ or obtain the combination weights w given the deformed mesh P'.

➢ P'-$step$: Given combination weights w, find best P'. It equals to solve a linear least squares for P'.

➢ w-$step$: Given deformed 3D model P', find best weight w. This is a non-linear least squares problem because of $T_i(w)$. With the Jacobian matrix w.r.t. to the rotation weight w_R and scaling/shear weight $w_{S'}$, we can use non-linear least squares algorithm to solve it.

Our solution

- Build deformation base on normal 3D face dataset

- Formulate 3D caricature generation problem as an optimization problem

3D Caricature Generation

- Use weakly perspective projection to set up the relationship between 3D model and 2D image.

$$\mathbf{q}_i = \underbrace{s\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}}_{\Pi}\underbrace{\mathbf{R}}_{r}\mathbf{p}_i + \mathbf{t}$$

$$\Pi r {p_i}' + t$$

3D Caricature Generation

- Use weakly perspective projection to set up the relationship between 3D model and 2D image.

$$\mathbf{q}_i = \underbrace{s \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}}_{\Pi} \underbrace{\mathbf{R}}_{r} \mathbf{p}_i + \mathbf{t}$$

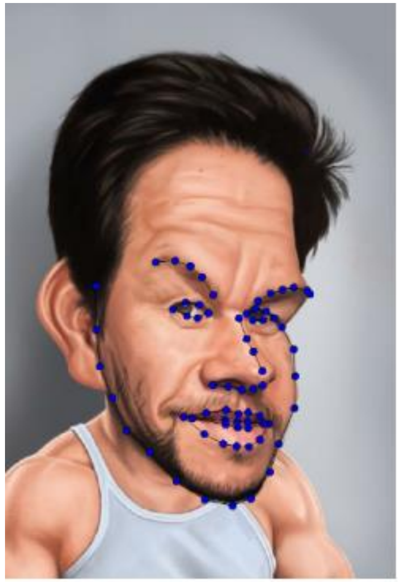
$$\Pi r p_i + t$$

3D Caricature Generation

- Defined landmark fitting loss:

$$E_{lan}(\Pi, r, t, P') = \sum_{v_i \in L} ||\Pi r p_i{}' + t - q_i||^2$$

to measure the distance of projected 3D landmarks and 2D landmarks



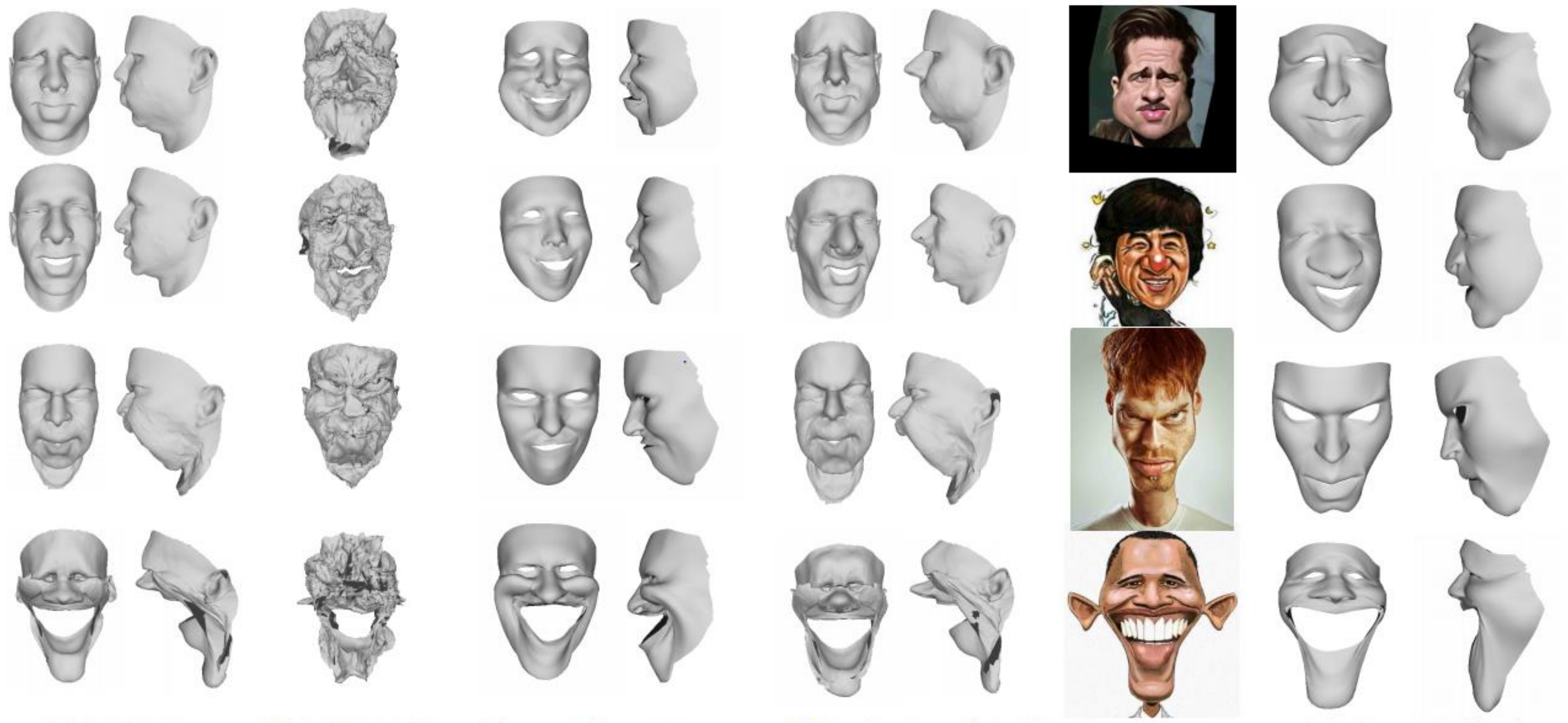
3D Caricature Generation

- The generation problem is formulated as an optimization problem:

$$\min_{P', w, \Pi, r, t} E_{def}(P', w) + \lambda E_{lan}(\Pi, r, t, P')$$

To solve it, we also iterate P'-$step$ and w-$step$ to obtain 3D model.

Comparision



(a)3DMM (b)3DMM(-) (c)FaceWareHouse (d)Caricatured 3DMM (e) Our Method

Pipeline

landmarks
detection

Pipeline

landmarks detection

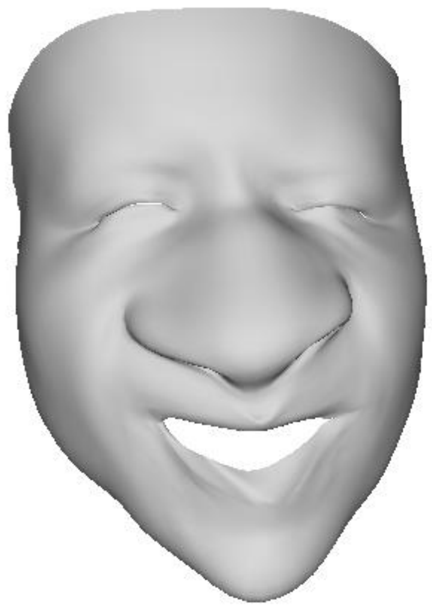
Pipeline

landmarks detection

update landmarks

Pipeline

Pipeline



Some results of our methods

Discussion

- Comprehension
- Q&A