Outlier Rejection in Deformable Model Tracking

Siome Goldenstein[1] Christian Vogler[2] Jorge Stolfi[1] Vladimir Pavlovic[3] Dimitris Metaxas[3]

[1] {siome,stolfi}@ic.unicamp.br
Instituto de Computação - Unicamp
Caixa Postal 6176, Campinas, SP, Brazil 13084-971

[2] Christian.Vogler@gallaudet.edu
Gallaudet Research Institute - Gallaudet University
800 Florida Ave. NE, Washington DC, USA 20002-3695

[3] {vladimir,dnm}@cs.rutgers.edu
Department of Computer Science - Rutgers University
110 Frelinghuysen Road, Piscataway, NJ, USA 08854-8019

Abstract

Deformable model tracking is a powerful methodology that allows us to track the evolution of high-dimensional parameter vectors from uncalibrated monocular video sequences. The core of the approach consists of using low-level vision algorithms, such as edge trackers or optical flow, to collect a large number of 2D displacements, or motion measurements, at selected model points and mapping them into 3D space with the model Jacobians. However, the low-level algorithms are prone to errors and outliers, which can skew the entire tracking procedure if left unchecked.
There are several known techniques in the literature, such as RANSAC, that can find and reject outliers. Unfortunately, these approaches are not easily mapped into the deformable model tracking framework, where there is no closed-form algebraic mapping from samples to the underlying parameter space. In this paper we present two simple, yet effective ways to find the outliers. We validate and compare these approaches in an 11-parameter deformable face tracking application against ground truth data.
keywords: *"Outlier Rejection", "Robust Methods", "Deformable Models", "3D Face Tracking".*

1. Introduction

Tracking deformable models is a hard task. We need to estimate the underlying parameters of our model, both rigid and nonrigid, from only two-dimensional video sequences. Face tracking, for example, is the first step in a series of important applications such as surveillance, face recognition, human-computer interaction, and animation.

It is very hard to find correspondences between image pixels and model points. Typically, we use computer vision algorithms (such as edge trackers and optical flow), together with the inductive assumption of tracking, to estimate pixel correspondences between two consecutive images. We can then use an optimization procedure to find the new value of the model parameters that minimize the two-dimensional displacements between the model points and the corresponding image pixels.

Computer vision algorithms are subject to errors. When these errors are all characterized by a well-behaved distribution, then the use of a large number of image displacements works as an averaging, or smoothing, procedure, and these errors cancel one another out. Unfortunately, sometimes there are gross outliers - elements that should not be there and are not a good description of the underlying dynamics. Outliers can occur because of invalid assumptions, numerical errors, or just because some heuristics are not guaranteed always to work. It has been known that even a small number of outliers can "poison" the result of an algorithm, and deformable model tracking is no exception.

Robust algorithms in computer vision are hard to come by [14]. Many applications are tailored to one particular class of of data, and do not necessarily generalize well to different types of inputs. 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. Since we do not have the noise model, the least that we can do is to eliminate the obvious outliers. In this paper we describe and compare two techniques to detect such outliers. The first method works in the image space, but requires some numerical approximations. For this reason, we develop a novel method that detects outliers in parameter space, without the need for any approximations.

The rest of the paper is organized as follows: We discuss related work, provide an overview of deformable model tracking, and describe the need for a robust outlier rejection procedure. We then discuss the two approaches for finding the outliers, and compare their performance in validation experiments with ground truth data.

1.1. Related work

Deformable models and their representations are an active area of research in computer vision. Stereo and shape from shading methods obtain initial fits [21]. Within a similar framework, [9] uses anthropometric data and inspired deformations to generate faces. A learning-based statistical model can help tracking of face models [3]. Eigen-based approaches, such as PCA decompositions, can successfully track, fit, and even recognize objects [15, 2, 17, 19]. In [5] the head is modeled as a cylinder, and in [1] as a plane, and in [17, 18] tracking is used for animation. In [4], tracking uses adaptive texture. A powerful deformable volumetric model has been also used for fast face tracking [23] and subtle motion [27] capture. Integration of distinct cues have been used to reduce the effect of outliers particular to a specific algorithm [8, 11]. In [11], the deformable model cues have their distributions measured, but it ignores the effects of possible outliers, and in [12], this distribution is used to measure the observation of a predictive filter.

Outlier rejection is an important step in any field that deals with noisy and corrupted data. Several methods use sampling of multiple minimum-size sets to estimate common underlying parameters, and find out which elements should be discarded. Among them there are RANSAC [10], MLESAC [26], and IMPSAC [24]. M-estimators look for optimum weighting of each element, instead of just trying to discard outliers [13, 6, 7]. There are good overviews and comparisons of these methods, both in the general statistics literature [20], as well as applied to computer vision particular problems [25].

2. Deformable Models for Tracking

Using a deformable model is appropriate whenever we would like to track a non-rigid object, about which we have a lot of prior knowledge (i.e., general shape and range of deformations). A deformable model has its geometric shape

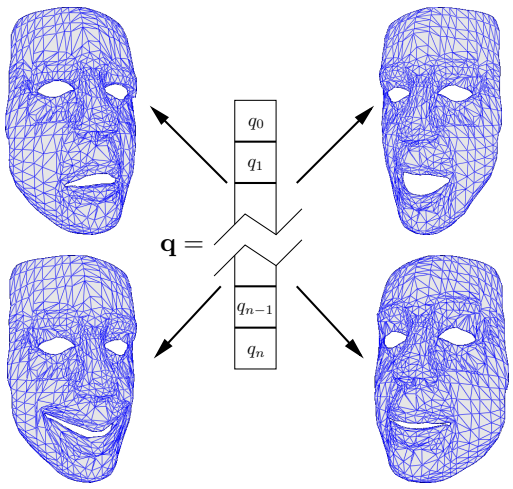

Figure 1. Parameter Vector controls shape and rigid transformation.

fully determined by the value of a parameter vector $\mathbf{q}$. Some parts of $\mathbf{q}$ are responsible for the rigid motion, whereas others change its shape. For example, in a deformable model for face tracking, as in Figure 1, there are parameters for eyebrow movement, and mouth opening and stretching. For every 2D image point p_i on the surface of our model, we have a function

$$p_i = F_i(\mathbf{q}) \tag{1}$$

that evaluates its position for every value of the parameter vector $\mathbf{q}$.

During a tracking session, for every new frame k, we look for the value of the parameter vector $\mathbf{q}_k$ that achieves the best model-image correspondence. In the general case, this task is very complicated. Even when we can find the correspondence between image pixels and model points, the functions F_i are usually nonlinear, and in the general case there is no closed-form solution for this inverse problem.

Nevertheless, we can use computer vision algorithms, such as edge trackers and optical flow, to find image to image correspondences between frames. We then use these two dimensional displacements $\mathbf{f}_i$, which we also call *image forces*, to adjust the value of $\mathbf{q}_{k-1}$ to $\mathbf{q}_k$ iteratively. This adjustment is nothing more than a local optimization in parameter space: the search for a new $\mathbf{q}$ that minimizes the sum of the magnitude of all image forces $\mathbf{f}_i$.

The first step is to map all image forces into a single contribution in parameter space, a *generalized force*

$$\mathbf{f}_g = \sum_i B_i^\top \mathbf{f}_i, \tag{2}$$

where B_i is the projected Jacobian of the model at point p_i

to which the image force $\mathbf{f}_i$ is applied. Using the generalized force $\mathbf{f}_g$ we solve the dynamical system

$$\dot{\mathbf{q}} = K\mathbf{q} + \mathbf{f}_g, \tag{3}$$

where K is a *stiffness matrix*, using a simple gradient descent method.

Obviously, the quality of the solution depends on the estimate of $\mathbf{f}_g$, which in turn depends on the quality of the image force estimates $\mathbf{f}_i$. Computer vision algorithms are known for their inconsistency, so the quality of the estimates of $\mathbf{f}_i$ tends to be uneven. The common way to deal with this problem is to calculate a very large number of image forces, and to hope that there is only a small number of outliers which will be washed out through the averaging process of Equation 2. Unfortunately, in practice, at best, ignoring outliers results in noisy tracking. At worst, in the presence of noisy data, or corrupted video sequences, there typically is a significantly larger percentage of outliers than normal, which can cause the system to lose track.

3. Outlier Rejection

The collection of image forces gathered by a computer vision algorithm is normally corrupted by noise. In addition, another major source of errors comes from data point outliers, due to failures of the underlying vision algorithms, such as violations of assumptions, occlusions, and so on. If we do not take such outliers into account, they can dramatically throw off our estimates of the generalized forces.

Any attempt to detect the outliers based solely on the 2D characteristics of the image is bound to fail, because it completely discards any information that we have from the model. For instance, in our face model shown in Figure 1, different points of the model are controlled by different parameter subspaces, which in turn affects how we expect these points to behave over time, with respect to trajectory, velocity, and so on. This information is expressed directly in the Jacobians of the points.

In general terms, robust outlier rejection is based on the idea that a point is likely to be an outlier if in some frame it dramatically differs from the expected behavior. In the following we describe two different approaches to outlier rejection that combine the information from the 2D image and the model Jacobians.

3.1. Image Space Outlier Rejection

Many tracking applications use a Kalman filter, to combine a prediction of the system's state with a measurement, or observation. The prediction model can be based on an engineer's empirical knowledge of the problem at hand, or can even be learned from data.

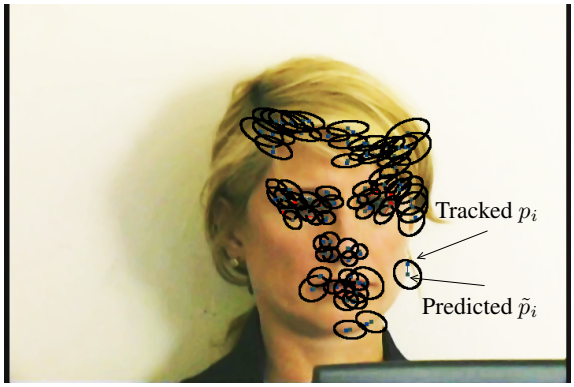

Figure 2. Image space-based outlier rejection

The simplest method to perform outlier rejection of image forces in a deformable model framework is to use the Kalman filter's prediction of the state [12], before the fusion with the observation, to weed out the outliers. At frame k, we have the multivariate Gaussian prediction $\tilde{\mathbf{q}}_k$ of the parameter vector for frame $k+1$, the covariance matrix $\Lambda_{\tilde{\mathbf{q}}_k}$ giving the uncertainty in the prediction, and the associated predicted points in image space

$$\tilde{p}_i = F_i\left(\tilde{\mathbf{q}}_k\right). \tag{4}$$

We cannot calculate the covariance matrix $\Lambda_{\tilde{p}_i}$ of $\tilde{p}_i$ exactly, because the functions F_i are nonlinear. We can, however, calculate an approximation with a linearization of the F_i:

$$\Lambda_{\tilde{p}_i} \approx \tilde{B}_i^\top \Lambda_{\tilde{\mathbf{q}}_k} \tilde{B}_i, \tag{5}$$

where $\tilde{B}_i$ is the projected Jacobian of the model at the predicted image point $\tilde{p}_i$.

This covariance matrix defines an ellipsoid. If the prediction model is good, the computer vision algorithms will track and place the actual point p_i at frame $k+1$ somewhere in the vicinity of the predicted $\tilde{p}_i$; that is, it will be most likely contained within this ellipsoid (Figure 2). If p_i falls outside it, it is considered to be an outlier. Thus, we can use the *Mahalanobis metric* in conjunction with a threshold to determine which image forces should be considered and which ones should be rejected:

$$\mathbf{x}^\top \Lambda_{\tilde{p}_i}^{-1} \mathbf{x} \leq threshold, \tag{6}$$

where $\mathbf{x} = p_i - \tilde{p}_i$ is the difference between the respective tracked and predicted image points.

This method requires a good prediction model of the system's evolution, and assumes that a Gaussian distribution can properly represent the parameter vector's distribution. The most serious limitation of this approach is that it does

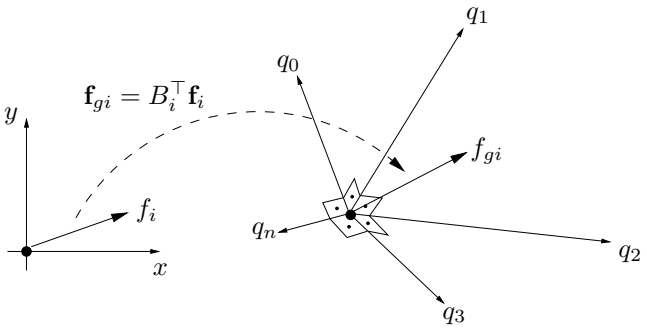

Figure 3. Projection of two-dimensional force into parameter space.

not take into account the nonlinearities of F_i when projecting the Gaussian distribution into image space. This limitation suggests that a better approach is to perform the outlier rejection in the parameter space, thus avoiding the projection and associated nonlinearity issues entirely.

3.2. Parameter Space Outlier Rejection

We would like to avoid a direct linearization of F_i, which is necessary to project a Gaussian into image space, as in the approach of the previous section. From Equation 2, $\mathbf{f}_g$ is defined as the sum of the image forces' projections into parameter space

$$\mathbf{f}_g = \sum_i \mathbf{f}_{gi}, \tag{7}$$

where

$$\mathbf{f}_{gi} = B_i^\top \mathbf{f}_i,$$

illustrated in Figure 3.

If we were follow the same idea as in the previous section, we would like to predict the value of the generalized force $\mathbf{f}_g$ for frame $k+1$, and use the Mahalanobis metric on this higher dimension Gaussian distribution of the parameter-space projection of the 2D forces (Equation 7). Unfortunately, this approach turns out to be very unreliable, because $\mathbf{f}_g$ is directly related to the first derivative with respect to the model parameters, and thus the prediction is inherently too noisy to be of use.

The approach that we follow, instead of predicting $\mathbf{f}_g$, is based on estimating the distribution of $\mathbf{f}_g$ in the current frame and rejecting any force $\mathbf{f}_{gi}$ that is not compatible with this distribution. In previous work we showed that, as $\mathbf{f_g}$ is the sum of the parameter-space projection of many 2D forces (Equation 2), a Gaussian reasonably represents the distribution of the resulting generalized force $\mathbf{f}_g$ [11], so once we have estimated $\mathbf{f}_g$ we can, in principle, again use the Mahalanobis metric to test the individual $\mathbf{f}_{gi}$. We now show how to measure the parameters of this Gaussian (mean and covariance matrix) from the individual forces and their associated Jacobians, and then show how to use a modified Mahalanobis metric to identify forces that are not compatible with the distribution.

Estimating the Distribution of $\mathbf{f}_g$

If we treat the generalized forces as cloud of points in the parameter space, we can group them in a matrix

$$F_g = \begin{bmatrix} | & | & & | \\ f_{g1} & f_{g2} & \cdots & f_{gN} \\ | & | & & | \end{bmatrix}, \tag{8}$$

and calculate the mean μ and the covariance matrix Λ through

$$\mu = \frac{1}{N} F_g \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \text{and} \quad \Lambda = \frac{1}{N} F_g F_g^\top. \tag{9}$$

However, these simple estimates will be incorrect, because of the phenomenon of *parameter unobservability*. When the original 2D force f_i is non-zero, a zero entry in the j^{th} row of a generalized force $\mathbf{f}_{gi}$ can have two different meanings:

1. The j^{th} parameter is already at a minimum, and thus $\frac{\partial p_i}{\partial j} \cdot \mathbf{f}_i = 0$.
2. The point p_i does not depend on the j^{th} parameter, and so $\frac{\partial p_i}{\partial j} = 0$.

In the first case, the j^{th} entry of the generalized force should be zero, but in the second case the parameter is *unobservable:* we cannot draw any conclusions about its value, because it does not affect the point p_i to which the 2D force $\mathbf{f}_i$ is being applied. With this interpretation in mind, it is clear why it is incorrect to interpret F_g as a simple cloud of data points in parameter space — these unobservable points drag the mean and covariance values down.

To find the mean μ, we need to estimate each component μ_j using only the subset S_j of generalized forces for which the parameter j is observable:

$$S_j = \left\{ p_i \,\middle|\, \frac{\partial p_i}{\partial j} \neq 0 \right\}. \tag{10}$$

Then

$$\mu_j = \frac{1}{N_j} \sum_i^N f_{gi,j}, \tag{11}$$

where $f_{gi,j}$ is the j^{th} component of $\mathbf{f}_{gi}$, and $N_j = |S_j|$.

The calculation of the covariance matrix Λ is more complicated. If we start with $\Lambda = F_g F_g^\top$, and divide each element by a different number of valid terms (analogous to

Equation 11), we may obtain a non-positive-definite matrix. The off-diagonal elements of the covariance matrix cannot be calculated from a subset of points that is different from the subset used to calculate the pair of associated diagonal elements; otherwise the positive-definiteness property does no longer hold. To solve this problem, we assume that if parameters j and k are observable from the different sets of points S_j and S_k, respectively, they should be treated as independent. Thus, the cross-terms between these parameters are zero:

$$\Lambda_{jk} = \begin{cases} \frac{1}{N_j} \sum_i^N \left(f_{gi,j} - \mu_j\right)\left(f_{gi,k} - \mu_k\right) & \text{if } S_j = S_k; \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The covariance matrix calculated this way can always be viewed as a block diagonal matrix (after rearranging the parameters):

$$\Lambda = \begin{bmatrix} \square & & \\ & \square & \\ & & \square \end{bmatrix},$$

where every sub-block is positive definite. Thus, Λ is positive definite.

Observable Subspace Mahalanobis Metric

The rejection criterion for every generalized force is based on the Mahalanobis distance, restricted to the subset of observable parameters. The reason for this restriction is that when a point p_i has an unobservable parameter j, its generalized force has a j^{th} component $\mathbf{f}_{gi,j} = 0$, and the distance between zero and μ_j might be large enough to pull the Mahalanobis distance above the threshold.

We accomplish this restriction by projecting the forces and covariance matrix into the observable subspaces for each point. If a point p_i has l observable parameters, we can build a projection matrix P_i, with dimensions $l \times n$, composed of only 0s and 1s, that projects a force from the n-dimensional parameter space into the l-dimensional observable subspace. This same matrix is also used to project the covariance matrix into the observable subspace, so the acceptance criterion for a force is its observable subspace Mahalanobis distance

$$\left(\mathbf{f}_{gi} - \mu\right)^\top P_i^\top \, P_i \Lambda P_i^\top \, P_i \left(\mathbf{f}_{gi} - \mu\right) \leq threshold. \quad (13)$$

4. Validation with 3D Face Tracking

Recall that the image space-based rejection method involves a linearization, whereas the parameter space-based rejection method does not. All other factors equal, we therefore would expect the parameter space-based rejection method to do better. To test this hypothesis, we implemented the two methods in our existing 3D face tracking system.

Figure 4. Subject with eight markers physically drawn in the face.

For a quantitative evaluation of the methods, we produced a video sequence at 60 Hz where we physically drew markers on the subject's face (Figure 4). We extracted the 2D image position of each marker for all frames in a semi-automated manner based on thresholding. The 3D model allowed us to compare the actual 2D marker position with the projected position of marker in the 3D model at each frame of the tracking experiment. Our evaluation criterion is the distance between the projected and actual 2D marker positions, where lower numbers obviously mean better tracking.

We used an all-purpose deformable face model with 1101 nodes, 2000 faces, and 11 parameters that controlled both the rigid transformation as well as the facial deformations (eyebrows, lip stretching, smiling, jaw opening, etc.) of each of these points (see Figure 1). Before we can track a new subject, the model's mesh first needs to be fitted to the face in an image at rest position, without the effect of deformations. Note that this process needs to be only done once for every subject, through methods such as [2, 9, 16].

For the purposes of these validation experiments, we fitted the model in a semi-automatic way: the user manually selected a few dozen model-image correspondences. Fitting then consisted of solving Equation 3, with the user correspondences as the image forces $\{\mathbf{f}_i\}$ (Section 2, Equation 2), using a finite-element inspired set of shape deformations. Because the definitions of the facial expression deformations are independent of the base mesh [11], the model was ready for tracking immediately after fitting.

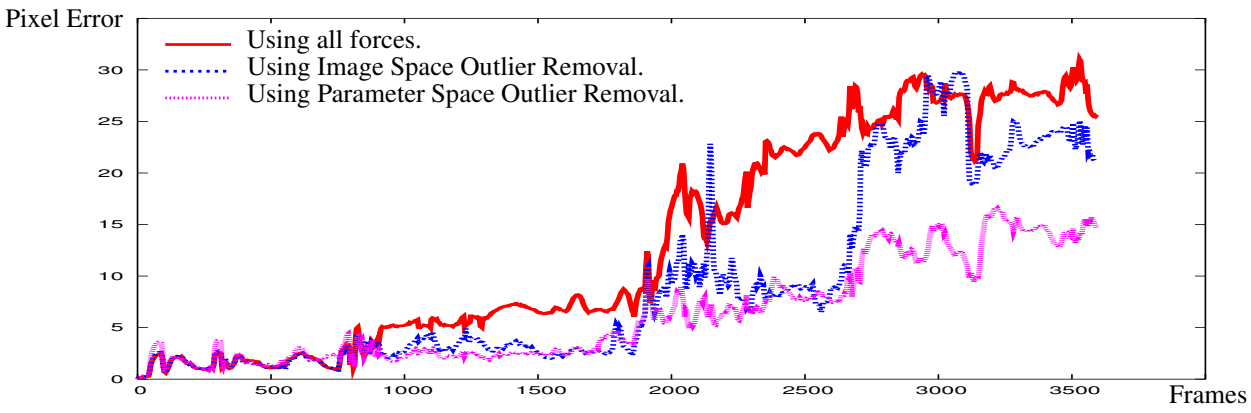

Figure 5. Mean errors of visible markers.

To isolate and understand the effect of the outlier rejection techniques, we simplified the tracking procedure as much as possible, leaving only one point-tracking algorithm (cue), and removing tethering procedures. This algorithm selected the model points that were most suitable for tracking [22], and then tracked them to obtain their displacements f_i at each new frame.

We compared the effects of no rejection, image space-based rejection (Section 3.1), and parameter space-based rejection (Section 3.2). Figure 5 shows the results of these three different techniques. No outlier rejection (red, solid curve) fares the worst, with the error quickly growing out of control. Image space-based rejection (dotted blue curve) and parameter space-based rejection (dashed purple curve) fare much better, with parameter-space based rejection being more robust over the long run.

5. Conclusions

In this paper we have introduced two approaches for rejection of the outlier forces generated from low-level computer vision algorithms. The first approach uses a multi-dimensional Gaussian prediction, and the second one builds a covariance matrix from the available data. In both techniques, the final outlier detection criterion is a Mahalanobis distance.

The first approach predicts the actual value of the parameters as a Gaussian distribution, projects this Gaussian into image space through a first order linearization of the deformable function (the Jacobian), and tests the 2D forces against these 2D Gaussians. When the tracking system already has this prediction (through the use of a Kalman filter, for example), this approach is an easy and computationally cheap way to detect, and reject, outliers. It has a disadvantage of incorrectly classifying some points, because of the extra uncertainty of the prediction stage, and the first order approximation necessary to propagate a Gaussian distribution through a non-linear function. The performance of this procedure is closely related to the quality of the prediction.

The second approach builds a mean and covariance matrix based on the available forces. We have to take special care with the parameter observability issue, but there is no predictive step involved, thus increasing robustness. On the other hand, compared to the first method, this approach requires an extra step to calculate the covariance matrix. However, since the outlier rejection is performed only once per frame, this extra computational cost is insignificant compared to the remaining operations necessary to track a frame. In addition, since this approach works in a subspace of the parameter space, it requires no linear approximations. Overall, these factors and the validation results support a clear preference for the second approach.

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 Gallaudet Research Institute.

References

[1] M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, August 1999.

[3] M. Brand and R. Bhotika. Flexible flow for 3D nonrigid tracking and shape recovery. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 315–322, 2001.

[4] L. Brown. 3D head tracking using motion adaptive texture-mapping. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 998–1005, 2001.

[5] 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 Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000.

[6] H. Chen and P. Meer. Robust computer vision through kernel density estimation. In *Proceedings of European Conference of Computer Vision*, pages 236–250, 2002.

[7] H. Chen and P. Meer. Robust regression with projection based m-estimators. In *Proceedings of International Conference of Computer Vision*, pages 878–885, 2003.

[8] D. de Carlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2000.

[9] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *Proceedings of the SIGGRAPH*, pages 67–74, 1998.

[10] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[11] S. Goldenstein, C. Vogler, and D. Metaxas. Statistical Cue Integration in DAG Deformable Models. 2003.

[12] S. Goldenstein, C. Vogler, and D. Metaxas. 3D facial tracking from corrupted movie sequences. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2004.

[13] R. Marona. Robust m-estimators of multivariate location and scatter. *Ann. Stat.*, 4:51–67, 1976.

[14] P. Meer, C. V. Stewart, and D. E. Tyler. Robust computer vision: An interdisciplinary challenge. In *Computer Vision and Image Understanding*, number 78, pages 1–7, 2000.

[15] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[16] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of the SIGGRAPH*, pages 75–84, 1998.

[17] F. Pighin, R. Szeliski, and D. Salesin. Resynthesizing facial animation through 3D model-based tracking. In *Proceedings of International Conference of Computer Vision*, pages 143–150, 1999.

[18] F. Pighin, R. Szeliski, and D. Salesin. Modeling and animating realistic faces from images. *International Journal of Computer Vision*, 50(2):143–169, 2002.

[19] A. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *Proceedings of International Conference of Computer Vision*, pages 59–66, 2003.

[20] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[21] D. Samaras, D. Metaxas, P. Fua, and Y. Leclerc. Variable albedo surface reconstruction from stereo and shape from shading. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 480–487, 2000.

[22] J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[23] H. Tao and T. Huang. Visual Estimation and Compression of Facial Motion Parameters: Elements of a 3D Model-Based Video Coding System. *International Journal of Computer Vision*, 50(2):111–125, 2002.

[24] P. Torr and C. Davidson. IMPSAC: A synthesis of importance sampling and random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):354–365, 2003.

[25] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

[26] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[27] Z. Wen and T. Huang. Capturing subtle facial motions in 3D face tracking. In *Proceedings of International Conference of Computer Vision*, pages 1343–1350, 2003.