TRACKING OF FACIAL FEATURE POINTS BY COMBINING SINGULAR TRACKING RESULTS WITH A 3D ACTIVE SHAPE MODEL

Moritz Kaiser, Dejan Arsić, Shamik Sural and Gerhard Rigoll
Technische Universität München, Arcisstr. 21, 80333 Munich, Germany
{moritz.kaiser, arsic, sural, rigoll}@tum.de

Keywords: Facial feature tracking, 3D Active Shape Model, Face pose estimation.

Abstract: Accurate 3D tracking of facial feature points from one monocular video sequence is appealing for many applications in human-machine interaction. In this work facial feature points are tracked with a Kanade-Lucas-Tomasi (KLT) feature tracker and the tracking results are linked with a 3D Active Shape Model (ASM). Thus, the efficient Gauss-Newton method is not solving for the shift of each facial feature point separately but for the 3D position, rotation and the 3D ASM parameters which are the same for all feature points. Thereby, not only the facial feature points are tracked more robustly but also the 3D position and the 3D ASM parameters can be extracted. The Jacobian matrix for the Gauss-Newton optimization is split via chain rule and the computations per frame are further reduced. The algorithm is evaluated on the basis of three handlabeled video sequences and it outperforms the KLT feature tracker. The results are also comparable to two other tracking algorithms presented recently, whereas the method proposed in this work is computationally less intensive.

1 INTRODUCTION

Accurate 3D tracking of 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 appealing for many applications in human-machine interaction. In contrast to 2D tracking, information about the 3D head position and the 3D head movement can be extracted and it is also possible to perform pose independent emotion and expression recognition (Gong et al., 2009). Statistical models are a powerful tool for solving this task. Often, models that are based on the entire appearance of faces are employed (Cootes et al., 1998; Faggian et al., 2008). Nevertheless, for many applications it is sufficient to track only a certain number of selected facial feature points. The advantage of only tracking a sparse set of points is that it is less complicated and computationally less intensive.

1.1 Conception of this Work

For model-free 2D tracking the Kanade-Lucas-Tomasi (KLT) (Tomasi and Kanade, 1991) feature tracker generates reasonable results and it is computationally efficient. The KLT feature tracker is a local approach that computes the shift of each facial feature point separately. Often due to noise, illumination changes or weakly textured neighborhood the KLT feature tracker loses track of some feature points. An unrealistic constellation of points is the result as depicted in Figure 1. In this work we combine the tracking results for each single point with a 3D Active Shape Model (ASM). The 3D ASM guarantees a certain structure between the locally tracked points. Hence, the efficient Gauss-Newton optimization (Bertsekas, 1999) is not solving for the 2D shift of several facial feature points separately but for the parameters of a 3D ASM and rotation, translation and scale parameters which are the same for all feature points. Thereby, not only the facial feature points are tracked more robustly but also additional information like the parameters of the 3D ASM for emotion and expression recognition and the 3D head position and movement are extracted.

1.2 Previous Work

Model-free tracking algorithms are not suitable for extracting 3D information from a monocular video sequence. Thus, model-based methods are applied. A popular approach for tracking is to use statistical models based on the entire appearance of the face. The method described in (Cristinacce and Cootes, 2006a) uses texture templates to locate and track facial features. In (Cristinacce and Cootes, 2006b) texture templates are used to build up a constrained local appearance model. The authors of (Sung et al., 2008) propose a face tracker that combines an Active Appearance Model (AAM) with a cylindrical head model to be able to cope with cylindrical rotation. In (Fang et al., 2008) an optical flow method is combined with a statistical model, as in our approach. However, the optical flow algorithm by (Brox et al., 2004) is applied which is computationally more intensive than a Gauss-Newton optimization for only several facial feature points. All those methods rely on statistical models that are based on the entire texture of the face which is computationally more intensive than statistical models based on only a sparse set of facial feature points.

The authors of (Heimann et al., 2005) employed a statistical model which is based only on a few feature points, namely a 3D Active Shape Model, which is directly matched to 3D point clouds of organs for detection. The matching takes from several minutes up to several hours. (Tong et al., 2007) use a set of facial feature points whose spatial relations are modeled with a 2D hierarchical shape model. Multi-state local shape models are used to model small movements in the face.

This paper is organized as follows. In Section 2 our algorithm is presented, where first a tracking framework for a generic parametric model is explained and then the 3D parametric model we employ is described. Quantitative and qualitative results are given in Section 3. Section 4 gives a conclusion and outlines future work.

2 PROPOSED ALGORITHM

In this section, a tracking framework is illustrated, that can be applied if the motion can be described by a parametric model. The least-squares estimation of the parameters of a generic parametric model is explained. Subsequently, the parametric model that is employed in this work, namely a 3D ASM, is presented. It is explained how the parameter estimation, that has to be performed for each frame, can be carried out more efficiently. Furthermore, some constraints on the parameter estimation are added.

Following the ISO typesetting standards, matrices and vectors are denoted by bold letters (I, x) and scalars by normal letters (I, t).

2.1 Tracking Framework for Parametric Models

Assume that the first frame of a video sequence is acquired at time $t_0 = 0$. Vector $x_i = (x, y)^T$ describes the location of point i in a frame. The location changes over time according to a *parametric model* $x_i(\mu_t)$, where μ_t is the *parameter vector* at time t. The set of points for which the parameter vector is supposed to be the same is denoted by $\mathcal{T} = \{x_1, x_2, \ldots, x_N\}$. It is assumed that the position of the N points and thus also the parameters μ_0 for the first frame are known. The brightness value of point $x_i(\mu_0)$ in the first frame is denoted by $I(x_i(\mu_0), t_0 = 0)$.

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

$$I(x_i(\mu_0), 0) = I(x_i(\mu_t), t). \qquad (1)$$

For better readability we write $I_i(\mu_0, 0) = I_i(\mu_t, t)$. The energy function that has to be minimized at every time step t for each location i in order to obtain μ_t is

$$E_i(\mu_t) = \big(I_i(\mu_t, t) - I_i(\mu_0, 0)\big)^2. \qquad (2)$$

Knowing μ_t at time t, we only need to compute $\Delta\mu$ in order to determine $\mu_{t+\tau} = \mu_t + \Delta\mu$. The energy function becomes

$$E_i(\Delta\mu) = \big(I_i(\mu_t + \Delta\mu, t + \tau) - I_i(\mu_0, 0)\big)^2. \qquad (3)$$

If τ is sufficiently small, $I_i(\mu_t + \Delta\mu, t + \tau)$ can be linearized with a Taylor series expansion considering only the first order term and ignoring the second and higher order terms

$$I_i(\mu_t + \Delta\mu, t + \tau)$$
$$\approx I_i(\mu_t, t) + \frac{\partial}{\partial\mu} I_i(\mu_t, t) \cdot \Delta\mu + \tau \frac{\partial}{\partial t} I_i(\mu_t, t), \qquad (4)$$

with $\frac{\partial}{\partial\mu} I = \big(\frac{\partial}{\partial\mu_1} I, \frac{\partial}{\partial\mu_2} I, \ldots, \frac{\partial}{\partial\mu_{N_p}} I\big)$. By approximating

$$\tau \frac{\partial}{\partial t} I_i(\mu_t, t) \approx I_i(\mu_t, t + \tau) - I_i(\mu_t, t), \qquad (5)$$

Equation (3) becomes

$$E_i(\Delta\mu)$$
$$= \big(\frac{\partial}{\partial\mu} I_i(\mu_t, t) \cdot \Delta\mu + I_i(\mu_t, t + \tau) - I_i(\mu_0, 0)\big)^2. \qquad (6)$$

The error is defined as

$$e_i(t + \tau) = I_i(\mu_t, t + \tau) - I_i(\mu_0, 0). \qquad (7)$$

If we assume that E_i is convex, the minimization problem can be solved by

$$\frac{\partial}{\partial\Delta\mu} E_i$$
$$= \big(\frac{\partial}{\partial\mu} I_i(\mu_t, t) \cdot \Delta\mu + e_i(t + \tau)\big) \cdot \frac{\partial}{\partial\mu} I_i(\mu_t, t) = 0. \qquad (8)$$

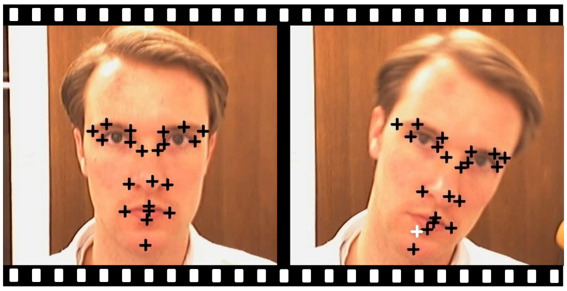

Figure 1: The facial feature points of the left image are tracked by a KLT feature tracker. In the right image it can be seen that one facial feature point (white cross) drifted off.

Taking into account all points i, the system of linear equations can be solved for $\Delta\mu$ by

$$\Delta\mu = -(J^T J)^{-1} J^T e(t+\tau), \tag{9}$$

where

$$e(t+\tau) = \begin{pmatrix} e_1(t+\tau) \\ \vdots \\ e_N(t+\tau) \end{pmatrix}. \tag{10}$$

J denotes the $N \times N_p$ Jacobian matrix of the image with respect to μ:

$$J = \begin{pmatrix} \frac{\partial}{\partial\mu} I_1(\mu_t, t) \\ \vdots \\ \frac{\partial}{\partial\mu} I_N(\mu_t, t) \end{pmatrix}, \tag{11}$$

where N is the number of points with equal parameters and N_p is the number of parameters.

For the KLT feature tracker there are only two parameters to estimate, namely the shift in x- and y-direction. $\frac{\partial}{\partial x} I_i(\mu_t, t)$ can be estimated numerically very efficiently with e.g. a Sobel filter. The set $\mathcal{T}$ for which equal parameters are assumed is a squared neighborhood around the feature point that should be tracked. For the case of tracking several facial feature points, each facial feature point would have to be tracked separately. Although the KLT feature tracker works already quite well, single points to track might drift off due to a weakly textured neighborhood, noise, illumination changes or occlusion. Figure 1 (left) shows the first frame of a video sequence with 22 feature points and Figure 1 (right) illustrates the tracked points several frames later. Most of the feature points are tracked correctly but one (marked with a white cross) got lost by the feature tracker. The human viewer can immediately see that the spacial arrangement of the facial feature points (white and black crosses) is not typical for a face. The structure between the facial feature points can be ensured by a parametric model. In the case of landmarks on a face a 3D ASM seems appropriate.

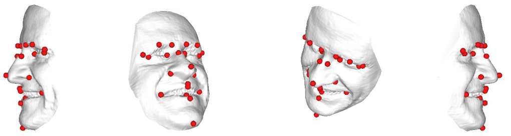

Figure 2: Multiple views of one 3D facial surface. The facial feature points are manually labeled to build up a 3D ASM.

2.2 3D Active Shape Model

The 2D ASM was presented in (Cootes et al., 1995). The 3D ASM can be described analogously. An object is specified by N feature points. The feature points are manually labeled in N_I training faces, as shown in Figure 2 exemplary for one 3D facial surfaces. The 3D point distribution model is constructed as follows. The coordinates of the feature points are stacked into a shape vector

$$s = \left(x_1, y_1, z_1, \ldots, x_N, y_N, z_N\right). \tag{12}$$

The 3D shapes of all training images can be aligned by translating, rotating and scaling them with a Procrustes analysis (Cootes et al., 1995) so that the sum of squared distances between the positions of the feature points is minimized. 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 $U = \left[u_1 | \ldots | u_{N_e}\right]$. Thus, a shape can be approximated by only N_e parameters:

$$s \approx M(\alpha) = \bar{s} + U \cdot \alpha, \tag{13}$$

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

$$M(\alpha) = \begin{pmatrix} M_1(\alpha) \\ \vdots \\ M_N(\alpha) \end{pmatrix}; \; M_i(\alpha) = \begin{pmatrix} M_{x,i}(\alpha) \\ M_{y,i}(\alpha) \\ M_{z,i}(\alpha) \end{pmatrix}. \tag{14}$$

The parameters α 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

$$x_i(\mu) = sR(\theta_y, \theta_z) M_i(\alpha) + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \tag{15}$$

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

$$R(\theta_y, \theta_z) = \begin{pmatrix} \cos\theta_y \cos\theta_z & \cos\theta_y \sin\theta_z & -\sin\theta_y \\ -\sin\theta_y & \cos\theta_z & 0 \end{pmatrix}. \quad (16)$$

We directly insert the 3D ASM of Equation 15 into our tracking framework for parametric models of Equation 9 in order to estimate the parameters μ_t in a least-squares sense for each frame.

2.3 Efficient Parameter Estimation

Since J of Equation 9 depends on time-dependent quantities, it must be recomputed for each frame. Especially for the model of Equation 15 the numerical computation of $\frac{\partial}{\partial\mu} I(\mu_t, t)$ 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 Equation 1, it is assumed that $\frac{\partial}{\partial x} I_i(\mu_t, t) = \frac{\partial}{\partial x} I_i(\mu_0, 0)$, we obtain

$$J = \begin{pmatrix} \frac{\partial}{\partial x} I_1(\mu_0, 0) \frac{\partial}{\partial \mu} x_1(\mu_t) \\ \vdots \\ \frac{\partial}{\partial x} I_N(\mu_0, 0) \frac{\partial}{\partial \mu} x_N(\mu_t) \end{pmatrix}. \quad (17)$$

The numerical derivative $\frac{\partial}{\partial x} I_i(\mu_0, 0)$ and the analytical derivative $\frac{\partial}{\partial \mu} x_i(\mu_t)$ can be computed offline. Thus, for each frame the current estimation of μ has to be plugged in the Jacobian matrix and the new parameters are computed according to Equation 9.

2.4 Constraints on the 3D ASM

In order to prevent unrealistic results several constraints on the parameters are added. We require the rotation θ_y, θ_z and also the 3D ASM parameters α not to become too large:

$$\frac{\theta_y}{\sigma^2_{\theta_y}} = -\frac{\Delta\theta_y}{\sigma^2_{\theta_y}}; \; \frac{\theta_z}{\sigma^2_{\theta_z}} = -\frac{\Delta\theta_z}{\sigma^2_{\theta_z}} \quad (18)$$

$$\frac{\alpha_j}{\sigma^2_{\alpha_j}} = -\frac{\Delta\alpha_j}{\sigma^2_{\alpha_j}}. \quad (19)$$

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

$$\frac{\Delta\mu_j}{\sigma^2_{\Delta\mu_j}} = 0. \quad (20)$$

The constraints can easily be appended at the bottom of J as further equations that the parameters $\Delta\mu$ have to satisfy. In order to have the right balance between all equations, the equations of (9) are multiplied by $1/\sigma^2_N$, where σ^2_N is the variance of the error e_i. Note that $\sigma^2_{\alpha_j}$ can be directly taken from the principal component analysis performed for the 3D ASM, while the other variances have to be estimated.

2.5 Coarse-To-Fine Refinement

The coarse-to-fine refinement is a widely-used strategy to deal with larger displacements for optical flow and correspondence estimation. In practice, E_i is often not convex as we have assumed in Equation 8. Especially, if there is a large displacement of the facial feature points between two consecutive frames, the least-squares minimization converges to only a local minimum instead of the global minimum. This problem can be - at least partially - overcome by applying a Gaussian image pyramid that has to be created for each new frame. The computation of the parameter vector is performed 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.

3 EXPERIMENTAL RESULTS

We built the 3D ASM from the Bosphorus Database (Savran et al., 2008) where we used 2761 3D images from 105 individuals. The images are labeled with $N = 22$ facial feature points (Figure 2). Four pyramid levels were employed for the Gaussian image pyramid. On a 3GHz Intel® Pentium® Duo-Core processor and 3GB working memory the computation of the parameters took on average 9.83 ms per frame.

3.1 Quantitative Evaluation

The proposed algorithm was evaluated on the basis of three video sequences, each lasting roughly one minute. In each video sequence an individual performs motions, such as translation, rotation about the y- and z-axis and it also changes the facial expression. The image resolution of the video sequence is 640×480 pixels. Every 10th frame 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

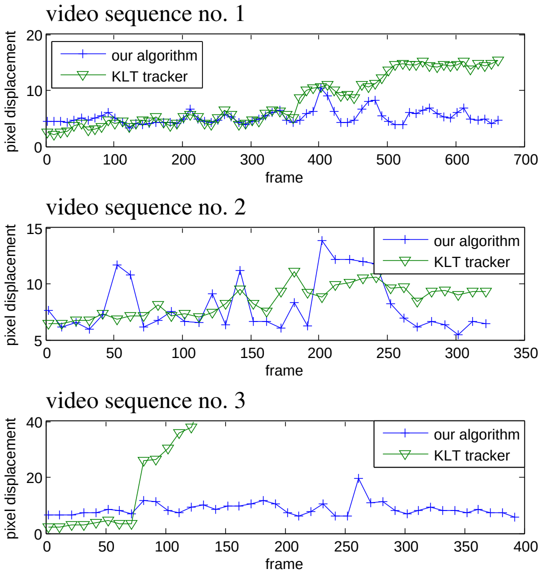

Figure 3: Pixel displacement for three test video sequences.

the $N = 22$ facial feature points:

$$D = \frac{1}{N}\sum_{i=1}^{N} d_i. \tag{21}$$

Figure 3 shows the results for the three video sequences. As a baseline system the KLT feature tracker was chosen. Observe that already for the first frame there is a small pixel displacement indicating that a portion of the displacement is owed to the fact that the annotation cannot always be performed unequivocally. For higher frame numbers the KLT feature tracker loses track of some points and thus the pixel displacement accumulates. Our algorithm is able to prevent this effect and shows robust performance over the whole sequence. In Table 1, the pixel displacement averaged over all labeled frames of a sequence is depicted. Our algorithm outperforms the KLT tracker. The results are also comparable with the results reported recently by other authors. (Tong et al., 2007) tested their multi-stage hierarchical models on a dataset of 10 sequences with 100 frames per sequence. Considering that their test sequences had half of our image resolution their pixel displacement is similar to ours. Also the pixel displacement that (Fang et al., 2008) reported for their testing database of 2 challenging video sequences is comparable to ours. However, both methods are computationally considerably more intensive than our tracking scheme.

3.2 Qualitative Evaluation

Figure 4 shows two sample frames of video sequence no. 1. (The 3 video sequences are available together in a single file as supplementary material.) The information box on the lower left corner shows position details. For the left image it can be observed that the rotation about the z-axis is detected correctly and in the right image the rotation about the y-axis is estimated properly. Generally, it can be qualitatively confirmed that not only the points are tracked reliably in the 2D video sequence but also 3D motion and expressions can be extracted from the sequence. 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. In contrast, current 2D face databases have more landmarks, some of them roughly hundred points. It is expected that a 3D ASM with more points would further improve the tracking and parameter estimation results.

Table 1: Pixel displacement averaged over a whole video sequence.

video sequence	no. 1	no. 2	no. 3
KLT tracker	8.07	8.45	43.34
our algorithm	5.19	8.03	8.71

4 CONCLUSIONS AND FUTURE WORK

A method for 3D tracking of facial feature points from a monocular video sequence is presented. The facial feature points are tracked with a simple Gauss-Newton estimation scheme and the results are linked with a 3D ASM. Thus, the efficient Gauss-Newton minimization computes the 3D position, rotation and 3D ASM parameters instead of the shift of each feature point separately. It is demonstrated how the amount of computations that must be performed for each frame can be further reduced. Results show that the algorithm tracks the points reliably for rotation, translations, and facial expressions. It outperforms the KLT feature tracker and delivers results comparable to two other methods published recently, 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 integrate a weighting matrix that depends on the rotation parameters to reduce the influence of facial feature points that might disappear.

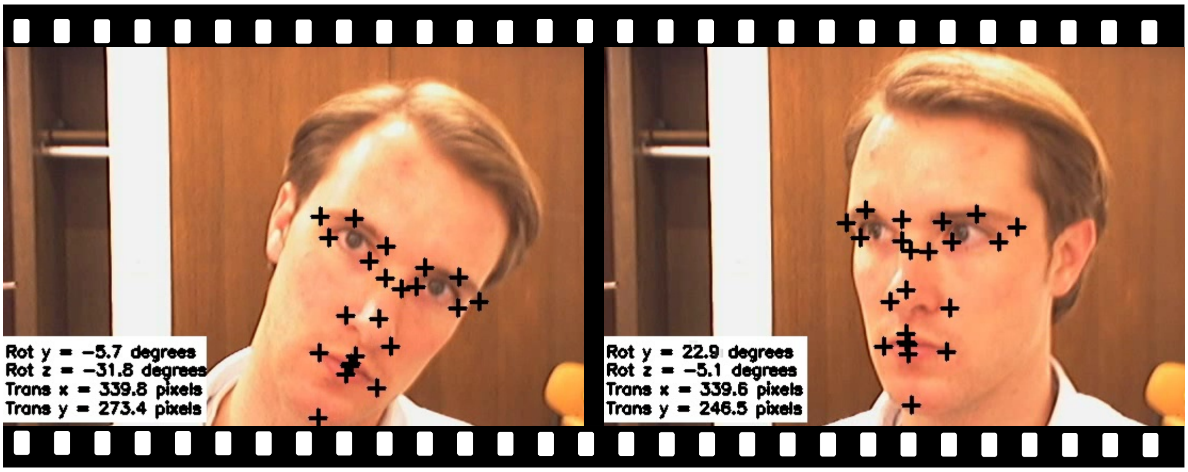

Figure 4: Features tracked by our algorithm. The information box shows rotation and translation parameters.

REFERENCES

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, 2nd edition.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *ECCV (4)*, pages 25–36.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *ECCV (2)*, pages 484–498.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models — their training and application. *CVIU*, 61(1):38–59.

Cristinacce, D. and Cootes, T. F. (2006a). Facial feature detection and tracking with automatic template selection. In *FG*, pages 429–434.

Cristinacce, D. and Cootes, T. F. (2006b). Feature detection and tracking with constrained local models. In *BMVC*, pages 929–938.

Faggian, N., Paplinski, A. P., and Sherrah, J. (2008). 3d morphable model fitting from multiple views. In *FG*, pages 1–6.

Fang, H., Costen, N., Cristinacce, D., and Darby, J. (2008). 3d facial geometry recovery via group-wise optical flow. In *FG*, pages 1–6.

Gong, B., Wang, Y., Liu, J., and Tang, X. (2009). Automatic facial expression recognition on a single 3d face by exploring shape deformation. In *ACM Multimedia*, pages 569–572.

Heimann, T., Wolf, I., Williams, T. G., and Meinzer, H.-P. (2005). 3d active shape models using gradient descent optimization of description length. In *IPMI*, pages 566–577.

Savran, A., Alyüz, N., Dibeklioglu, H., Çeliktutan, O., Gökberk, B., Sankur, B., and Akarun, L. (2008). Bosphorus database for 3d face analysis. In *BIOID*, pages 47–56.

Sung, J., Kanade, T., and Kim, D. (2008). Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal of Computer Vision*, 80(2):260–274.

Tomasi, C. and Kanade, T. (1991). Detection and tracking of point features. Technical report, Carnegie Mellon University.

Tong, Y., Wang, Y., Zhu, Z., and Ji, Q. (2007). Robust facial feature tracking under varying face pose and facial expression. *Pattern Recognition*, 40(11):3195–3208.