

3D Facial Tracking from Corrupted Movie Sequences

Siome Goldenstein
IC-Unicamp, Brazil
siome@ic.unicamp.br

Christian Vogler
Gallaudet University, USA
Christian.Vogler@gallaudet.edu

Dimitris Metaxas
Rutgers University, USA
dnm@cs.rutgers.edu

Abstract

In this paper we perform 3D face tracking on corrupted video sequences. We use a deformable model, combined with a predictive filter, to recover both the rigid transformations and the values of the parameters that describe the evolution of the facial expressions over time.

To be robust, predictive filters need a good observation of the system's state. We describe a new method to measure, at each moment in time, the correct distribution of an observation of the parameters of a high-dimensional deformable model. This method is based on bounding the confidence regions of the 2D image displacements with affine forms, and propagating them into parameter space. Using Lindeberg's theorem, we measure a good Gaussian approximation of the parameters in a manner that avoids many of the traditional assumptions about the observations' distributions.

We demonstrate in experiments on sequences with compression artifacts, and poor-quality video sequences of Lauren Bacall and Humphrey Bogart from the 1950s, that, without any learning involved, our method is sufficiently robust to extract information from degraded image sequences. In addition, we provide ground truth validation.

keywords: “Kalman filter”, “Particle Filter”, “3D face tracking”, “deformable models”, “parametric estimation”.

1. Introduction

In this paper we address the problem of tracking three-dimensional deformable faces from poor-quality monocular video sequences, such as movies recorded in the 1950s. These movie sequences exhibit noise, flickering, varying illumination, and digital compression artifacts. We do not have camera calibration information, and we have no guarantee that the intrinsic parameters remain constant. All these characteristics make the tracking problem unusually difficult. To address these difficulties, we present a deformable model-based approach combined with a powerful statistical estimation technique and predictive filtering that does not require learning. This approach recovers the orientation and position of the faces at every frame, as well as the value of parameters that control the facial expressions.

Face tracking is a first step in a series of important ap-

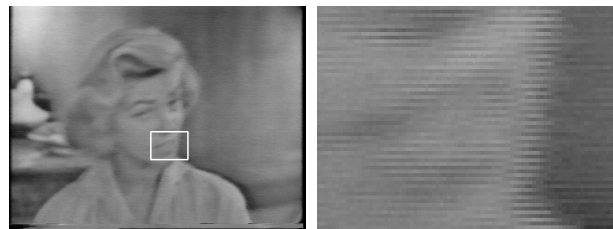


Figure 1. (a) TV interview of Lauren Bacall. (b) Zoom into the image.

plications such as surveillance, face recognition, human-computer interaction, and animation. In surveillance, a system could try to discern a subject's focus of interest based on her gaze direction. In face recognition from arbitrary video, it is necessary to know the shape and location of each face part before running the recognition. Human-computer interaction applications can infer things from the facial expressions and deformations. In computer graphics, it is desirable to use real facial expressions to animate models.

Robust algorithms in computer vision are hard to come by [17]. Tracking three-dimensional models from a noisy image stream, as we do in this paper, without a proper statistical representation of the noise, is a daunting task. This is particularly important for surveillance and military applications, where the transmission channel might irreversibly corrupt the data, and in tracking from uncontrolled TV and movie footage for computer graphics and special effects. In this paper, in particular, we use sequences from public domain footage of TV interviews recorded in the 1950s (Figure 1). In them the signal was interlaced, their cameras were not steady, and they have been through several media conversions, which all contributed to poor image quality.

To accomplish tracking on such sequences, we use predictive filters (Section 2) combined with a 3D deformable model tracking approach [18]. To use the predictive filters, we require an estimate of the probability distribution of the deformable model's state.

To obtain this estimate we apply various low-level computer vision algorithms to extract local two-dimensional

displacements – *2D image forces* – for selected points on the 3D model, as well as symmetric convex confidence regions that bound them (the support of their probability distributions). We represent these estimates of the image forces with *affine forms*, and use *affine arithmetic* [21] to project them into parameter space to form a multidimensional force: the *generalized force*, or the *cue*, of the algorithm. When the algorithm selects image points sufficiently far apart from one another, we can reasonably assume that the resulting image forces are independent. Under these circumstances, the conditions for the Lindeberg theorem are satisfied [12], and we can approximate the cue as a multivariate Gaussian distribution (Section 3).

We then show how to use the multivariate Gaussian approximation of the cue’s generalized force to estimate the distribution of a direct measurement of the system’s state as input to predictive recursive filters (Section 4), and validate our approach with tracking of impoverished video sequences (Section 5.1) and ground truth data (Section 5.2).

1.1. Related Work

Deformable models [18] and active contours [9, 2] can all be used in tracking applications. In 3D face tracking there are special applications, such as tracking with the optical flow constraint [10], using a volumetric model [22], 2D texture deformations over a cylinder [7, 6], a model-texture hybrid approach [25], using a linear set of morphable faces [20], and dynamic motion templates [24]. Creating the right model for these techniques is a hard task [3, 26, 20]. Additionally, model-free non-rigid motion can be recovered [5, 4].

There are several general statistical approaches designed for tracking, estimation, and prediction. The *Kalman filter* [16], for example, treats the parameters, as well as the observations, as multivariate Gaussians and also uses a linear predictive model. *Particle filter* [13] techniques, also known in computer vision as *condensation* [14], propagate the evolution of non-Gaussian sampled distributions through nonlinear operations. The *Unscented Kalman filter* [23] also works with particles, but has a limited number of carefully selected samples that grows linearly with the dimension of the state vector. All predictive filters require knowledge of the observations’ distributions, and different applications estimate it in different ways. For example, in [14] it is estimated from the gradient information of the neighborhood, while in [11] it was precomputed from the knowledge of the environment.

2. Predictive Filters: A Quick Overview

Predictive, or Bayesian, filters are a well established approach for tracking. They present a solid framework for

combining a model of the system’s evolution with an observation of the system’s state. The system’s evolution model can be designed according to an engineer’s knowledge of the problem at hand, or learned from data. Both the model of evolution, and the observation of the state of the system, are subject to corruptions and errors, modeled as zero-mean multivariate noises with known distributions.

In a discrete-time approach, using the notation of \mathbf{q}_k for the distribution of the random variable representing the state vector \mathbf{q} at time sample k , the system’s model provides us with a reasonable prediction $\tilde{\mathbf{q}}_{k+1}$ of the state vector distribution at time sample $k + 1$:

$$\tilde{\mathbf{q}}_{k+1} = f_k(\mathbf{q}_k, \mathbf{w}_k), \quad (1)$$

where $f_k : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}^n$, n is the dimension of the state vector \mathbf{q} , and l is the dimension of the multivariate noise \mathbf{w} .

At each time sample k we make an observation \mathbf{y}_k , or measurement, of the system. These observations are related to the state vector as:

$$\mathbf{y}_k = h_k(\mathbf{q}_k, \mathbf{v}_k), \quad (2)$$

where $h_k : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$, p is the dimension of the measurements \mathbf{y} , and r is the dimension of the multivariate noise \mathbf{v} . All the information available at time sample k is represented as $D_k = \{y_1, \dots, y_k\}$. We also need to know the prior, $p(\mathbf{q}_1|D_0) = p(\mathbf{q}_1)$, which is the probability distribution function of the state vector at the starting time.

Predictive filters recursively find $p(\mathbf{q}_k|D_k)$ from $p(\mathbf{q}_{k-1}|D_{k-1})$ and y_k . This computation is done in two steps. In the first step

$$p(\mathbf{q}_k|D_{k-1}) = \int p(\mathbf{q}_k|\mathbf{q}_{k-1}) p(\mathbf{q}_{k-1}|D_{k-1}) d\mathbf{q}_{k-1},$$

but using Equation 1 it becomes

$$p(\mathbf{q}_k|D_{k-1}) = \int f_k(\mathbf{q}_{k-1}, \mathbf{w}_{k-1}) p(\mathbf{q}_{k-1}|D_{k-1}) d\mathbf{q}_{k-1}. \quad (3)$$

The next step, also called correction, is a direct application of the Bayes rule:

$$p(\mathbf{q}_k|D_k) = \frac{p(\mathbf{y}_k|\mathbf{q}_k) p(\mathbf{q}_k|D_{k-1})}{p(\mathbf{y}_k|D_{k-1})},$$

and using Equation 2 we get

$$p(\mathbf{q}_k|D_k) = \frac{h_k(\mathbf{q}_k, \mathbf{v}_k) p(\mathbf{q}_k|D_{k-1})}{\int h_k(\mathbf{q}_k, \mathbf{v}_k) p(\mathbf{q}_k|D_{k-1}) d\mathbf{q}_k}. \quad (4)$$

Each predictive filter solves the Bayesian recursive Equations 3 and 4 in a different way, using different assumptions. The Kalman filter has a closed algebraic solution, but it assumes multivariate Gaussians distributions,

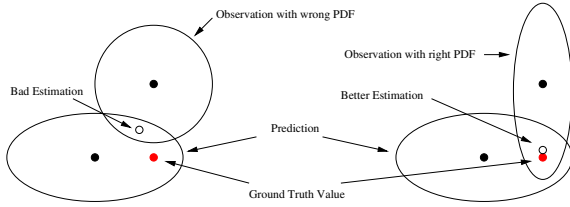


Figure 2. A 2D example of the effect of the observation's PDF in the estimation process. On the left, an arbitrary distribution, and on the right a measured distribution.

and that f_k and h_k are linear. Since Gaussian distributions are preserved under linear transformations, the Kalman filter only has to estimate the mean and covariance matrices.

A particle filter does not assume f_k or h_k to be linear, and neither does it use any particular parametric distribution. It applies f to samples that represent the prior, and according to the Bayes rule, uses h_k to resample from this new distribution. Although it is easy to understand and implement, precautions have to be taken to avoid particle collapse [15]. Additionally, the number of necessary samples grows fast with the dimension of the state vector. The Unscented filter tries to solve the latter problem with a set of carefully chosen samples, where the number of particles grows only linearly with the dimension of the state vector, and guarantees that at least the second moments are correct.

Predictive filters are ideal for tracking from noisy and corrupted data, as the measurements are smoothed out according to the system's evolution model. Unfortunately, they work as expected only if we know the real distribution of the measurement, $p(\mathbf{y}_k|\mathbf{q}_k)$ at every point in time. In Figure 2 we can see that an incorrect estimate of the observation's distribution will lead to an poor final estimate. One of the main attractions of our framework in this paper is that it provides us with the needed distributions.

3. Deformable Models for Tracking

In the deformable model framework, tracking the displacement of \mathbf{q} between two frames is achieved through the solution of a dynamical system:

$$\dot{\mathbf{q}} = \mathbf{K}\mathbf{q} + \mathbf{f}_g, \quad (5)$$

where \mathbf{K} is a *stiffness matrix*, and \mathbf{f}_g is a *generalized force*, describing the contribution of all cues. We numerically integrate Equation 5, starting at the value of \mathbf{q} at the end of the previous frame. The system converges to the closest equilibrium point of the internal and external forces [18]. We call its straightforward solution, where the contributions from the cues are averaged into a single \mathbf{f}_g , the *deterministic approach*.

However, averaging ignores that different cues can be structurally different. Sometimes they come from distinct images or cameras, and sometimes they affect disjoint sets of points. In the latter scenario, these different cues complement each other. For example, a point tracker cue works best in regions with complex texture, while a shape from shading cue works best in regions without texture. Cues can even come from three-dimensional data (like a range scanner). Therefore, a statistical approach is indicated. In our framework, each cue c models local 2D image contributions, $f_{c,j}$, as two-dimensional affine forms [21]. A two-dimensional affine form $\hat{f}_{c,j}$ is represented as:

$$\hat{f}_{c,j} = \mathbf{f}_{c,j,0} + \sum_{i=1}^m \mathbf{f}_{c,j,i} \varepsilon_i, \quad (6)$$

where the coefficients $\mathbf{f}_{c,j,i}$ are two dimensional vectors and ε_i are symbolic real variables whose values are unknown, but guaranteed to lie in the interval $[-1 \dots 1]$. The quantity $\mathbf{f}_{c,j,0}$ is the *central value*, and the ε_i are called the *noise variables*. Each noise variable ε_i represents an independent component of the total uncertainty. Here, $\hat{f}_{c,j}$ represents the region of uncertainty around the image force as a symmetric convex polygon, whose number of faces depends on m .

The cue's generalized force is the sum of its local contributions, after we project them into parameter space, using the Jacobian of the deformable model at each point:

$$\hat{f}_{g,c} = \sum_j \mathbf{B}_j^\top \hat{f}_{c,j}, \quad (7)$$

where \mathbf{B}_j^\top is the projected model Jacobian at point j , and $f_{c,j}$ is the image force that cue c applies at point j . Since $\hat{f}_{c,j}$ is an affine form bounding the image force, and \mathbf{B} is a 2-by- n matrix, $\mathbf{B}^\top \hat{f}_{c,j}$ is a set of affine operations over an affine form [21]. Equation 7 results in an n -dimensional affine form that represents the cue's generalized force.

If the 2D forces are selected at points sufficiently apart; that is, if we enforce a minimum distance between the points, it is reasonable to assume that they are independent in Equation 7 [12], and thus all noise variables in $\hat{f}_{g,c}$ are independent. Using *Lindeberg's theorem* [8, pp 205+], it is possible to prove that the sum will converge to a Gaussian distribution $\hat{f}_{g,c}$ with mean μ_c and covariance matrix Λ_c [12], and estimate its mean and covariance matrix as well through direct expectation properties. Since we ensure that $E[\varepsilon_i] = 0$, the mean vector is $\mu_c = E[\hat{f}_{g,c}] = \mathbf{f}_{g,c,0}$, and the covariance matrix is $\Lambda_c = E[(\hat{f}_{g,c} - \mathbf{f}_{g,c,0})(\hat{f}_{g,c} - \mathbf{f}_{g,c,0})^\top]$, which after manipulations has elements

$$\lambda_{ij} = \sum_{k=1}^m \mathbf{f}_{g,c,ki} \mathbf{f}_{g,c,kj} E[\varepsilon_k^2].$$

We then use *maximum likelihood estimation* (MLE) [1, 16] to find the optimal \mathbf{f}_g .

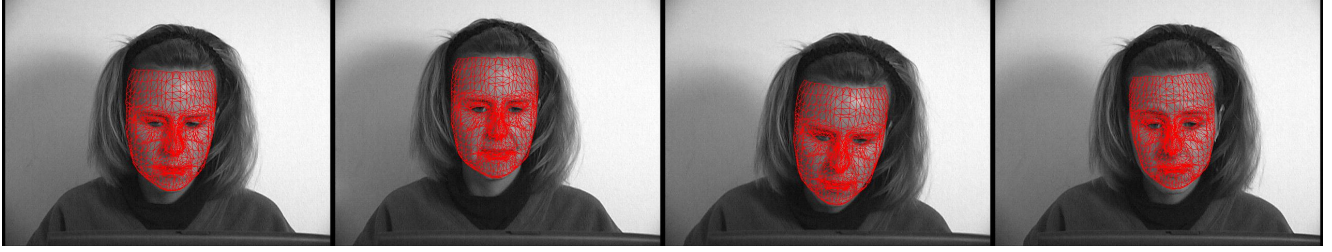


Figure 3. Snapshots of a 3600-frame sequence with compression artifacts (DV encoder).

4. Simple Kalman Filter in a Deformable Model Framework

Now we have the necessary pieces to formulate the predictive filter framework. The method described in the previous section finds the mean and covariance matrix of \tilde{f}_g . Using an adaptive Euler integration approach, we find the solution of Equation 5 to obtain $\tilde{\mathbf{q}}_{t,obs}$, the observations of the parameter vector at time t , and v_t , the additive Gaussian noise with zero mean:

$$\begin{aligned} \mathbf{q}_{t,obs}^i &= \mathbf{q}_{t,obs}^{i-1} + \epsilon_i \dot{\mathbf{q}}^{i-1} \quad \text{and} \\ Cov(v_t^i) &= Cov(v_t^{i-1}) + \epsilon_i^2 Cov(\dot{\mathbf{q}}^{i-1}), \end{aligned} \quad (8)$$

where i is the integration step counter, ϵ_i is the integration step at step i , and with initial conditions $\mathbf{q}_{t,obs}^0 = \mathbf{q}_{t-1}$ and $Cov(v_t^0) = Cov(\mathbf{q}_{t-1})$.

In this application, our observations are already obtained in parameter space, so we use a simple linear predictive model for the evolution of our system. The observations are derived from the cue's distributions, and have a Gaussian noise, so we can apply a Kalman filter [16, 1]:

$$\begin{cases} \mathbf{q}_t = A_t \mathbf{q}_{t-1} + w_t \\ \mathbf{z}_t = \mathbf{q}_{t,obs} + v_t \end{cases}, \quad (9)$$

where matrix A_t is the linear model of evolution for the parameter vector \mathbf{q} , and w_t is a Gaussian noise with zero mean. In our experiments we calculate A_t at each frame in order to predict the same transformation on \mathbf{q}_{t-1} as the system had effectively done between \mathbf{q}_{t-2} and \mathbf{q}_{t-1} , based on standard Kalman filter theory [1].

Our method provides a computationally efficient method to measure the distribution of the observation \mathbf{z}_t at each point in time in systems with a high dimension.

5. Real 3D Face Tracking Experiments

We now show experiments that demonstrate the robustness of our cue integration and predictive filtering framework. In addition, we show an experiment with ground truth data that evaluates the accuracy of the deformable tracking

framework with deterministic and statistical cue integration, and their combinations with predictive filtering using our technique for estimating the covariance matrices.

We implemented the techniques in this paper in our deformable model tracking system. We adapted and simplified a geometric model of a head, made available by the computer graphics group of the University of Washington as part of [19]. The result was a static mask model of a generic face with 1101 nodes and 2000 faces. We then marked regions on the surface that are affected by a parameter, and how this effect varied in this region. The model has 11 parameters, one to raise/lower eyebrows, two for the smiling/stretching of the mouth, one for the opening of the jaw, and another 7 for the reference frame. For simplicity, we modeled the jaw movement as an affine transformation, which is sufficient for small openings of the mouth, even though a rotation operation is the correct way to model the jaw.

Before tracking, the model needs to be fitted to the first frame of the sequence in a one-time procedure. In these experiments, we fitted by generating forces from manually specified correspondences between the model and image points, and then integrating Equation 5. Note that the system tracks the the mouth movement and does not lose track, even if the shape slightly drifts for short amounts of time (see also the movie).

5.1. Robustness

The video data were corrupted in two ways:

DV Compression Artifacts Figure 3 has snapshots of a tracking sequence with this model, fitted to a subject in a psychiatric experiment, which was designed to evaluate the relationship between stress and facial expressions. It was captured in grayscale at 30 Hz with a mini-DV camera and contained noticeable compression artifacts. This sequence had approximately 2000 frames (2 min). We integrated three cues, which were a point tracker, an edge tracker, and an optical flow estimator. As the images and the accompanying video show, the system kept good track throughout.

Old Movie Sequences Compression artifacts already pose problems to tracking systems, but they pale next to the problems posed by old videos, which suffer from interlacing, varying illuminations, flicker, jittery camera control,



Figure 4. Humphrey Bogart on TV.



Figure 5. Lauren Bacall on TV.

and noise due to degradation of the film masters (see also Figure 1). Especially damning, the illumination constraints that are needed to compute optical flow are not valid, so we are left only with point and edge tracking. We tested our system on two sequences featuring Humphrey Bogart and Lauren Bacall that were recorded in the 1950s. Each sequence was approximately 600 frames long. As the accompanying video and Figures 4, 5 show, the image quality was extremely poor, yet the system managed to keep track of the rigid model parameters and the facial deformations. These results demonstrate that a statistical approach to deformable models is powerful enough to handle extremely bad-quality data without learning, and without using optical flow.

5.2. Validation

To validate and compare the statistical approach and the embedding into a Kalman filter framework, we used a special concocted sequence, where the subject had 8 markers physically drawn in the face with makeup. We extracted the 2D coordinates of each marker and used our system to track the sequence — avoiding image information from the neighborhood of the markers. The error of each marker is defined as its 2D distance from our estimate to its actual position. In Figure 6 we show a snapshot of the sequence and the plot comparing the average error (in pixels) of the markers between a simple non-statistical method (average of the cues), a Kalman filter using our method to estimate the distribution of each cue (but without the MLE cue integration), the tracking using the statistical cue integration, and a tracking experiment using both cue integration and a Kalman filter. The sequence had 900 frames at 15Hz (1

minute) at a 320×240 resolution. Our system used an average of 0.7s/frame (including 0.2s for image processing), with at most 600 steps in the integration of Equation 5.

In the error plot, we observe that a Kalman filter only achieves significant gains in accuracy when the distribution of the observations is correctly estimated with our cue integration model. The purely deterministic approach, with and without a Kalman filter, eventually loses track, with drifting errors of 35–45 pixels. In contrast, the experiments using the MLE cue integration, with and without a Kalman filter, keep track, with drifting errors in the range of 6–12 pixels.

6. Conclusions

Our approach to estimating the probability distributions of the deformable model's system state, combined with a predictive filter, significantly improves the robustness and accuracy of the tracking process, as demonstrated on the ground truth data, and makes tracking of impoverished video sequences feasible. The difference in accuracy between the statistical and deterministic methods grows as the quality of the data degrades. Finally, there is no learning involved in our method, only a one-time fitting process of the face model to an image frame at the beginning of the tracking sequence. This illustrates how reliably we extract information from the degraded image sequence.

The Gaussian approximation holds well only if there are few outliers in the image forces. We avoid most of them by allowing each cue to select only the points of the model that have good image attributes, but future work should apply explicit outlier rejection methods. For instance, ideally, optical flow would provide use with information of points

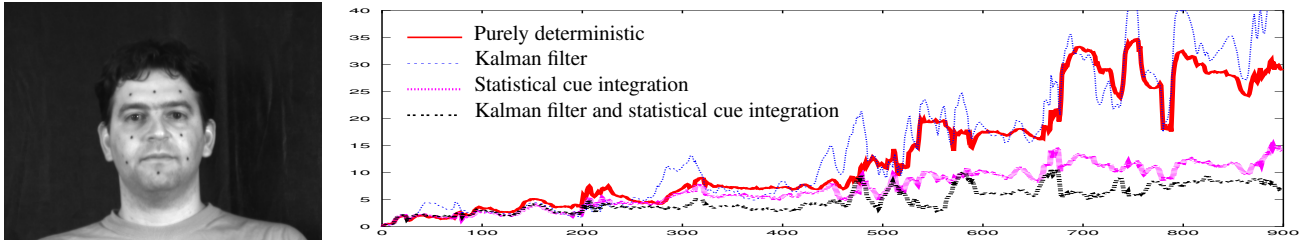


Figure 6. Average pixel error of ground truth tracking sequence using different methods. Best to worst: Combination of MLE cue integration/Kalman filter, MLE, pure deterministic, Kalman filter.

that are not good for either the point tracker or the edge tracker. Unfortunately, in our experiments the optical flow turned out to be a liability, because the illumination constraints were being violated so blatantly. To make it work, we are currently investigating low-level image processing restoration techniques, relaxing the flow constraints, as well as robust outlier rejection procedures.

Acknowledgments

The research in this paper was supported by NASA Cooperative Agreements 9-58 with the National Space Biomedical Research Institute, CNPq, CAPES, FAPESP, FAEP-Unicamp, and research scientist funds by the Galaudet Research Institute.

References

- [1] G. Bishop and G. Welch. An introduction to the kalman filter. SIGGRAPH 2001 Course Notes, 2001.
- [2] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag, 1998.
- [3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, August 1999.
- [4] M. Brand and R. Bhotika. Flexible flow for 3d nonrigid tracking and shape recovery. In *CVPR*, pages 315–322, 2001.
- [5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *CVPR*, 2000.
- [6] L. Brown. 3d head tracking using motion adaptive texture-mapping. In *CVPR*, pages 998–1005, 2001.
- [7] M. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Trans. PAMI*, 22(4):322–336, 2000.
- [8] K. Chung. *A Course in Probability Theory*. Academic Press, Inc., 2nd edition, 1974.
- [9] T. Cootes and C. Taylor. Active shape models - their training and application. *CVIU*, 61(1):38–59, 1995.
- [10] D. de Carlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99–127, July 2000.
- [11] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *CVPR*, 1999.
- [12] S. Goldenstein, C. Vogler, and D. Metaxas. Statistical Cue Integration in DAG Deformable Models. *IEEE Trans. PAMI*, 25(7):801–813, 2003.
- [13] N. Gordon, D. Salmon, and A. Smith. A novel approach to nonlinear/nongaussian bayesian state estimation. *IEEE Proc. Radar Signal Processing*, (140):107–113, 1993.
- [14] M. Isard and A. Blake. CONDENSATION: conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.
- [15] O. King and D. Forsyth. How does CONDENSATION behave with a finite number of samples? In *ECCV*, pages 695–709, 2000.
- [16] P. Maybeck. *Stochastic Models, Estimation, and Control*. Academic Press, 1979.
- [17] P. Meer, C. V. Stewart, and D. E. Tyler. Robust computer vision: An interdisciplinary challenge. In *CVIU*, number 78, pages 1–7, 2000.
- [18] D. Metaxas. *Physics-based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*. Kluwer Academic Publishers, 1996.
- [19] F. Pighin, R. Szeliski, and D. Salesin. Resynthesizing facial animation through 3d model-based tracking. In *ICCV*, pages 143–150, 1999.
- [20] F. Pighin, R. Szeliski, and D. Salesin. Modeling and animating realistic faces from images. *IJCV*, 50(2):143–169, 2002.
- [21] J. Stolfi and L. Figueiredo. *Self-Validated Numerical Methods and Applications*. 21^o Colóquio Brasileiro de Matemática, IMPA, 1997.
- [22] H. Tao and T. Huang. Visual Estimation and Compression of Facial Motion Parameters: Elements of a 3D Model-Based Video Coding System. *IJCV*, 50(2):111–125, 2002.
- [23] E. A. Wan and R. van der Merwe. *Kalman Filtering and Neural Networks*, chapter Chapter 7 : The Unscented Kalman Filter, (50 pages). Wiley Publishing, 2001.
- [24] L. Wang, W. Hu, and T. Tan. Face tracking using motion-guided dynamic template matching. In *ACCV*, 2002.
- [25] Z. Wen and T. Huang. Capturing subtle facial motions in 3d face tracking. In *ICCV*, pages 1343–1350, 2003.
- [26] Z. Z. Ying Shan, Zicheng Liu. Model-based bundle adjustment with application to face modeling. In *ICCV*, pages 644–651, 2001.