Robust Bayesian Fitting of 3D Morphable Model

Claudia Arellano and Rozenn Dahyot

School of Computer Science and Statistics

Trinity College Dublin, Ireland

CVMP 2013

THE 10TH EUROPEAN CONFERENCE ON VISUAL MEDIA PRODUCTION

7th November 2013

Introduction I

Morphable models are popular to represent families of shapes by a mean shape and eigenvectors such that any member of the family can be reconstructed efficiently as a linear combination of the mean shape and the eigenvectors. The mean and eigenvectors are computed using PCA.

Introduction II

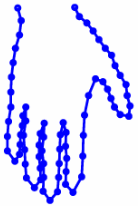
For illustration here, we have used:

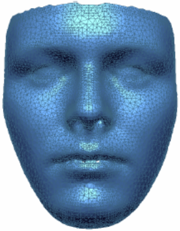
- 2D hand shape model (provided by Tim Cootes)

  `http://www.isbe.man.ac.uk/~bim/data/hand_data.html`

- 3D Basel Face Model (BFM) provided by Basel University:

  `http://faces.cs.unibas.ch/bfm/`

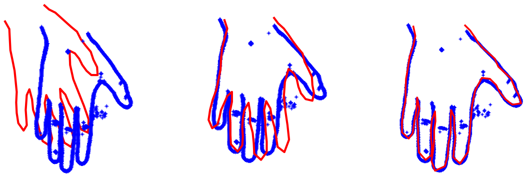




Introduction III

We propose to manipulate the morphable model to align it and fit it to a target shape:

- Rigid transformation: Rotation matrix R and translation $\mathbf{t}$
- Non-Rigid deformation: PCA coefficients $\alpha_1, \alpha_2, \alpha_3, ...\alpha_J$ computed on J eigenvectors



Introduction IV

The unknown to estimate is

$$\Theta = [\mathrm{R}, \mathbf{t}, \alpha_1, \alpha_2, \alpha_3, ...\alpha_J]$$

and we propose to define a robust cost function $\mathcal{C}$ to perform this estimation:

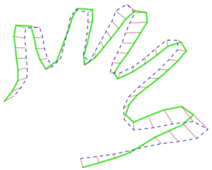
$$\hat{\Theta} = \arg\min \mathcal{C}(\Theta)$$

Related Work I

Matching point clouds $\{u_i\}_{i=1,\cdots,n}$ and $\{u_i^{'}\}_{i=1,\cdots,n'}$ (with $u_j^{'}(\Theta)$)

- Iterative Closest Point (ICP) algorithm is in two steps:
  - Find correspondences $\{(u_i, u_j^{'})\}_{(i,j)\in\mathcal{I}}$
  - Estimate Θ e.g.

$$\widehat{\Theta} = \arg\min_{\Theta} \left\{ \sum_{(i,j)\in\mathcal{I}} \|u_i - u_j^{'}(\Theta)\|^2 \right\}$$



Related Work II

Matching probability density functions $\{u_i\}_{i=1,\cdots,n} \to p(x)$ and $\{u_i^{'}\}_{i=1,\cdots,n'} \to p^{'}(x|\Theta)$

- Kernel correlation:

$$\langle p|p^{'}\rangle = \int p(x)\, p^{'}(x|\Theta)\, dx$$

- Euclidian distance

$$\begin{aligned}\mathcal{L}_2(\Theta) &= \|p - p^{'}\|^2 = \int (p(x) - p^{'}(x|\Theta))^2\, dx \\ &= \|p\|^2 - 2\,\langle p|p^{'}\rangle + \|p^{'}\|^2\end{aligned}$$

Related Work III

About using $\mathcal{L}_2$

- When p and $p^{'}$ are Gaussian mixtures, $\mathcal{L}_2$ can be computed explicitly.
- $\mathcal{L}_2 E$ is a special case of $\mathcal{L}_2$ where one pdf is using Dirac Kernels.
- Current modelings for $\mathcal{L}_2$ and $\mathcal{L}_2 E$ use isotropic covariance matrices,
- no prior information about Θ can be used.

GMM for morphable model I

The GMM is defined by:

$$p^{'}(x|\Theta) = \sum_{i=1}^{n'} G(x; u_i^{'}(\Theta), \Sigma_i^{'})\ \pi_i^{'}$$

- Means:

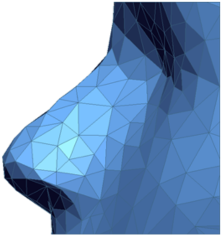
$$u_i^{'}(\Theta) = \mathrm{R}\left(\mu_i + \sum_{j=1}^{J} \alpha_j\ e_{ji}\right) + \mathbf{t}$$

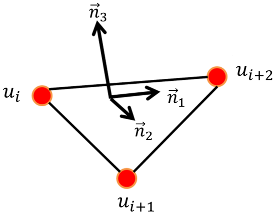
- The weights are $\pi_i^{'} \propto \sqrt{|\Sigma_i^{'}|}$ subject to $\sum_i \pi_i^{'} = 1$.

GMM for morphable model II

- Covariance matrices $\Sigma_i' = \mathrm{Q}\Lambda\mathrm{Q}^T$. Where $\mathrm{Q} = [\vec{n}_1|\vec{n}_2|\vec{n}_3]$ and Λ is:

$$\Lambda = \begin{pmatrix} h_{t1}^2 & 0 & 0 \\ 0 & h_{t2}^2 & 0 \\ 0 & 0 & h^2 \end{pmatrix}$$



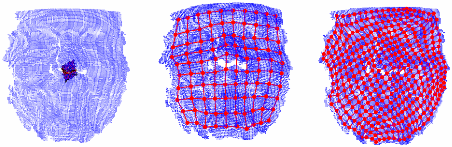


GMM for the target point cloud

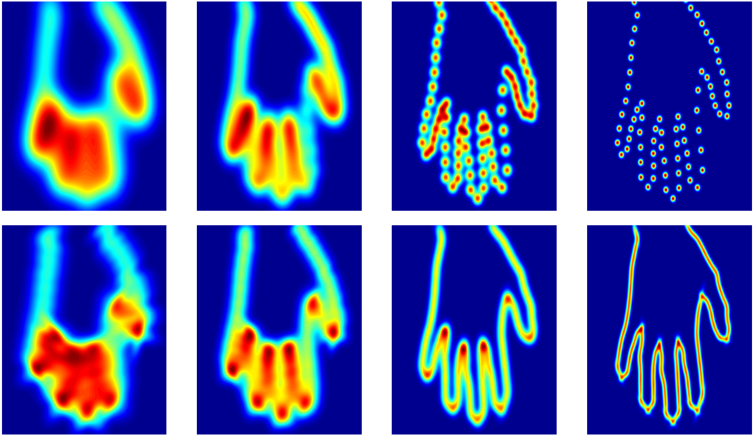
The GMM is defined by using the point cloud $\{u_i\}_{i=1,\cdots,n}$ from the depth camera:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} G(x; u_i, \Sigma_i)$$

with $\Sigma_i = h^2\mathrm{I}$. Alternatively, to down-sample the target point cloud, we use self organising map reducing n and estimating automatically non-isotropic covariance matrices:



Comparison isotropic Vs non-isotropic covariances



(a) $h = 30$ (b) $h = 20$ (c) $h = 10$ (d) $h = 5$

Bayesian $\mathcal{L}_2$ I

- Euclidian distance between GMMs

$$\mathcal{L}_2(\Theta) = \int \left(p(x) - p^{'}(x|\Theta)\right)^2 \, dx$$

- $\mathcal{L}_2(\Theta)$ is interpreted as a negative log likelihood, and Bayesian estimation can then be performed with:

$$\widehat{\Theta} = \arg\min_{\Theta} \left\{ \frac{\mathcal{L}_2(\Theta)}{\sigma_d^2} - \log(p_\Theta(\Theta)) \right\}$$

with p_Θ prior pdf on the parameters of interest.

Bayesian $\mathcal{L}_2$ II

A Normal prior p_Θ with mean 0 is chosen for Θ:

$$p_\Theta(\mathrm{R}, \mathbf{t}, \alpha_1, \cdots, \alpha_J) \propto \prod_{j=1}^{J} G(\alpha_j; 0, \sigma_j^2)$$

$\{\sigma_j\}$ are the eigenvalues of the Morphable model, and a flat prior is chosen for the rigid transformation parameters (i.e. variance very large).

Bayesian $\mathcal{L}_2$ III

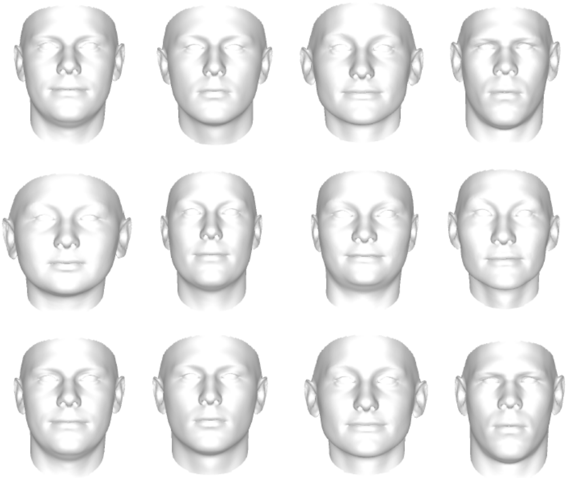
Estimation is performed in two steps

1. the rigid transformation is estimated using the mean shape,
2. the morphable model parameters are then estimated.

This two steps can be repeated until convergence. In practice, only one iteration was necessary. To avoid local optimum, an annealing strategy is used with the (orthogonal) bandwidth h used as temperature.

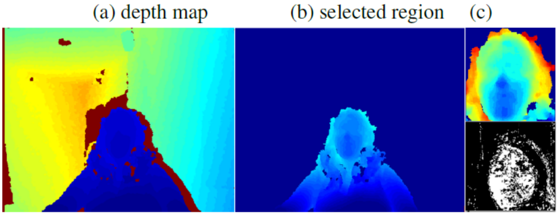
Application to 3D face model fitting to Kinect data I

Assessment on synthetic faces: target (top), random initial guess (middle), estimated (bottom).



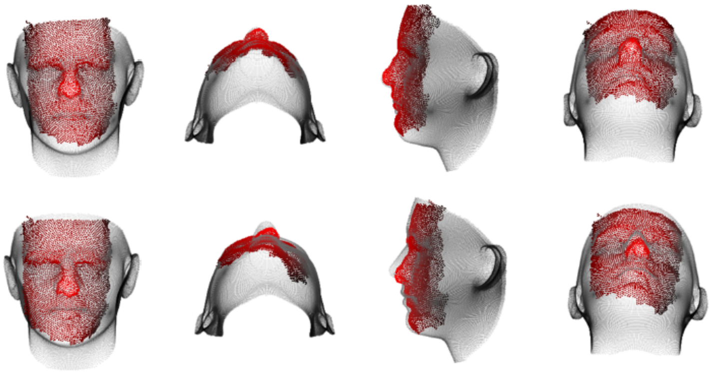
Application to 3D face model fitting to Kinect data II

On real data captured with Kinect sensor, preprocessing is performed to extract a point cloud for the face in view:

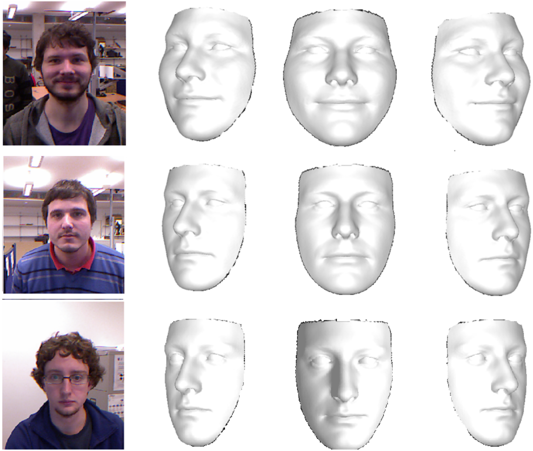


Application to 3D face model fitting to Kinect data III

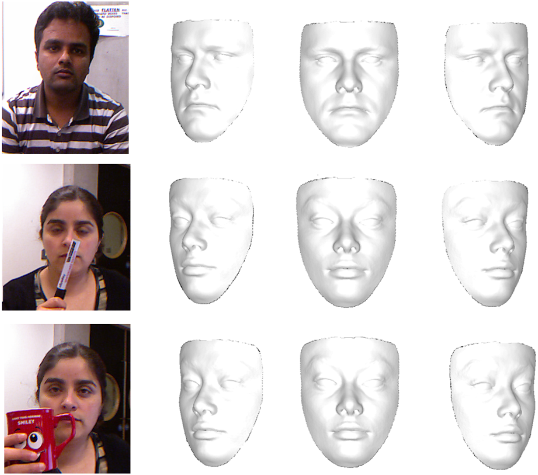
Rigid transformation Estimation:



Application to 3D face model fitting to Kinect data IV



Application to 3D face model fitting to Kinect data V



Application to 3D face model fitting to Kinect data VI

Malahanobis distance $d_{i,j}$ between the estimated parameters of faces

Faces	F1	F2	F3	F4	F5	F6
F1	0	0.3596	0.5374	0.6925	0.7609	0.7815
F2		0	0.3396	0.5041	0.6141	0.6286
F3			0	0.3885	0.4978	0.5155
F4				0	0.5508	0.5475
F5					0	**0.0888**
F6						0

Conclusion

We have proposed:

- a Bayesian framework based on the $\mathcal{L}_2$ distance between GMMs
- GMMs are tailored for representing shapes (e.g. curves or surfaces) using efficiently the covariance matrices.
- The Bayesian framework was tested for fitting 2D and 3D morphable models.

Any Question?