

Real-Time 3D Face Tracking with Mutual Information and Active Contours

Giorgio Panin and Alois Knoll

Technical University of Munich, Informatics Faculty
Boltzmannstrasse 3, 85748 Munich, Germany
{panin, knoll}@in.tum.de

Abstract. We present a markerless real-time, model-based 3D face tracking methodology. The system combines two robust and complimentary optimization-based strategies, namely active contours and mutual information template matching, in order to obtain real-time performances for full 6dof tracking. First, robust head contour estimation is realized by means of the Contracting Curve Density algorithm, effectively employing local color statistics separation for contour shape optimization. Afterwards, the 3D face template is robustly matched to the underlying image, through fast mutual information optimization. Off-line model building is done using a fast modeling procedure, providing a unique appearance model for each user. Re-initialization criteria are employed in order to obtain a complete and autonomous tracking system.

Key words: Real-time Face Tracking, Nonlinear Optimization, 3D Template Matching, Mutual Information, Active Contours, Local Statistics Contour Matching, 3D Face Modeling

1 Introduction

Real-time 3D face tracking is an important problem in computer vision. Several approaches have been proposed and developed with different model definitions, concerning shape, degrees of freedom, use of multiple appearance/shading templates, use of natural facial features, etc.; a careful choice of model complexity is a critical issue for real-time tracking, often forcing to resort to approximate solutions, in terms of the output provided to the end user or to subsequent processing modules.

The focus of this paper concerns fast and reliable *6dof* face pose estimation and tracking in real-time, based on a hierarchical integration of two robust high-level visual modalities. In order to motivate our approach, we first consider here related state-of-the-art methodologies, from the available literature on the subject.

The system proposed in [1] employs a robust multi-layer fusion of different visual cues in the hierarchical framework named IFA [2], proceeding from coarse to accurate visual modalities, and providing the result from the top-level tracker

as output; in this work, simple template and feature point models are used at the high levels.

Other approaches, using natural face features detection and tracking in a monocular setting, have been presented in [3][4]. In particular, in [3] a 3D face model is fitted by matching features across subsequent frames, with an approach combining RANSAC [5] and Particle Filters [6] under frame-to-frame epipolar constraints. [4] combines on-line and off-line information by matching local features and optimizing a robust least-squares global cost function. Although this approach can provide a better stability, precision and speed, due to the use of local optimization techniques, the joint use of online and offline information may pose additional choices, concerning the overall cost function parameters, and the models required for tracking.

Template-based approaches also show to be well-suited for face tracking tasks, concerning both precision and robustness issues. In [7], shape and appearance parameters are optimized at the same time under a 2D piece-wise affine deformation model; although the 3D pose of the face is not directly provided by the estimation algorithm, it can be subsequently estimated from the set of 2D parameters, at the price of quite complex computations involving a Kalman-filter based methodology [8]. The approach [9], closer to ours, directly employs a full 3D shape and appearance template, with multiple appearance models and robust nonlinear least-squares optimization. The 3D shape of the face is used together with multiple texture models of the user under different light conditions; these informations are provided off-line.

Generally speaking, the need for multiple light/appearance models can constitute a major drawback of most template-based approaches, forcing the user to provide off-line several reference images for the textured model. This motivation leads to consider more general similarity functions for template matching, in order to provide robustness of the system while using a few, or even a single appearance model, during all of the tracking task.

Motivated by the previous considerations, in this paper we propose a novel model-based approach fusing two complimentary visual modalities, namely active contour head tracking and face template matching; both modules are based on fast, local optimization of robust cost functions, and at the same time require off-line a minimal set of modeling operations. The paper is organized as follows: Section 2 describes the framework for providing the textured template. Section 3 states the overall tracking algorithm. Section 4 deals with the solution to head contour fitting, employing a fast implementation of the CCD algorithm, while Section 5 describes robust template matching optimizing mutual information similarity index. Finally, Section 6 provides experimental results obtained from a complete implementation of the proposed system.

2 Face Template Modeling

Off-line, given two photos of the subject to be tracked in front and profile views (Fig. 1), we adapt a generic 3D head model by employing part of the model-

fitting methodology given in [10], where a few pre-selected feature lines (e.g. eyes, lips, cheek contour) are marked onto the photos, and a least-squares fitting of the mesh is performed.



Fig. 1. Automatic 3D model adaptation.

After shape adaptation, the same pictures provide the texture map, and the resulting model is shown in Fig. 1 on the left side. From the complete model, deformable parts (eyes and mouth) are removed, and a limited area covering front and part of the side view is extracted, obtaining the textured template shown on the right side.

Almost all of the described operations can be automatically performed, so that with the proposed method a new template is built by a non-expert user within very short time.

3 Overall Tracking Methodology

3.1 Geometry Representation

We express the rigid transformation between camera and head coordinate frames with the homogeneous (4×4) transformation matrix T

$$T = \begin{bmatrix} R & \theta_T \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

with R the rotation matrix, and $\theta_T = [X, Y, Z]^T$ the translation vector. 3D rotations are expressed in terms of XYZ Euler angles

$$R(\theta_R) = R_x(\alpha) R_y(\beta) R_z(\gamma); \theta_R = [\alpha, \beta, \gamma]^T \quad (2)$$

with R_x, R_y, R_z the respective rotation matrices. Overall head pose is then given by the 6-vector θ

$$\theta = [\theta_R, \theta_T] \quad (3)$$

and a body point in homogeneous coordinates ${}^b\bar{\mathbf{x}}$ transforms to camera coordinates ${}^c\bar{\mathbf{x}}$ according to

$${}^c\bar{\mathbf{x}} = T(\theta)^b \bar{\mathbf{x}} \quad (4)$$

Concerning the intrinsic camera parameters, we adopt a simple pinhole model with focal length f , so that points in camera space ${}^c\mathbf{x} = [x_{1c}, x_{2c}, x_{3c}]^T$ project to the screen $\mathbf{y} = [y_1, y_2]^T$ as

$$y_1 = f \frac{x_{1c}}{x_{3c}} + \frac{r_{y1}}{2}; \quad y_2 = -f \frac{x_{2c}}{x_{3c}} + \frac{r_{y2}}{2} \quad (5)$$

with r_{y1}, r_{y2} the horizontal and vertical image resolution.

By assuming camera calibration to be performed off-line, we can indicate the overall body-to-screen mapping with

$$\mathbf{y} = P(\mathbf{x}, \theta) \quad (6)$$

3.2 On-line 3D Pose Estimation

In a template matching framework, local optimization of the similarity function alone could be used for estimating the full 6dof pose of the face, starting from a pre-defined hypothesis θ_0 .

However, due to the 3D/2D projection (6), any similarity function shows a narrow convergence region, and non-quadratic behavior outside a limited area around the optimum. This results in a difficult optimization problem that may take many cost function evaluations, or get stuck in local optima.

Therefore, as stated in the introduction, in the present approach pose estimation has been split into two sub-tasks, by hierarchically estimating translation and rotation parameters θ_R, θ_T

1. First, translations θ_T are estimated via robust head contour tracking, using the CCD algorithm and an elliptical contour model. This module also provides a coarse estimate of the in-plane rotation angle γ , which can be used for initializing the next module
2. Afterwards, full rotation parameters θ_R are estimated by maximizing mutual information between the 3D textured template and the underlying image; translation parameters θ_T are kept fixed to the value given by the previous module, while rotations are initialized with the ones $\theta_R(t-1)$ obtained from the previous timestep.

Both modules employ suitable criteria in order to detect tracking failures, and automatically re-initialize the system in case of loss. In order to re-initialize the contour tracker, a generic face detector [11] is employed; for the template matching module, rotations are re-initialized to zero (front view).

4 Head contour tracking module

The first step of our methodology consists of robustly estimating the translation parameters θ_T . For this purpose, we first model the head external contour as a

planar ellipse (Fig. 2) with proper axes length. The degrees of freedom of the contour pose directly correspond to 4 of the 6 pose parameters θ , namely 3D translation and in-plane rotation angle

$$\Phi = (\gamma, \theta_T) \quad (7)$$

Contour tracking is performed through the real-time version of the Contracting Curve Density (CCD) algorithm [12][13]. This method aims at fitting a parametric curve model to the image stream, by making use of local self-adapting separation criteria, and a two-step optimization procedure akin to Expectation Maximization.

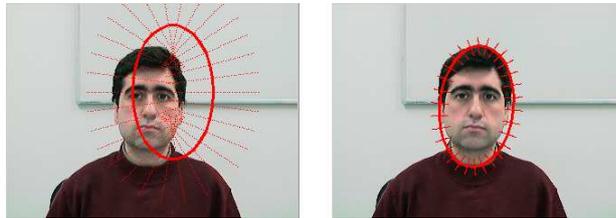


Fig. 2. Contour model fitting to the head boundary, before and after 12 steps of CCD optimization. Sample points used by the algorithm are shown.

Contour estimation at each frame makes also use of prior knowledge of pose parameters in a Bayesian framework, which in the re-initialization phase is obtained from a generic face detector [11], while in subsequent frames comes from the previous estimation result. A dynamic model, although not required, can further improve tracking performance through Kalman filtering [13]; however, for the present tracking task this improvement has shown not to be necessary.

In CCD, the following two-step procedure is iterated, refining the estimate and, at the same time, gaining certainty on the posterior density (that *contracts* to a small, unimodal pdf), until a satisfactory result is found.

4.1 Step 1: Compute local color statistics around the head contour

Current mean and covariance matrix ($\mathbf{m}_\Phi, \Sigma_\Phi$) are set to the initial values (Φ_0, Σ_Φ^*), with statistics modeled by multivariate Gaussian distributions.

A set of points along normal directions for each sample position k on both sides of the curve is taken (Fig. 2) up to a distance h related to the current parameter uncertainty Σ_Φ , and normal segments to the base shape are uniformly sampled along into L sample points

$$\mathbf{x}_{kl} = \mathbf{x}_{k0} + d_l \mathbf{n}_k \quad (8)$$

with $d_l = d(\mathbf{x}_{kl}, \mathbf{x}_{k0})$ the perpendicular distance.

According to the length h , two weighting functions w_1, w_2 and fuzzy *assignment* values a_1, a_2 of each color pixel to the respective region, are computed for each contour side

$$\begin{aligned} a_1(d) &:= \frac{1}{2} \left[\operatorname{erf} \left(\frac{d}{\sqrt{2}\sigma} \right) + 1 \right]; \quad a_2 := 1 - a_1 \\ w_{1/2}(d) &:= \left(\frac{a_{1/2}(d) - 0.5}{0.5} \right)^6 \left[e^{-d^2/2\hat{\sigma}^2} - e^{-\gamma_2} \right]^+ \end{aligned} \quad (9)$$

where all of the constants inside are properly selected in advance.

In order to collect local color statistics, base sample points \mathbf{x}_{kl} are projected according to the current hypothesis $\Phi = \mathbf{m}_\Phi$ (Fig. 2) by applying (6)

$$\mathbf{y}_{kl} = P(\mathbf{x}_{kl}, [0, 0, \Phi]) \quad (10)$$

and local, weighted RGB statistics up to the second order are collected

$$\nu_{ks}^{(0)} = \sum_{l=1}^L w_{kls}; \quad \nu_{ks}^{(1)} = \sum_{l=1}^L w_{kls} \mathbf{I}_{kl}; \quad \nu_{ks}^{(2)} = \sum_{l=1}^L w_{kls} \mathbf{I}_{kl} \mathbf{I}_{kl}^T \quad (11)$$

with $s = 1, 2$ the curve side, and \mathbf{I}_{kl} the observed pixel colors at image locations \mathbf{y}_{kl} .

Color statistics of order $o = 0, 1, 2$ along the contour are blurred with exponential filtering both in space and time

$$\begin{aligned} \nu_{ks}^{(o)} &= \sum_{k_1=1}^K b(k - k_1) \nu_{k_1s}^{(o)} \\ \tilde{\nu}_{ks}^{(o)}(t) &= \tau \nu_{ks}^{(o)} + (1 - \tau) \tilde{\nu}_{ks}^{(o)}(t - 1) \end{aligned} \quad (12)$$

and the resulting quantities are finally normalized, giving a set of expected color values and covariance matrices for each sample position and contour side

$$\bar{\mathbf{I}}_k^{(s)} = \frac{\tilde{\nu}_{ks}^{(1)}(t)}{\tilde{\nu}_{ks}^{(0)}(t)}; \quad \bar{\Sigma}_k^{(s)} = \frac{\tilde{\nu}_{ks}^{(2)}(t)}{\tilde{\nu}_{ks}^{(0)}(t)} - \bar{\mathbf{I}}_k^{(s)} \bar{\mathbf{I}}_k^{(s)T} \quad (13)$$

Normalized local statistics are finally indicated with $S := (\bar{\mathbf{I}}_k^{(s)}, \bar{\Sigma}_k^{(s)})$.

4.2 Step 2: Optimize contour shape

In the second step of CCD, estimated statistics (13) are used in order to evaluate the contour fit function, its gradient and the Hessian matrix for parameter update. We first write the MAP cost function as

$$E = E_1(\Phi, \mathbf{m}_\Phi^*, \Sigma_\Phi^*) + E_2(\Phi, S) \quad (14)$$

where

$$\begin{aligned} E_1 &:= \frac{1}{2} (\Phi - \mathbf{m}_\Phi^*)^T (\Sigma_\Phi^*)^{-1} (\Phi - \mathbf{m}_\Phi^*) + Z(\Sigma_\Phi^*) \\ E_2 &:= \sum_{k,l} \left[\frac{1}{2} (\mathbf{I}_{kl} - \hat{\mathbf{I}}_{kl})^T \hat{\Sigma}_{kl}^{-1} (\mathbf{I}_{kl} - \hat{\mathbf{I}}_{kl}) + Z(\hat{\Sigma}_{kl}) \right] \end{aligned} \quad (15)$$

are, respectively, the prior log-probability of the pose, and the log-likelihood of observed colors \mathbf{I}_{kl} w.r.t. the expected statistics $(\hat{\mathbf{I}}_{kl}, \hat{\Sigma}_{kl})$, with normalizing terms Z .

Expected colors $\hat{\mathbf{I}}_{kl}$ and local covariances $\hat{\Sigma}_{kl}$ are given by the fuzzy assignment (9) to each contour side

$$\begin{aligned} \hat{\mathbf{I}}_{kl} &= a_1(d_l) \bar{\mathbf{I}}_k^{(1)} + (1 - a_1(d_l)) \bar{\mathbf{I}}_k^{(2)} \\ \hat{\Sigma}_{kl} &= a_1(d_l) \bar{\Sigma}_k^{(1)} + (1 - a_1(d_l)) \bar{\Sigma}_k^{(2)} \end{aligned} \quad (16)$$

where only the dependence of $\hat{\mathbf{I}}_{kl}$ on Φ is considered when computing the derivatives.

In order to implement the Gauss-Newton optimization step, the gradient of E w.r.t. Φ and the Gauss-Newton matrix are computed

$$\begin{aligned} \nabla_\Phi E_2 &= - \sum_{k,l} \left(\nabla_\Phi \hat{\mathbf{I}}_{kl} \right)^T \hat{\Sigma}_{kl}^{-1} (\mathbf{I}_{kl} - \hat{\mathbf{I}}_{kl}) + (\Sigma_\Phi^*)^{-1} (\Phi - \mathbf{m}_\Phi^*) \\ H_\Phi E_2 &= \sum_{k,l} \left(\nabla_\Phi \hat{\mathbf{I}}_{kl} \right)^T \hat{\Sigma}_{kl}^{-1} \nabla_\Phi \hat{\mathbf{I}}_{kl} + (\Sigma_\Phi^*)^{-1} \end{aligned} \quad (17)$$

with

$$\nabla_\Phi \hat{\mathbf{I}}_{kl} := \left(\bar{\mathbf{I}}_k^{(1)} - \bar{\mathbf{I}}_k^{(2)} \right) (\nabla_\Phi a_1(d_l))^T \quad (18)$$

and the parameter mean and covariance update is given by

$$\begin{aligned} \mathbf{m}_\Phi^{new} &= \mathbf{m}_\Phi - (H_\Phi E)^{-1} \nabla_\Phi E \\ \Sigma_\Phi^{new} &= c \Sigma_\Phi + (1 - c) (H_\Phi E)^{-1} \end{aligned} \quad (19)$$

where $c = 0.25$ is an exponential decay coefficient.

The whole procedure (Steps 1 and 2) is iterated with updated parameters until convergence. In this module, detection for tracking loss is performed by looking at the final value of E (14), indicating the overall color statistics separation between the two regions (inside and outside) along the contour line. If the separation index decreases below a suitable threshold, the CCD tracker is re-initialized.

5 Face template matching module

With the given appearance model, the template matching module aims at finding the orientation θ_R maximizing *consistency* between projected template points and underlying image pixels, as shown in Fig. 3.

Using mutual information as consistency measure allows to accommodate a more general model of variation between expected and observed appearance, in presence of different lighting, shading, and other nonlinear effects such as partial occlusions. As a result, by using mutual information we can keep a constant appearance model working through a wide range of rotations, at the same time gaining robustness with respect to occlusions and image noise.

For this task, we indicate with $M(\mathbf{x})$ the gray-scale intensity of points \mathbf{x} over the textured surface model of Fig. 1.



Fig. 3. Template matching result after mutual information optimization.

5.1 Histogram-based estimator

For real-time purposes, we use histogram-based mutual information [14]. Histograms have large computational advantages over other methods like as kernel density estimators [15], which generally exhibit an $O(N^2)$ complexity and make use of costly real-valued functions (e.g. exponential kernels). For this reason, histogram-based entropy estimators are widely used in medical image registration problems, where the volume of data is usually very large; on the other hand, in a real-time application sample sizes have to be kept relatively small, and therefore the finite-sample *bias* of the estimator cannot be anymore neglected.

An analysis of statistical properties (bias, variance) of the histogram entropy estimator for continuous densities has been given in [14]; in particular, the bias

contribution due to finite sample size N (termed *N-bias*) can be approximated by a second-order Taylor expansion of the density estimator around the “true” value p_{ij} , and the resulting correction for the joint entropy $\hat{H}(x_1, x_2)$ is given by

$$\hat{H}(x_1, x_2) = - \sum_{i,j} \left(\frac{h_{ij}}{N} \log \frac{h_{ij}}{N} \right) + \frac{C^2 - 1}{2N} + \log(\Delta x_1 \Delta x_2) \quad (20)$$

where h_{ij} is the joint (template-image) histogram, with C^2 cells of size $(\Delta x_1, \Delta x_2)$.

In order to get a good statistical behaviour, a suitable number of histogram cells is required, for which we adopt a simple rule of thumb, by dividing each dimension into $I = J = \sqrt[3]{N}$ bins.

It is worth to note here that both bias and variance of $\hat{H}(\mathbf{x})$, like for any statistical estimator, grow up very fast with space dimension, so that the sample size N required to achieve the same performance in higher dimension (e.g. using a color template) would be too high for a real-time implementation, and at the same time not necessary. By using grayscale intensities, a bi-variate histogram allows a minimum sample size of $N = 1000$ sample points to be reliably employed.

5.2 Mutual Information optimization

In our framework, a set of N points \mathbf{x}_k is randomly drawn with uniform probability from the visible area of the template M at pose $\theta_0 = [\theta_R^0, \theta_T^0]$; the set of visible points is pre-computed, by removing back and self-occluded surface points¹.

Because of random sampling, the cost function will have a random component (noisy optimization); this indeed helps in avoiding local optima that for a small size N may be present.

For furtherly improving convergence and speed properties, the optimization is performed in a multi-resolution cascade, by Gaussian filtering both the model texture map and the underlying image; moreover, the sample set N is also kept lower for low resolution images, while increasing it for higher resolutions.

At each pose hypothesis θ_R , the random sample \mathbf{x}_k is projected onto the image I according to (6), to pixel coordinates $\mathbf{y}_k = P(\mathbf{x}_k, \theta)$, and the corresponding intensity pairs $(m_1, v_1), \dots, (m_N, v_N)$, with

$$m_k = M(\mathbf{x}_k); v_k = I(\mathbf{y}_k) \quad (21)$$

are collected into the joint histogram $h_{ij} = h(m, v)$.

Afterwards, marginal histograms $h(m)$ and $h(v)$ are computed by summation over rows and columns

$$h(m) \triangleq h_{i,\bullet} = \sum_{j=1}^C h_{ij}; h(v) = h_{\bullet,j} = \sum_{i=1}^C h_{ij} \quad (22)$$

¹ Since the previous pose θ_R^0 is close to the current one, the visible point set can be computed just once, before the optimization loop

and bias-corrected mutual information is finally given by

$$\hat{I}_{m,v}(x_1, \dots, x_N, \theta_T^0, \theta_R) = \sum_{i,j} \frac{h_{ij}}{N} \log \frac{h_{ij}N}{h_{i,\bullet}h_{\bullet,j}} - \frac{(C-1)^2}{2N} \quad (23)$$

In order to optimize (23) w.r.t. θ_R , we apply here the standard Nelder-Mead *simplex* algorithm [16]. This algorithm is well-suited for low dimensional problems, and at the same time performs well in presence of a noisy cost function.

After optimization, tracking loss is detected by setting a lower threshold on the output mutual information value. By optimizing only over rotational degrees of freedom θ_R , this module yields stable and reliable results within less than 100 function evaluations, ensuring real-time performances on common platforms.

6 Experimental results

Our face modeling and tracking system has been successfully applied to on-line tracking experiments. We show here results from a video sequence of 1550 frames, using the model of Fig. 1; in absence of ground-truth for this experiment, accuracy of the result can be visually evaluated by looking at the matching between superimposed template and image pixels (Fig. 4).



Fig. 4. Face tracking in presence of different light conditions, partial occlusion and wide rotation angles.

Throughout all of the sequence, tracking has been performed in real-time at a frame rate of 10-15 fps on a common Desktop PC with 3GHz CPU, using a standard FireWire camera with an image resolution of 640x480; due to the nature of both estimation modules, the input images are directly used without the need for any preprocessing operation (noise filtering, features enhancement etc.); results have been on-line recorded during the tracking task.

Fig. 4 shows a selection of frames, with the results of both contour tracking and template matching, and the frontal part of the textured model superimposed at estimated 3D pose. A single appearance template has been sufficient for successfully tracking the user throughout almost the whole sequence, apart from two cases of too wide lateral rotations; after losing the track, the system automatically recovered within few frames, without the need for any manual intervention.

In particular, the top rows of Fig. 4 show robustness of the system under strong light changes and partial occlusion; thanks to the use of mutual information as robust matching index, the tracker did not lose accuracy of pose estimation despite dark lighting (frames 506, 659), light switching (frame 550), and occlusions (frame 772). The same observation concerns contour tracking that, by employing locally self-adaptive color separation criteria, is able to overcome abrupt changes in color statistics both inside and outside the enclosed region.

The frames displayed in the bottom rows show tracking performance under wider face rotations; wide in-depth rotations (frame 1400) usually pose more difficult problems with respect to in-plane rotations (frame 1356) for face tracking systems, in particular due to the shading difference with respect to the frontal pose, and the smaller visible template area. Despite these difficulties, mutual information template matching performs well also in these cases, where different shadings have the effect of reducing the maximum similarity value, but not its location in pose space.

7 Conclusions

We presented a real-time 3D face tracking system combining two complementary and robust model-based visual modalities; possible applications for the system include hands-free Human-Computer Interfaces, 3D gaze tracking, real-time video augmentation, and many others as well.

Planned developments, in the direction of increasing speed, robustness and versatility, include the following ones: augmenting the system with further visual modalities such as color and motion cues; a parallel (or pipeline) implementation scheme for an efficient integration of all modules; exploration of suitable data fusion methodologies for this system, incorporating dynamic models and multiple hypotheses for multi-target tracking in presence of crowded situations; introduction of deformable models on top of the rigid template, in order to detect lips expression and eye gaze.

References

1. Toyama, K.: Look, ma — no hands!’ hands-free cursor control with real-time 3d face tracking (1998)
2. Toyama, K., Hager, G.: Incremental focus of attention for robust visual tracking (1996)
3. Lu, L., Dai, X.T., Hager, G.: A particle filter without dynamics for robust 3d face tracking. In: CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 5, Washington, DC, USA, IEEE Computer Society (2004) 70
4. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26** (2004) 1391–1391
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24** (1981) 381–395
6. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision* **29** (1998) 5–28
7. Matthews, I., Baker, S.: Active appearance models revisited. Technical Report CMU-RI-TR-03-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2003)
8. Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real-time combined 2d+3d active appearance models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2004) 535 – 542
9. Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models (1999)
10. Park, I.K., Zhang, H., Vezhnevets, V., Choh, H.K.: Image-based photorealistic 3-d face modeling. In: FGR. (2004) 49–56
11. Viola, P.A., Jones, M.J.: Robust real-time face detection. In: ICCV. (2001) 747
12. Hanek, R., Beetz, M.: The contracting curve density algorithm: Fitting parametric curve models to images using local self-adapting separation criteria. *Int. J. Comput. Vision* **59** (2004) 233–258
13. Panin, G., Ladikos, A., Knoll, A.: An efficient and robust real-time contour tracking system. In: ICVS. (2006) 44
14. Moddemeijer, R.: On estimation of entropy and mutual information of continuous distributions. *Signal Processing* **16** (1989) 233–246
15. Viola, P., William M. Wells, I.: Alignment by maximization of mutual information. *Int. J. Comput. Vision* **24** (1997) 137–154
16. Nelder, J., Mead, R.: A simplex method for function minimization. *Computer Journal* **7** (1965) 308–313