ROBUST TRACKING OF FACIAL FEATURE POINTS WITH 3D ACTIVE SHAPE MODELS

Moritz Kaiser, Dejan Arsić, Shamik Sural, and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München, Germany
moritz.kaiser@tum.de

ABSTRACT

Exact 3D tracking of facial feature points is appealing for many applications in human-machine interaction. In this work a 3D Active Shape Model (ASM) that can be shifted, scaled, and rotated is used to track the points. The efficient Gauss-Newton method is applied to estimate the 3D ASM, rotation, translation, and scale parameters. If the head turns to one side, some points might be occluded but they are still considered for the estimation of the parameters. A robust error norm that reduces (or ideally cancels) the influence of occluded points is applied. With some algebraic transformations the computational cost per frame can be further reduced. The proposed algorithm is evaluated on the basis of the Airplane Behavior Corpus.

Index Terms— Tracking, face recognition, minimization methods, robustness

1. INTRODUCTION

Obtaining 3D information about facial feature points from one monocular video sequence is a challenging task, since only 2D information is available from the video. The 3D tracking is crucial for many applications in human-machine interaction. In contrast to 2D tracking, pose independent emotion and expression recognition can be performed [1, 2]. In this work a 3D Active Shape Model (ASM) is employed to track the points. The efficient Gauss-Newton method is applied to estimate the 3D ASM, rotation, translation and scale parameters, as presented in [3]. However, a problem arises when the head turns to one side and only a profile view is visible. Although some points are occluded, they are still considered for the computation of the parameters and hence corrupt the estimation. A robust error norm that decreases (or ideally cancels) the influence of occluded points is applied. The robust error norm is integrated into the Gauss-Newton method and with some algebraic transformations the computational cost per frame can be further reduced.

Most of the previous works on 3D tracking of facial feature points from a monocular video are based on the entire texture (*appearance*) of the face (see e.g. [4, 5, 6, 7, 8]). However, limiting 3D tracking on a sparse set of facial feature points has several advantages. Some areas are not very distinctive, such as chin, cheeks or forehead and thus just cost computational time. If working with a sparse set of facial feature points, the 3D tracking can be computed at real time and still enough computational time is left for applications that use the location of the points. Furthermore, even if real time is required more sophisticated methods to estimate the 3D positions of the points can be applied. In [9, 2] different approaches to track a sparse set of facial points have been presented. In [10, 11], the application of robust estimation to statistical models is studied for the 2D case.

The paper is organized as follows. In Sec. 2 the 3D ASM is explained and a method to robustly estimate the 3D ASM parameters is described in Sec. 3. Qualitative and quantitative results are given in Sec. 4. Section 5 gives a conclusion and outlines future work.

2. 3D ACTIVE SHAPE MODEL

Following the ISO typesetting standards, matrices and vectors are denoted by bold letters ($\boldsymbol{I}$, $\boldsymbol{x}$) and scalars by normal letters (I, t). In this section the 3D ASM is explained. The 3D ASM parameters determine the position of all facial feature points. In the next section it is explained how the 3D ASM parameters are estimated for each frame. The 3D ASM can be described analogously to the 2D ASM presented in [12]. An object is specified by N feature points. Feature points are manually labeled in all 3D faces of a training database, an example of which is shown in Fig. 1. The 3D point distribution model is constructed as follows. The coordinates of the feature points are stacked into a shape vector

$$\boldsymbol{s} = \left(x_1, y_1, z_1, \ldots, x_N, y_N, z_N\right)^T. \tag{1}$$

The 3D shapes of all training images can be aligned by translating, rotating and scaling them with a Procrustes analysis [12] so that the sum of squared distances between the positions of the feature points is minimized. The mean shape $\overline{\boldsymbol{s}}$ is computed and subtracted from each shape vector. Subsequently, the mean-free shape vectors are written column-wise into a matrix and principal component analysis is applied on that matrix. The eigenvectors corresponding to the N_e largest eigenvalues λ_j are concatenated in a matrix $\boldsymbol{U} = \left[\boldsymbol{u}_1 | \ldots | \boldsymbol{u}_{N_e}\right]$. Thus, a shape can be approximated by only N_e parameters:

$$\boldsymbol{s} \approx \boldsymbol{M}(\boldsymbol{\alpha}) = \overline{\boldsymbol{s}} + \boldsymbol{U} \cdot \boldsymbol{\alpha}, \tag{2}$$

where $\boldsymbol{\alpha}$ is a vector of N_e model parameters and $\overline{\boldsymbol{s}}$ is the mean shape. We denote the 3D ASM by the $3N$-dimensional vector

$$\boldsymbol{M}(\boldsymbol{\alpha}) = \begin{pmatrix} \boldsymbol{M}_1(\boldsymbol{\alpha}) \\ \vdots \\ \boldsymbol{M}_N(\boldsymbol{\alpha}) \end{pmatrix}; \; \boldsymbol{M}_i(\boldsymbol{\alpha}) = \begin{pmatrix} M_{x,i}(\boldsymbol{\alpha}) \\ M_{y,i}(\boldsymbol{\alpha}) \\ M_{z,i}(\boldsymbol{\alpha}) \end{pmatrix}. \tag{3}$$

The parameters $\boldsymbol{\alpha}$ describe the identity of an individual and its current facial expression. Additionally, we assume that the face is translated in x- and y-direction by t_x and t_y, rotated about the y- and z-axis by θ_y and θ_z, and scaled by s. The scaling is a simplified way of simulating a translation in z-direction, where it is assumed that the z-axis comes out of the 2D image plane. Thus, the model we will employ becomes

$$\boldsymbol{x}_i(\boldsymbol{\mu}) = s\boldsymbol{R}(\theta_y, \theta_z)\boldsymbol{M}_i(\boldsymbol{\alpha}) + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \tag{4}$$

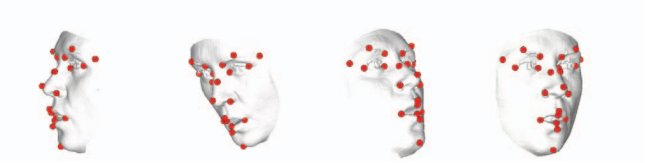

Fig. 1. Multiple views of one 3D facial surface. The facial feature points are manually labeled to build up a 3D ASM. When the facial surface is turned aside some facial feature points are occluded.

where $\boldsymbol{\mu} = (s, \theta_y, \theta_z, t_x, t_y, \boldsymbol{\alpha})$ is the $(5+N_e)$-dimensional parameter vector and

$$\boldsymbol{R}(\theta_y, \theta_z) = \begin{pmatrix} \cos\theta_y \cos\theta_z & \cos\theta_y \sin\theta_z & -\sin\theta_y \\ -\sin\theta_y & \cos\theta_z & 0 \end{pmatrix}. \quad (5)$$

In the next section it is described how the parameter vector $\boldsymbol{\mu}_t$ is estimated for each frame.

3. ROBUST PARAMETER ESTIMATION

The 3D ASM, rotation, translation, and scale parameters determine the position of all facial feature points and thus have to be estimated for each new frame. A set of nonlinear equations including a robust error norm is formed. Since the set of equations is nonlinear, it is linearized with two first order Taylor series expansions and solved iteratively. Some flexible constraints on the 3D ASM are also integrated into the minimization.

Without loss of generality it is assumed that the first frame of a video sequence is acquired at time $t_0 = 0$. The brightness of a point at position $\boldsymbol{x} = (x, y)^T$ at time t is denoted by $I(\boldsymbol{x}, t)$. The position $\boldsymbol{x}_i(\boldsymbol{\mu}_t)$ of one of the N facial feature points at time t depends on the 3D ASM parameter vector $\boldsymbol{\mu}_t$.

The *brightness constancy assumption* implies that at a later time the brightness of the point to track is the same

$$I(\boldsymbol{x}_i(\boldsymbol{\mu}_t), t) = I(\boldsymbol{x}_i(\boldsymbol{\mu}_0), t_0 = 0). \quad (6)$$

For better readability we write $I_i(\boldsymbol{\mu}_t, t) = I_i(\boldsymbol{\mu}_0, 0)$. If $\boldsymbol{\mu}_t$ is estimated at time t, the residual of point i is

$$r_i(\boldsymbol{\mu}_t) = I_i(\boldsymbol{\mu}_t, t) - I_i(\boldsymbol{\mu}_0, 0). \quad (7)$$

In order to obtain $\boldsymbol{\mu}_t$ we need to minimize the energy function

$$E(\boldsymbol{\mu}_t) = \sum_{i=1}^{N} \rho(r_i(\boldsymbol{\mu}_t)). \quad (8)$$

3.1. Robust Error Norm

In Fig. 2 the effect of a robust error norm is demonstrated for a very simple estimation example. Assume a polynomial of second order $ax^2 + bx + c = y$. For a sufficient number of given (x, y)-pairs (red circles) a, b, and c can be estimated by minimizing $\rho(ax^2 + bx + x - y)$. For a general least-squares problem the error function is $\rho(r) = r^2/2$. Figure 2(a) shows some robust error norms. A point with r greater than a certain threshold γ is an outlier, here called *occluded* point, and should not influence the minimization too much. Figure 2(b) shows the estimated polynomials of order 2 for these error norms. Obviously the three points in the upper left corner are *occluded* and should not corrupt the estimation.

This error norm can also be applied for the more complex estimation of the 3D ASM parameters, if some points are occluded. For our problem it is important that occluded points influence the minimization as little as possible and thus the Talwar function given by

$$\rho(r) = \begin{cases} r^2/2 & \text{if } |r| \le \gamma \\ \gamma^2/2 & \text{if } |r| > \gamma \end{cases} \quad (9)$$

was chosen. Tests confirmed that choice.

If the energy function (Eq. 8) is convex, a global minimum can be found by differentiating with respect to $\boldsymbol{\mu}_t$:

$$\left[\frac{\partial}{\partial \boldsymbol{\mu}_t} E(\boldsymbol{\mu}_t)\right]^T = \begin{pmatrix} \frac{\partial}{\partial \mu_1} E(\boldsymbol{\mu}_t) \\ \vdots \\ \frac{\partial}{\partial \mu_{N_p}} E(\boldsymbol{\mu}_t) \end{pmatrix} = \boldsymbol{0}, \quad (10)$$

where N_p is the number of parameters. Applying the chain rule yields the set of nonlinear equations

$$\left[\frac{\partial}{\partial \boldsymbol{\mu}_t} \boldsymbol{I}(\boldsymbol{\mu}_t, t)\right]^T \cdot \boldsymbol{\rho}'(\boldsymbol{r}(\boldsymbol{\mu}_t)) = \boldsymbol{J}^T(\boldsymbol{\mu}_t) \cdot \boldsymbol{\rho}'(\boldsymbol{r}(\boldsymbol{\mu}_t)) = \boldsymbol{0}, \quad (11)$$

where

$$\boldsymbol{J}(\boldsymbol{\mu}_t) = \begin{pmatrix} \frac{\partial}{\partial \mu_1} I_1(\boldsymbol{\mu}_t, t) & \cdots & \frac{\partial}{\partial \mu_{N_p}} I_1(\boldsymbol{\mu}_t, t) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial \mu_1} I_N(\boldsymbol{\mu}_t, t) & \cdots & \frac{\partial}{\partial \mu_{N_p}} I_N(\boldsymbol{\mu}_t, t) \end{pmatrix} \quad (12)$$

is the Jacobian matrix of $\boldsymbol{I}(\boldsymbol{\mu}_t, t)$ and $\boldsymbol{\rho}'(\boldsymbol{r}) = \left(\rho'(r_1), \rho'(r_2), \ldots, \rho'(r_N)\right)^T$. Equation 11 has to be solved for each frame to obtain $\boldsymbol{\mu}_t$.

3.2. Linearization

The preceding set of equations is nonlinear in $\boldsymbol{\mu}_t$. Thus the equations are solved iteratively. In order to obtain a set of linear equations two first order Taylor series expansions are performed.

First linearization: Suppose that $\boldsymbol{\mu}_t^{k+1} = \boldsymbol{\mu}_t^k + \Delta\boldsymbol{\mu}^k$. As in Eq. 6 it is assumed that

$$\boldsymbol{r}(\boldsymbol{\mu}_t^{k+1}) = \boldsymbol{I}(\boldsymbol{\mu}_t^{k+1}, t) - \boldsymbol{I}(\boldsymbol{\mu}_0, 0) = \boldsymbol{0}. \quad (13)$$

Applying the first order Taylor expansion

$$\boldsymbol{I}(\boldsymbol{\mu}_t^{k+1}, t) \approx \boldsymbol{I}(\boldsymbol{\mu}_t^k, t) + \boldsymbol{J}(\boldsymbol{\mu}_t^k)\Delta\boldsymbol{\mu}^k \quad (14)$$

yields

$$\boldsymbol{r}(\boldsymbol{\mu}_t^k) = -\boldsymbol{J}(\boldsymbol{\mu}_t^k)\Delta\boldsymbol{\mu}^k, \quad (15)$$

where the Jacobian matrix is defined as in Eq. 12.

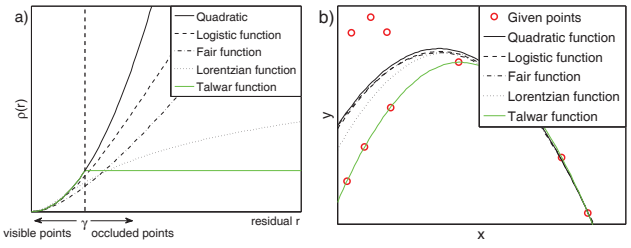

Fig. 2. (a) Plots of several robust error functions [13]. The Talwar function reduces the influence of occluded points most. (b) The effect of these robust error norms on a estimation of a polynomial of order 2.

Second linearization: Equation 11 is linearized with a first order Taylor expansion. Only because of the first linearization (Eq. 15) the second summand can be further transformed by applying the product rule:

$$\boldsymbol{J}^T \cdot \boldsymbol{\rho}'(\boldsymbol{r}(\boldsymbol{\mu}_t^k)) + \frac{\partial}{\partial \boldsymbol{\mu}} \left[\boldsymbol{J}^T \cdot \boldsymbol{\rho}'(\boldsymbol{r}(\boldsymbol{\mu}_t^k)) \right] \cdot \Delta\boldsymbol{\mu}^k \tag{16}$$

$$= \boldsymbol{J}^T \cdot \boldsymbol{\rho}'(\boldsymbol{r}(\boldsymbol{\mu}_t^k)) + \boldsymbol{J}^T \boldsymbol{D} \boldsymbol{J} \cdot \Delta\boldsymbol{\mu}^k = \boldsymbol{0}, \tag{17}$$

where $\boldsymbol{D}$ is a $N \times N$ diagonal matrix with $D_{ii} = \rho''(r_i(\boldsymbol{\mu}_t^k))$. We solve for

$$\Delta\boldsymbol{\mu}^k = -\left[\boldsymbol{J}^T \boldsymbol{D}(\boldsymbol{r}(\boldsymbol{\mu}_t^k)) \boldsymbol{J} \right]^{-1} \boldsymbol{J}^T \cdot \boldsymbol{\rho}'(\boldsymbol{r}_t^k(\boldsymbol{\mu}_t^k)). \tag{18}$$

Note that for each iteration $\boldsymbol{J}$, $\boldsymbol{D}$, and $\boldsymbol{\rho}'$ must be updated.

3.3. Efficient Computation

The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\mu}_t^k)$ (Eq. 12) must be recomputed at each iteration k. The numerical computation of $\frac{\partial}{\partial \boldsymbol{\mu}} \boldsymbol{I}(\boldsymbol{\mu}_t, t)$ at each time step is time consuming. Therefore, the derivative of the image with respect to the parameters is decomposed via chain rule into an easily computable spatial derivative of the image and a derivative of the parametric model with respect to the parameters which can be solved analytically. Additionally, if, as in Eq. 6, it is assumed that $\frac{\partial}{\partial \boldsymbol{x}} I_i(\boldsymbol{\mu}_t, t) = \frac{\partial}{\partial \boldsymbol{x}} I_i(\boldsymbol{\mu}_0, 0)$, we obtain

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial}{\partial \boldsymbol{x}} I_1(\boldsymbol{\mu}_0, 0) \frac{\partial}{\partial \boldsymbol{\mu}} \boldsymbol{x}_1(\boldsymbol{\mu}_t) \\ \vdots \\ \frac{\partial}{\partial \boldsymbol{x}} I_N(\boldsymbol{\mu}_0, 0) \frac{\partial}{\partial \boldsymbol{\mu}} \boldsymbol{x}_N(\boldsymbol{\mu}_t) \end{pmatrix}. \tag{19}$$

The numerical derivative $\frac{\partial}{\partial \boldsymbol{x}} I_i(\boldsymbol{\mu}_0, 0)$ and the analytical derivative $\frac{\partial}{\partial \boldsymbol{\mu}} \boldsymbol{x}_i(\boldsymbol{\mu}_t)$ can be computed offline. Thus, for each iteration we compute $\boldsymbol{J}$, $\boldsymbol{D}$, and $\boldsymbol{\rho}'$ and subsequently $\Delta\boldsymbol{\mu}^k$ is calculated according to Eq. 18. In comparison to numerically computing $\frac{\partial}{\partial \boldsymbol{\mu}} \boldsymbol{I}(\boldsymbol{\mu}_t, t)$, this reduces the computation time by roughly a factor of 2. For the threshold γ that separates visible from occluded points we apply an annealing strategy. We start with a large γ, i.e. all points are considered for the minimization, and then decrease it for each iteration until $\gamma_{\min} = 0.1$.

3.4. Convexity

In Eq. 10 we assumed that the energy function E is convex. In practice, this is often not the case. Thus, the starting values $\boldsymbol{\mu}$ have to be sufficiently close to the global minimum so that the method does not converge to only a local minimum. This can be achieved by applying the widely-used coarse-to-fine refinement. A Gaussian pyramid is created for each new frame. The parameters are computed iteratively for the coarsest level. Then, the parameters are taken as starting values for the next finer level for which the computation is performed again, and so on.

The starting values at the lowest level are the parameters computed for the last frame. For a sufficiently high frame rate usually the parameters of two consecutive frames do not differ too much and thus the starting values are also sufficiently close to the global minimum.

3.5. Constraints

Several constraints on the parameters are added to prevent unrealistic results. Solving a set of nonlinear equations with constraints is a complex task and thus we integrate the constraints as further equations. Thereby, the constrained minimization problem is converted into an unconstrained minimization problem which can be solved as presented above.

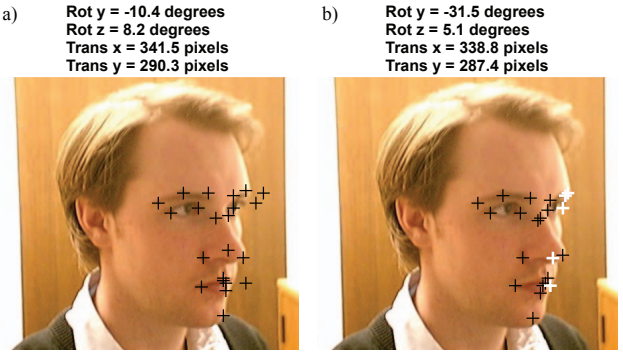

Fig. 3. (a) Tracking of facial feature points with a 3D ASM without robust error norm. (b) Tracking of facial feature points with a robust error norm. White crosses depict points that are detected as occluded. Those points are not considered for the parameter estimation and thus the parameters can be computed correctly.

We require the rotation θ_y, θ_z and also the 3D ASM parameters $\boldsymbol{\alpha}$ not to become too large:

$$\frac{\theta_y}{\sigma_{\theta_y}^2} = -\frac{\Delta\theta_y}{\sigma_{\theta_y}^2}; \quad \frac{\theta_z}{\sigma_{\theta_z}^2} = -\frac{\Delta\theta_z}{\sigma_{\theta_z}^2}; \quad \frac{\alpha_j}{\sigma_{\alpha_j}^2} = -\frac{\Delta\alpha_j}{\sigma_{\alpha_j}^2}. \tag{20}$$

We also require all parameters not to change too much from one frame to another

$$\frac{\Delta\mu_j}{\sigma_{\Delta\mu_j}^2} = 0. \tag{21}$$

The constraints can easily be appended at the bottom of $\boldsymbol{J}$ as further equations that the parameters $\Delta\boldsymbol{\mu}$ have to satisfy. In order to have the right balance between equations and constraints, the constraints are multiplied by the variance of the residual denoted by σ_N^2. The variances $\sigma_{\alpha_j}^2$ have already been computed at the principal component analysis performed for the 3D ASM and the other variances have to be estimated.

4. RESULTS

We built the 3D ASM from the Bosphorus Database [14] where we used 2761 3D images from 105 individuals. The images are labeled with $N = 22$ facial feature points, as depicted in Fig. 1 as an example. Four pyramid levels were employed for the Gaussian image pyramid. On an Intel® Core™ 2 Quad processor and 4GB working memory the computation of the parameters took on average 9.94 ms per frame.

4.1. Qualitative Evaluation

It can be qualitatively confirmed that the 22 facial feature points are tracked reliably in 2D webcam recordings with changing position, rotation and expressions. Figure 3 illustrates the effect of the robust error norm on the tracking results for a sample image from a webcam recording. The information box at the top shows the computed rotation and position parameters. If all feature points are equally considered for the parameter estimation, the results can be erroneous, as depicted in Fig. 3(a). Figure 3(b) shows that with a robust error norm occluded points can be detected properly (white crosses). Those points are not considered for the parameter estimation and thus the parameters are computed correctly.

4.2. Quantitative Evaluation

The proposed algorithm was evaluated on the basis of the Airplane Behavior Corpus (ABC) that was presented in [15]. The corpus contains a total of 11.5 h video. The scenes show closeup views of different individuals having various emotions. It includes fast movements, the individuals sometimes look aside so that facial feature points are occluded and the face is occasionally covered by hectic hand gestures. Therefore, the ABC is a suitable setting for testing the proposed algorithm's capability to cope with occlusions. The PAL standard was used for the videos, i.e., an image resolution of 576×720 pixels and 25 frames per second. Scenes of three individuals having a total length of 40 min were selected from the corpus. Every 25th frame of those scenes was manually annotated with 22 landmarks and those landmarks were used as ground truth. For each labeled frame the pixel displacement $d_i = \sqrt{\Delta x^2 + \Delta y^2}$ between the location estimated by our algorithm and the ground truth was computed. The pixel displacement was averaged over the $N = 22$ facial feature points: $D = \frac{1}{N}\sum_{i=1}^{N} d_i$. In Table 1, the pixel displacement averaged over all labeled frames of a sequence for each individual is depicted. For comparison the simple Kanade-Lucas-Tomasi [16] feature tracker and the method presented in [3] have been implemented and also tested with the dataset. For all three individuals the pixel displacement could be reduced. The improvement is most obvious for the videos of individual 3, since these contain a considerable amount of occlusions.

The results are also comparable with the results reported recently by other authors. The authors of [9] tested their multi-stage hierarchical models on a dataset of 10 sequences with 100 frames per sequence. Considering that their test sequences had less than half of our image resolution their pixel displacement is similar to ours. Also the pixel displacement that [7] reported for their testing database of 2 challenging video sequences is comparable to ours. In general, our method is computationally considerably cheaper than those methods. Additionally, our approach includes the consideration of completely occluded facial feature points and was also tested on a dataset with occlusions. It is also important to notice that our 3D ASM works with a relatively small number of facial feature points, since the 3D faces of the Bosphorus Database are labeled with only 22 landmarks. It is expected that a 3D ASM with more points would further improve the tracking results.

D [pixels]	individual 1	individual 2	individual 3
[16]	28.07	25.45	43.34
[3]	21.32	22.98	39.75
our algorithm	19.19	18.03	19.71

Table 1. Pixel displacement averaged over all video sequences from individual 1, 2, and 3.

5. CONCLUSION AND FUTURE WORK

A method for robust 3D tracking of facial feature points from a monocular video sequence is presented. Facial feature points are linked with a 3D ASM. The 3D ASM parameters are estimated with the efficient Gauss-Newton method. It is shown that the effect of occluded points that would normally perturb the estimation can be canceled with a robust error norm. Quantitative results based on the Airplane Behavior Corpus, where many occlusions occur, show that the proposed method is able to outperform other approaches. In contrast to other methods published recently our system is able to cope with completely occluded points while being computationally less intensive.

In our ongoing research we will analyze the effect of using gradient images and Gabor filtered images to further improve the tracking results. We have also planned to employ a second camera that could be easily integrated into the estimation scheme.

Acknowledgments

This work has been partially funded by the European Projects FP-033812 (AMIDA) and FP-214901 (PROMETHEUS) as well as by Alexander von Humboldt Fellowship for experienced researchers.

6. REFERENCES

[1] B. Gong, Y. Wang, J. Liu, and X. Tang, "Automatic facial expression recognition on a single 3d face by exploring shape deformation," in *ACM Multimedia*, 2009, pp. 569–572.

[2] J. Chen, M. Kim, Y. Wang, and Qiang Ji, "Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition," in *CVPR*, 2009, pp. 2655–2662.

[3] M. Kaiser, D. Arsić, S. Sural, and G. Rigoll, "Tracking of facial feature points by combining singular tracking results with a 3d active shape model," in *VISAPP*, 2010.

[4] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *SIGGRAPH*, 1999, pp. 187–194.

[5] J. A. Paterson and A. W. Fitzgibbon, "3d head tracking using non-linear optimization," in *BMVC*, 2003, vol. 2, pp. 609–618.

[6] J. Sung, T. Kanade, and D. Kim, "Pose robust face tracking by combining active appearance models and cylinder head models," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 260–274, 2008.

[7] H. Fang, N. Costen, D. Cristinacce, and J. Darby, "3d facial geometry recovery via group-wise optical flow," in *FG*, 2008, pp. 1–6.

[8] E. Muñoz, J. M. Buenaposada, and L. Baumela, "A direct approach for efficiently tracking with 3d morphable models," in *ICCV*, 2009, pp. 1–8.

[9] Y. Tong, Y. Wang, Z. Zhu, and Q. Ji, "Robust facial feature tracking under varying face pose and facial expression," *Pattern Recognition*, vol. 40, no. 11, pp. 3195–3208, 2007.

[10] M. Rogers and J. Graham, "Robust active shape model search," in *ECCV (4)*, 2002, pp. 517–530.

[11] B. Theobald, I. Matthews, and S. Baker, "Evaluating error functions for robust active appearance models," in *FG*, 2006, pp. 149–154.

[12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models — their training and application," *CVIU*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[13] D. P. O'Leary, "Robust regression computation using iteratively reweighted least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, pp. 466–480, 1990.

[14] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, "Bosphorus database for 3d face analysis," in *BIOID*, 2008, pp. 47–56.

[15] B. Schuller, M. Wimmer, D. Arsić, G. Rigoll, and B. Radig, "Audiovisual behavior modeling by combined feature spaces," in *ICASSP*, April 2007, vol. 2, pp. 733–736.

[16] C. Tomasi and T. Kanade, "Detection and tracking of point features," Tech. Rep., Carnegie Mellon University, April 1991.