Face Alignment Across Large Poses: A 3D Solution

Outline

- Face Alignment
- Related Works
- 3D Morphable Model
- Projected Normalized Coordinate Code
- Network Structure
- 3D Image Rotation
- Performance on Datasets
- Extensions

What is Face Alignment? Why do we need it?

- Face alignment can be formulated as a problem of searching over a face image for the pre-defined facial points (also called face shape), which typically starts from a coarse initial shape, and proceeds by refining the shape estimate step by step until convergence.
- Can be applied to face detection, face recognition, expression detection and so on.

http://blog.csdn.net/zhangjunhit/article/details/78435972

More about face alignment

- Modeling: each facial point can be detected according to visual patterns. However, when faces deviate from the frontal view, some landmarks become invisible. In medium-pose scenario, this can be tackled by changing semantic positions.
- Fitting: traditional linear/nonlinear cascaded regression is not enough.
- Data labeling: manual data labeling is not practical

Related works

- Generic face alignment: locate a sparse set of fiducial landmarks in 2D space
  - Active Appearance Model:

    $x = \mu + P*b$

    From mean model μ and some (b) linear combination of principal components P
    Fit one image to another image region, by altering b according to ΔI
  - Constrained Local Model
- Large pose alignment
  - 3D Alignment of Face in a Single Image (L. Gu, 2006)
    - Construct a compact 3D shape prior on the sparse 3D point set, and apply it to constrain the 2D facial points in different views.
    - Only based on sparse patch and facial points

3D Morphable Model

- In original Morphable 3D model (3DMM):

$$\mathbf{S}_{mod} = \sum_{i=1}^{m} a_i \mathbf{S}_i \,, \quad \mathbf{T}_{mod} = \sum_{i=1}^{m} b_i \mathbf{T}_i \,, \quad \sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_i = 1$$

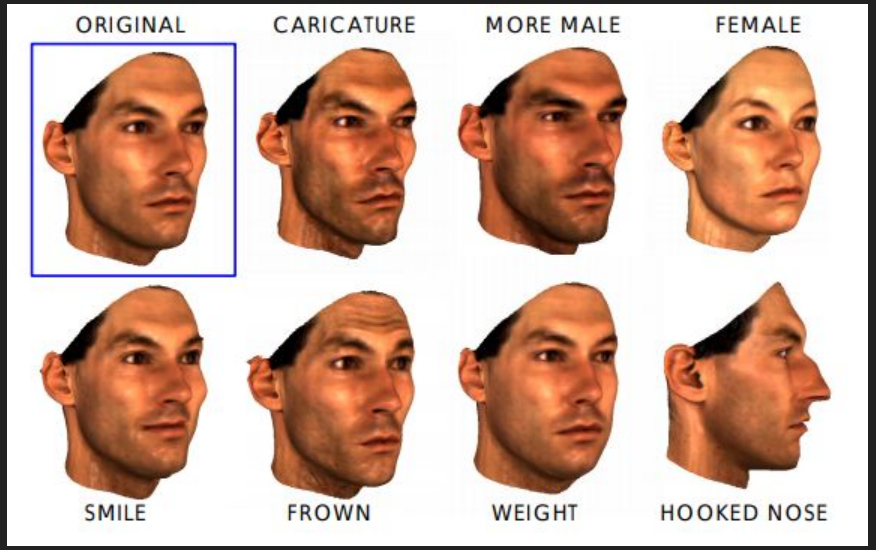
S is shape vector (X1, Y1, Z1, X2, …, Yn, Zn)

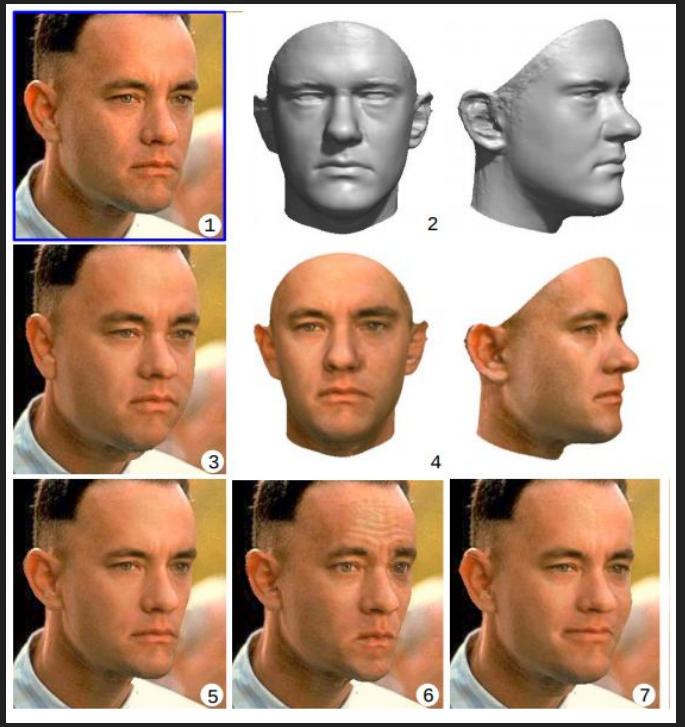
T is color values of n vertices (R1, G1, B1, R2, …, Gn, Bn)

$$S_{model} = \overline{S} + \sum_{i=1}^{m-1} \alpha_i s_i \,, \quad T_{model} = \overline{T} + \sum_{i=1}^{m-1} \beta_i t_i$$

A Morphable Model For The Synthesis Of 3D Faces, V. Blanz, 1999

3D Morphable Model





A Morphable Model For The Synthesis Of 3D Faces, V. Blanz, 1999

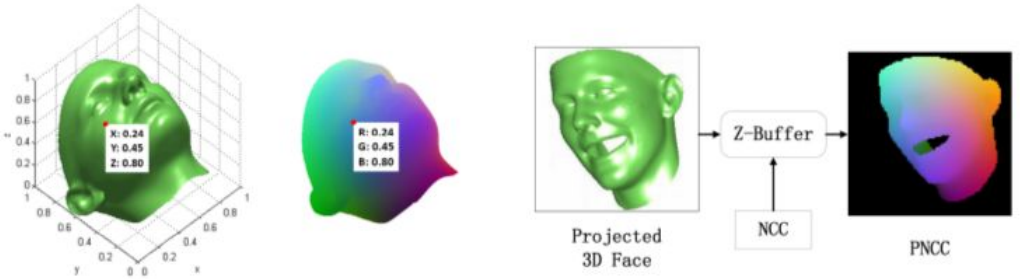
3D Morphable Model

$$\mathbf{S} = \overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}$$

$$V(\mathbf{p}) = f * \mathbf{Pr} * \mathbf{R} * (\overline{\mathbf{S}} + \mathbf{A}_{id}\boldsymbol{\alpha}_{id} + \mathbf{A}_{exp}\boldsymbol{\alpha}_{exp}) + \mathbf{t}_{2d}$$

- V(p) is the model construction and projection function, leading to the 2D positions of model vertices, where f is the scale factor, Pr is the orthographic projection matrix.
- The collection of all the model parameters is p = [f, pitch, yaw, roll, t_2d, α_id, α_exp], which is 234-dimensional.

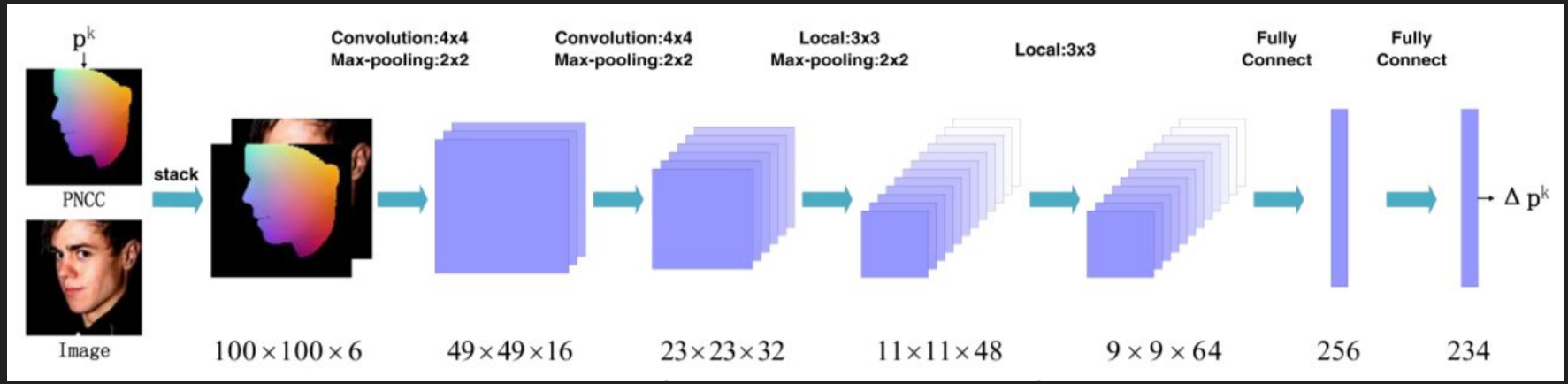
Projected Normalized Coordinate Code



(a) NCC (b) PNCC

Figure 3. The Normalized Coordinate Code (NCC) and the Projected Normalized Coordinate Code (PNCC). (a) The normalized mean face, which is also demonstrated with NCC as its texture (NCC_x = R, NCC_y = G, NCC_z = B). (b) The generation of PNCC: The projected 3D face is rendered by Z-Buffer with NCC as its colormap.

Cascaded CNN Network



Cost function:

$$E_{wpdc} = (\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0))^T \mathbf{W} (\Delta\mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0))$$

$$where \quad \mathbf{W} = diag(w_1, w_2, ..., w_n)$$

$$w_i = \|V(\mathbf{p}^d(i)) - V(\mathbf{p}^g)\| / \sum w_i \qquad (8)$$

$$\mathbf{p}^d(i)_i = (\mathbf{p}^0 + \Delta\mathbf{p})_i$$

$$\mathbf{p}^d(i)_j = \mathbf{p}^g_j, \quad j \in \{1, \ldots, i-1, i+1, \ldots, n\},$$

Cost function analysis

$$
\begin{aligned}
&E_{wpdc} = (\Delta \mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0))^T \mathbf{W} (\Delta \mathbf{p} - (\mathbf{p}^g - \mathbf{p}^0)) \\
&where \quad \mathbf{W} = diag(w_1, w_2, ..., w_n) \\
&w_i = \|V(\mathbf{p}^d(i)) - V(\mathbf{p}^g)\| / \sum w_i \qquad (8) \\
&\mathbf{p}^d(i)_i = (\mathbf{p}^0 + \Delta \mathbf{p})_i \\
&\mathbf{p}^d(i)_j = \mathbf{p}^g_j, \quad j \in \{1, \ldots, i-1, i+1, \ldots, n\},
\end{aligned}
$$

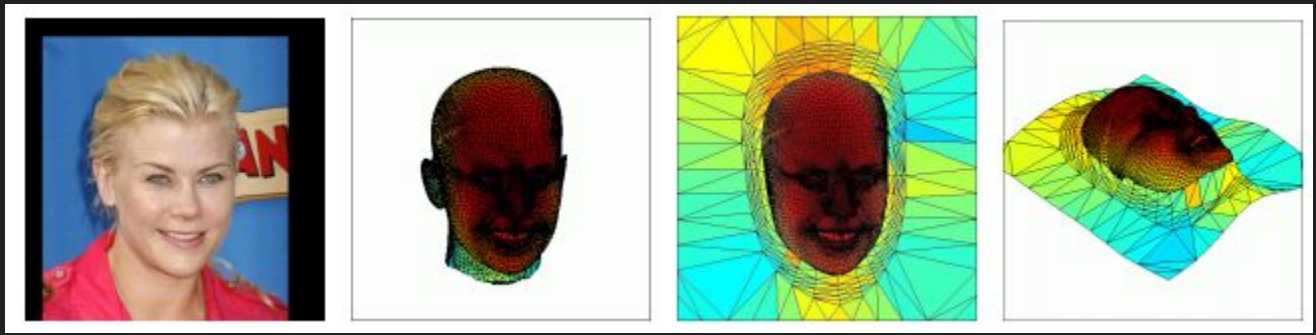
- CNN first concentrates on the parameters with larger wi, such as scale, rotation and translation. As the trained parameters **p** closer to ground truth, CNN will optimize less important parameters.

What is face profiling and why do we need it?

- All the discriminative methods rely on training data, especially for CNN
- Few released face alignment database contains large-pose samples
- Face profiling: generate profile view of faces from medium-pose images.
- By doing face profiling, the authors are able to generate as enough training data as possible

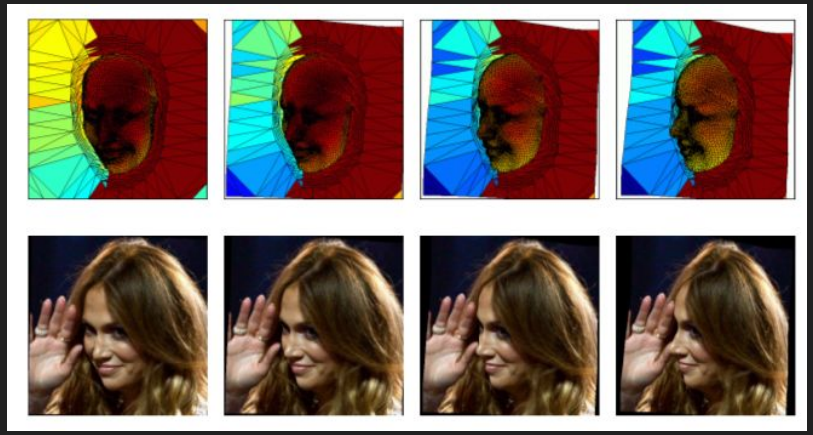
3D Image Meshing

- Fit a 3DMM through the Multi-Features Framework (MFF)
- The MFF algorithm can always fit with the groundtruth landmarks
- Few failed samples can be easily adjusted manually
- On external regions, follow the 3D meshing algorithm to estimate depth information of anchor points



Generating images by 3D rotation

- When the depth information is estimated, the face image can be rotated in 3D space to generate the appearances in larger poses.
- The authors enlarge the yaw of the depth image at the step of 5 degree until 90 degree. Through face profiling, we not only obtain the face appearances in large poses and but also augment the dataset to a large scale.



Initialization Regeneration

- How to avoid overfitting?
- Model the perturbation of a training sample with a set of similar face posture samples
- Create validation set from members which share similar face postures.
- Update:

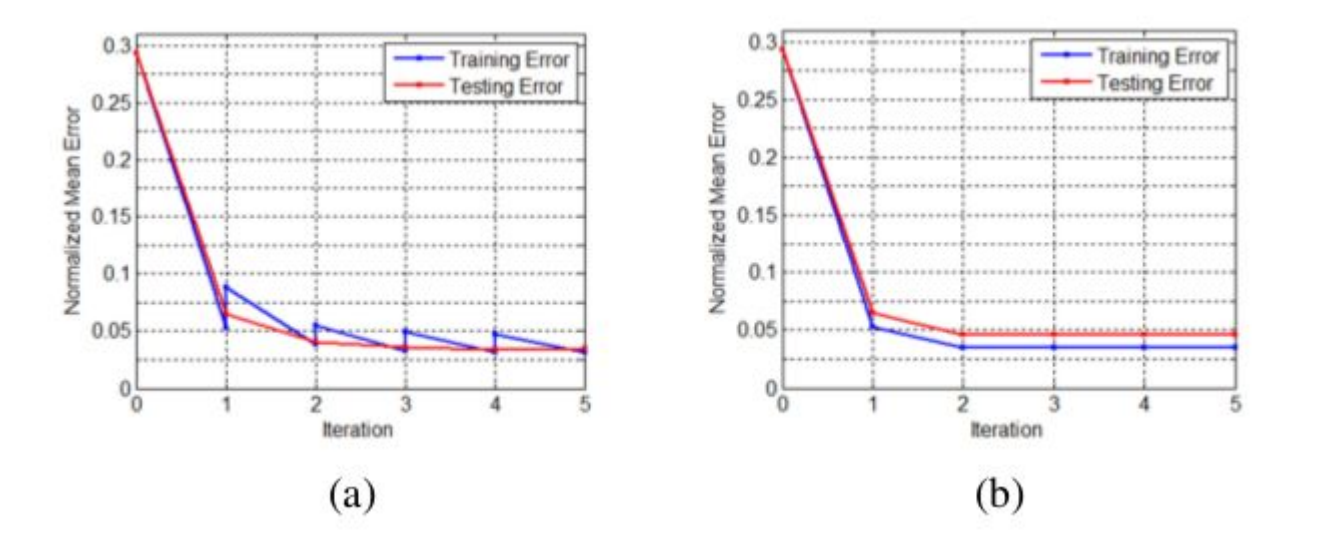
$$\mathbf{p}^k = \mathbf{p}^g - (\mathbf{p}^g_{v_i} - \mathbf{p}^k_{v_i})$$

- By doing so, “guide” derivative in a meaningful way.

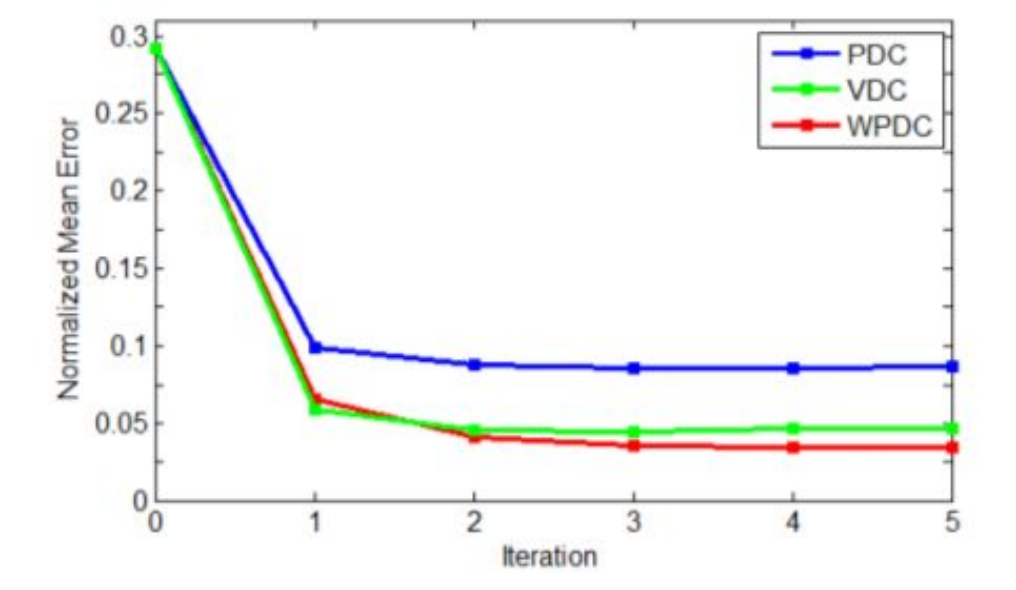
Landmark refinement

- HOG is adopted to refine the location of landmarks after 3DDFA.

Experiments: w/o regeneration



Experiments: different cost functions



Experiments: comparison

	AFLW Dataset (21 pts)					AFLW2000-3D Dataset (68 pts)				
Method	[0, 30]	[30, 60]	[60, 90]	Mean	Std	[0, 30]	[30, 60]	[60, 90]	Mean	Std
CDM [49]	8.15	13.02	16.17	12.44	4.04	-	-	-	-	-
RCPR [7]	6.16	18.67	34.82	19.88	14.36	-	-	-	-	-
RCPR(300W)	5.40	9.80	20.61	11.94	7.83	4.16	9.88	22.58	12.21	9.43
RCPR(300W-LP)	5.43	6.58	11.53	7.85	3.24	4.26	5.96	13.18	7.80	4.74
ESR(300W)	5.58	10.62	20.02	12.07	7.33	4.38	10.47	20.31	11.72	8.04
ESR(300W-LP)	5.66	7.12	11.94	8.24	3.29	4.60	6.70	12.67	7.99	4.19
SDM(300W)	**4.67**	6.78	16.13	9.19	6.10	**3.56**	7.08	17.48	9.37	7.23
SDM(300W-LP)	**4.75**	5.55	9.34	6.55	2.45	3.67	4.94	9.76	6.12	3.21
3DDFA	5.00	**5.06**	**6.74**	**5.60**	**0.99**	3.78	**4.54**	**7.93**	**5.42**	**2.21**
3DDFA+SDM	**4.75**	**4.83**	**6.38**	**5.32**	**0.92**	**3.43**	**4.24**	**7.17**	**4.94**	**1.97**

Conclusion

- Solve dense face alignment across large poses
- New face profiling algorithm generates dataset with large-pose faces
- Simple yet effective cost function
- Fit dense 3D morphable model with cascaded CNN

Extensions

- Since this method is still based on 3D morphable model, so the parameters has to based on PCA results from training data. Doing PCA over large dataset can be challenging.
- They still use HOG features to do post-processing for landmarks, which is not efficient.
- Potential improvement is to stacking more CNN layers to extract better feature map and then generate structure parameters or facial point positions.
- Instead of feeding new p into the next phase of network, maybe we can take advantage of the rotation parameters, thus iteratively improving the fitting.