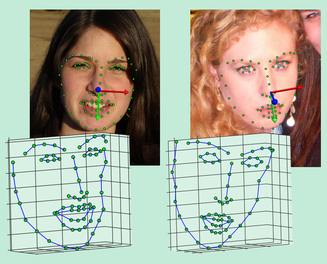
Regressing a 3D Face Shape from a Single Image



Sergey Tulyakov and Nicu Sebe

Department of Information Engineering and Computer Science, University of Trento, Italy
sergey.tulyakov@unitn.it, sebe@disi.unitn.it

Contribution

We present a novel method to estimate a 3D shape of a face from a single image. Contributions:

1. A real-time single-step 3D face shape estimation method
2. 3D invariant feature indexing
3. Localization of actual facial landmarks
4. Face basis computation for accurate head pose estimation

Cascade of 3D Regressors

Our method belongs to cascaded regressor methods (CRMs) and operates in the following way:

$$\Delta \boldsymbol{S}_t = r_t(H_t(I, \hat{\boldsymbol{S}}_{t-1})), \quad (1)$$

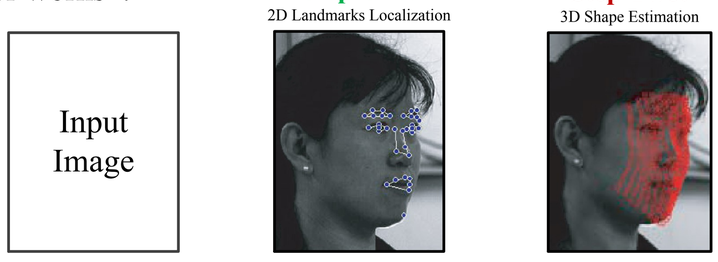
$$\hat{\boldsymbol{S}}_t = \hat{\boldsymbol{S}}_{t-1} + \Delta \boldsymbol{S}_t, \quad (2)$$

where H_t is a feature extraction function, r_t is a regressor learned at the t^{th}-level of the cascade.

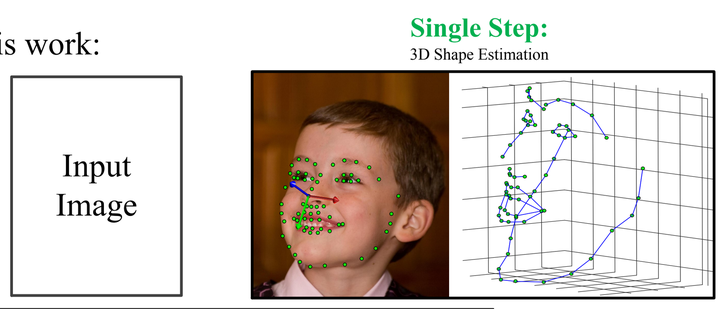
The major difference with previous CRM works is that at every step of the cascade we learn the 3D shape increment: $\Delta\, \boldsymbol{S}_t,\ \hat{\boldsymbol{S}}_t,\ \boldsymbol{S} \in \mathbb{R}^{3\times n}$.

Two Step vs Single Step

Prior works*:



This work:



* - [Jeni et al. 2015], [Cao et al. 2014], [Yi et al. 2013], [Taigman et at. 2014], [Blanz et al. 2003]

Nearest vs Actual Landmarks

We use the actual 3D positions of the points, regardless of their visibility:

$$\lambda \boldsymbol{x}^c = \boldsymbol{A}\boldsymbol{R}\boldsymbol{x}^w + \boldsymbol{t}, \quad (3)$$

Let $\tilde{\boldsymbol{x}}^c = [x^c, y^c, \lambda]$. A training example is then defined as $\{I, \tilde{\boldsymbol{S}}^c\}$, where $\tilde{\boldsymbol{S}}^c = [\tilde{\boldsymbol{x}}_1^c, \tilde{\boldsymbol{x}}_2^c, ..., \tilde{\boldsymbol{x}}_n^c]^T$ is a 3D shape.

During testing, to get the undistorted shape we apply

$$\hat{\boldsymbol{x}}^{wR} = \hat{\boldsymbol{A}}^{-1}(\lambda \hat{\boldsymbol{x}}^c - \boldsymbol{t}), \quad (4)$$

where $\hat{\boldsymbol{A}}$ is the estimated camera matrix.

Nearest visible landmarks:

Actual landmarks:

3D Invariant Indexing

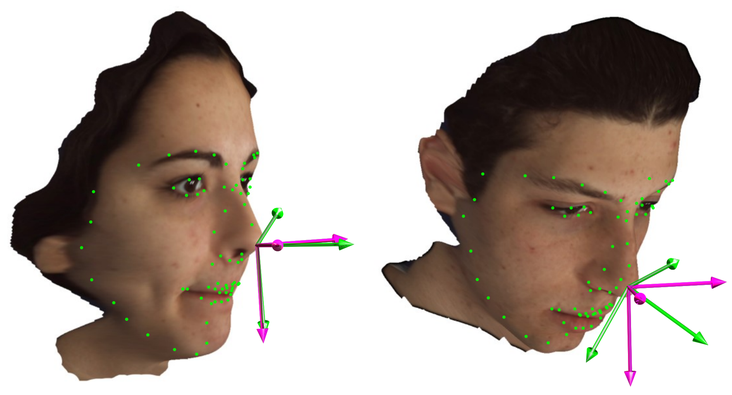
We define a face normal vector $\vec{\boldsymbol{n}}_t$ as the eigenvector with the smallest eigenvalue of the following covariance matrix:

$$\boldsymbol{C}_t = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_{i,t-1} - \bar{\boldsymbol{x}}_{t-1})(\boldsymbol{x}_{i,t-1} - \bar{\boldsymbol{x}}_{t-1})^T, \quad (5)$$

Having the basis $\boldsymbol{F}_t$ and the estimated scaling s we perform feature indexing in the following way:

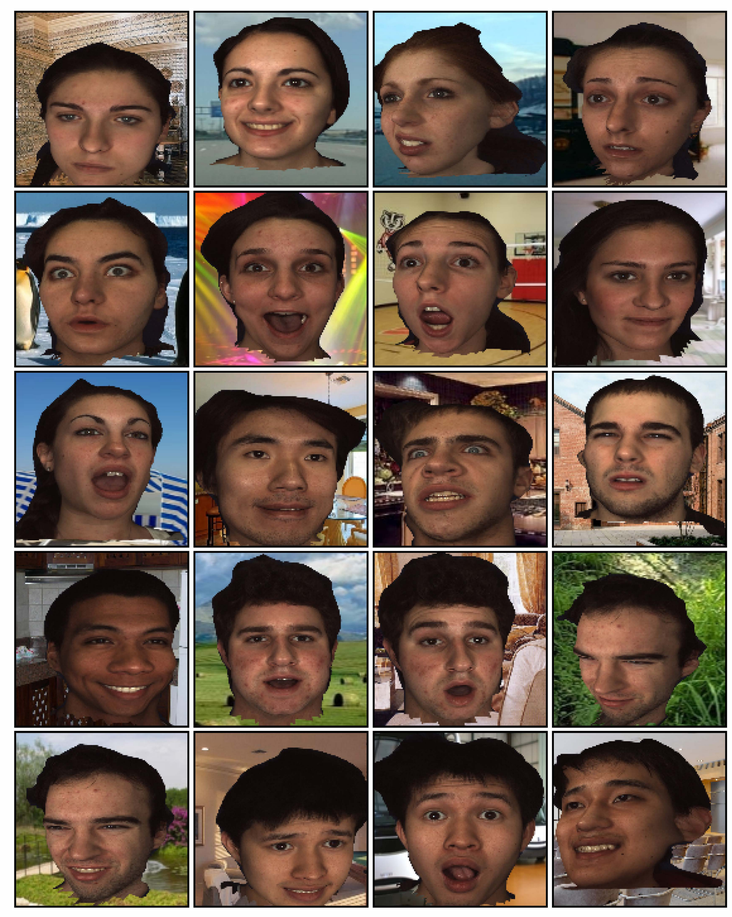
$$\tilde{\boldsymbol{u}}' = \boldsymbol{x}_{k_{\boldsymbol{u}}} + \frac{1}{s}\boldsymbol{F}_t^T \delta\tilde{\boldsymbol{x}}_{\boldsymbol{u}}, \quad (6)$$

The normal vector reveals the head pose of the subject.



Database

We render subjects from BU4DFE under varying head pose ($[-50°, 50°]$ for yaw and tilt) taking images containing no people as a background.



Results

Table 1: 2D and 3D landmarks localization errors.

Method	2D	3D
Kazemi et al. [13]	0.0522	-
Baseline indexing	**0.0515**	0.0610
3D Transform	**0.0515**	0.0607
Basis Transform	0.0518	**0.0592**

Table 2: Head pose estimation results.

Method	Yaw	Tilt
Ours $\hat{\boldsymbol{S}}^c$	0.52	0.54
Zhu and Ramanan [32]	0.76	-
Intraface	0.80	0.51
Ours $\hat{\boldsymbol{S}}^{wR}$	**0.81**	**0.85**

Figure 1: Selected qualitative results for head pose estimation.

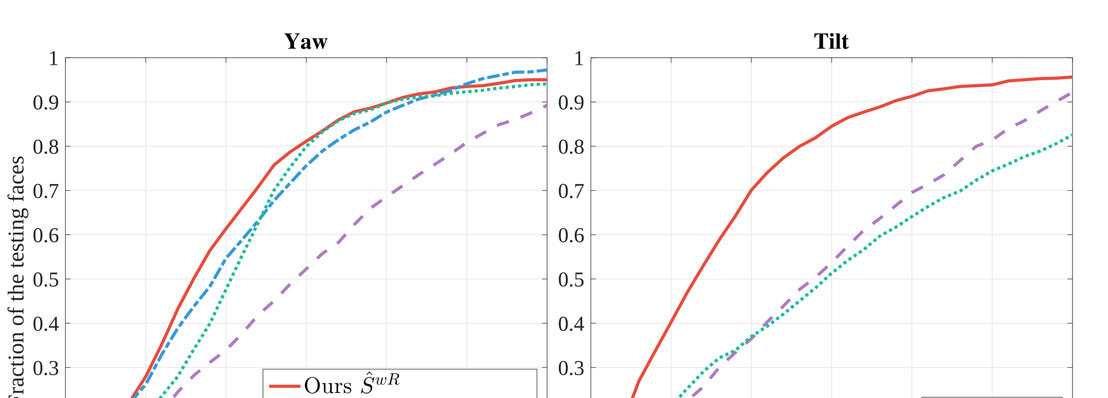


Figure 2: Cumulative error distribution rates for head pose estimation.

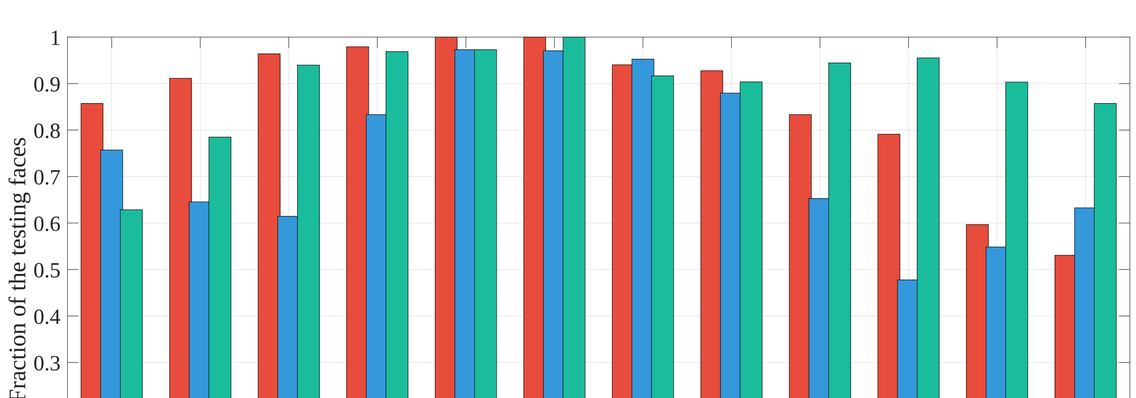


Figure 3: The distribution of the fraction of correctly recognized images.

Qualitative Results