Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 25 (2007) 274–284
image AND vision COMPUTING
www.elsevier.com/locate/imavis

Outlier rejection in high-dimensional deformable models

Christian Vogler [a,*], Siome Goldenstein [b], Jorge Stolfi [b], Vladimir Pavlovic [c], Dimitris Metaxas [c]

[a] Gallaudet Research Institute, Gallaudet University, 800 Florida Avenue NE, HMB S-433 Washington, DC 20002-3695, USA
[b] Instituto de Computação, Universidade Estadual de Campinas, Caixa Postal 6176, Campinas, SP 13084-971, Brazil
[c] Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854-8019, USA

Received 16 October 2004; received in revised form 11 August 2005; accepted 11 October 2005

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 three 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.
© 2006 Elsevier B.V. All rights reserved.

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 non-rigid, from only 2D 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 2D 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 [1]. Many applications are tailored to one particular class of data, and do not necessarily generalize well to different types of inputs. Tracking 3D models from a noisy image stream, as we do in this paper, without a proper statistical representation of the noise, is a daunting task (Section 2). 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 three techniques to detect such outliers. The first method works in the image space, but requires some numerical approximations (Section 3.1). For this reason, we develop a novel method that detects outliers in parameter space, without the need for any approximations (Sections 3.2 and 3.2.1). Finally, the third method consists of a robust extension to the detection of outliers in *parameter* space (Section 3.2.3). We test and

* Corresponding author. Tel.: +1 202 651 5400.
E-mail addresses: christian.vogler@gallaudet.edu (C. Vogler), siome@ic.unicamp.br (S. Goldenstein), stolfi@ic.unicamp.br (J. Stolfi), vladimir@cs.rutgers.edu (V. Pavlovic), dnm@cs.rutgers.edu (D. Metaxas).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.10.010

validate all three approaches in experiments on a sequence of images, to which we have applied markers to provide ground truth data (Section 4).

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 [2]. Within a similar framework, some researchers use anthropometric data and inspired deformations to generate faces [3]. A learning-based statistical model can help tracking of face models [4]. Eigen-based approaches, such as PCA decompositions, can successfully track, fit, and even recognize objects [5–8]. In [9], the head is modeled as a cylinder, and in [10] as a plane, and in [7,11] tracking is used for animation. In [12], tracking uses adaptive texture. A powerful deformable volumetric model has also been used for fast face tracking [13] and subtle motion [14] capture. Integration of distinct cues have been used to reduce the effect of outliers particular to a specific algorithm [15,16]. In [16], the deformable model cues have their distributions measured, but it ignores the effects of possible outliers, and in [17], 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 [18], MLESAC [19], and IMPSAC [20]. M-estimators look for optimum weighting of each element, instead of just trying to discard outliers [21–23]. There are good overviews and comparisons of these methods, both in the general statistics literature [24], 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 fully determined by the value of a parameter vector $\vec{q}$. Some parts of $\vec{q}$ are responsible for the rigid motion, whereas others change its shape. For example, in a deformable model for face tracking, as in Fig. 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

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

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

During a tracking session, for every new frame k, we look for the value of the parameter vector $\vec{q}_k$ 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 non-linear, and in the general case there is no closed-form solution for this inverse problem.

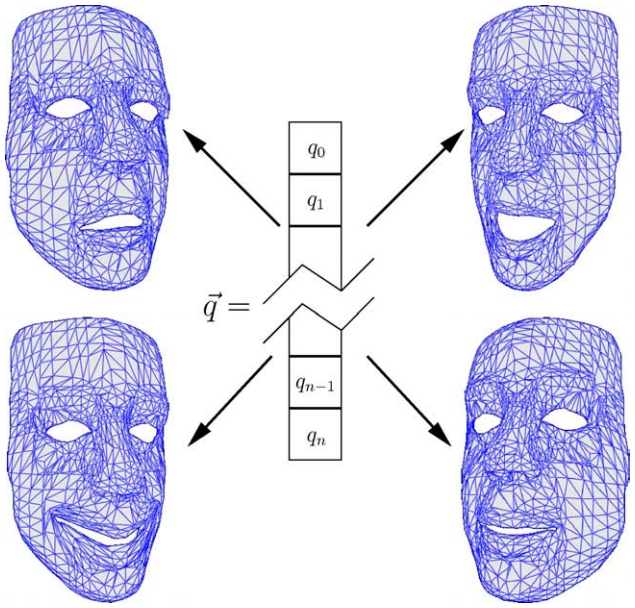

Fig. 1. Parameter vector controls shape and rigid transformation.

Nevertheless, we can use computer vision algorithms, such as point trackers, edge trackers and optical flow, to find image-to-image correspondences between frames. We then use these 2D displacements $\vec{f}_i$, which we also call *image forces*, to adjust the value of $\vec{q}_{k-1}$ to $\vec{q}_k$ iteratively. This adjustment is nothing more than a local optimization in parameter space: the search for a new $\vec{q}$ that minimizes the sum of the magnitude of all image forces $\vec{f}_i$.

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

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

where B_i is the projected Jacobian of the model at point p_i to which the image force $\vec{f}_i$ is applied. Using the generalized force $\vec{f}_g$, we solve the dynamical system

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

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

Obviously, the quality of the solution depends on the estimate of $\vec{f}_g$, which in turn depends on the quality of the image force estimates $\vec{f}_i$. Computer vision algorithms are known for their inconsistency, so the quality of the estimates of $\vec{f}_i$ 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 are only a small number of outliers that will be washed out through the averaging process of Eq. (2). Unfortunately, ignoring outliers results in noisy tracking at best. At worst, in the presence of noisy data, or corrupted video sequences, where there is a significantly larger percentage of outliers than normal, the system may 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 Fig. 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 three approaches to outlier rejection that combine the information from the 2D image and the model Jacobians. The first approach works in image space. In contrast, the second and third approach works in parameter space, with the difference that the latter uses robust statistical estimators.

Although different, these three approaches all rely on the same assumption that the generalized forces fit an underlying unimodal probability distribution. In earlier work [16,26], we proved that, assuming uniformly bounded and independently—but not identically—distributed image forces, this underlying distribution converges to a Gaussian as the number of contributions grows to infinity (Lindeberg theorem). All three approaches take advantage of this fact by either projecting this distribution into image space or using it directly in parameter space. Additionally, using the Berry–Essen theorem, we can estimate a bound for the error of a Gaussian approximation for this distribution. In [26], we determined experimentally that after using 60–80 forces, the change in the error bound begins to slow, so for a good approximation we need to use at least that many generalized forces after outlier rejection.

If we leave outliers undetected, they introduce a bias into the Gaussian approximation of the generalized forces, because the probability estimates for the outliers' corresponding image forces, as described in [16], are likely to be wildly incorrect. In addition, because they change the upper bound of the image forces' probability distributions, the error in the Gaussian approximation increases, as well, according to the Berry–Essen theorem. Hence, a much higher number of data points would be needed to smooth out effects of outliers. Discarding outliers removes this bias, and keeps the overall estimation more precise and stable. The net effect is that we need fewer image forces overall than we would if we ignored outliers, provided that we have at least 60–80 image forces left over after the outlier removal. In an inductive procedure such as tracking, this means that we can keep track and remain reasonably faithful longer and in the presence of temporary occlusions.

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.

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 [17], before the fusion with the observation, to weed out the outliers. At frame k, we have the multivariate Gaussian prediction $\tilde{\vec{q}}_k$ of the parameter vector for frame $k+1$, the covariance matrix $\Lambda_{\tilde{\vec{q}}_k}$ giving the uncertainty in the prediction, and the associated predicted points in image space

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

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

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

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 (Fig. 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

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

where $\vec{x} = p_i - \tilde{p}_i$ 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

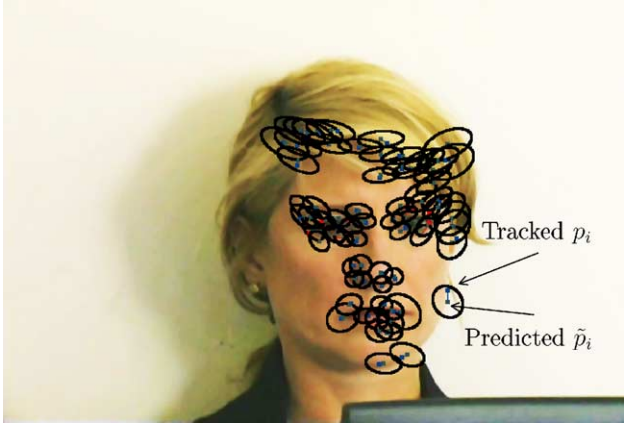

Fig. 2. Image space-based outlier rejection.

most serious limitation of this approach is that it does not take into account the non-linearities 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 non-linearity 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 Eq. (2), $\vec{f}_g$ is defined as the sum of the image forces' projections into parameter space

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

where

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

illustrated in Fig. 3.

If we were follow the same idea as in the previous section, we would like to predict the value of the generalized force $\vec{f}_g$ for frame $k+1$, and use the Mahalanobis metric on this higher dimensional Gaussian distribution of the parameter-space projection of the 2D forces (Eq. (7)). Unfortunately, this approach turns out to be very unreliable, because $\vec{f}_g$ 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 $\vec{f}_g$, is based on estimating the distribution of $\vec{f}_g$ in the current frame and rejecting any force $\vec{f}_{gi}$ that is not compatible with this distribution. In previous work, we showed that, as $\vec{f}_g$ is the sum of the parameter-space projection of many 2D forces (Eq. (2)), a Gaussian reasonably represents the distribution of the resulting generalized force $\vec{f}_g$ [16,26], so once we have estimated $\vec{f}_g$ we can, in principle, again use the Mahalanobis metric to test the individual $\vec{f}_{gi}$. 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.

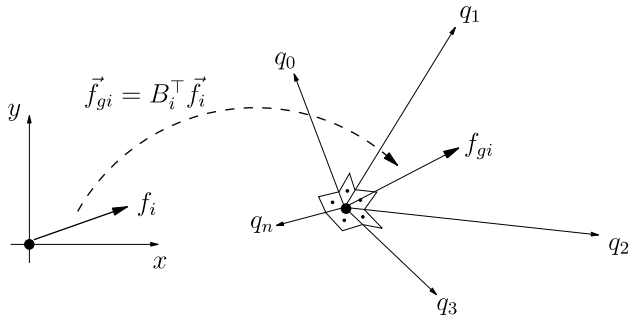

Fig. 3. Projection of 2D force into parameter space.

3.2.1. Simple estimation of the distribution of $\vec{f}_g$

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 then at a first glance could calculate the mean μ and the covariance matrix Λ via

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

Even if we leave the question of robustness aside, these 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 jth row of a generalized force $\vec{f}_{gi}$ can have two different meanings:

(1) The jth parameter q_j is already at a minimum, and thus $(\partial p_i/\partial q_j)\cdot\vec{f}_i = 0$.
(2) The point p_i does not depend on the jth parameter, and so $(\partial p_i/\partial q_i) = 0$.

In the first case, the jth 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 $\vec{f}_i$ 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:

$$S_j := \left\{ \vec{f}_{gi} \middle| \frac{\partial p_i}{\partial q_j} \neq 0 \right\}. \tag{10}$$

Then

$$\mu_j = \frac{1}{N_j} \sum_{\vec{f}_{gi} \in S_j} f_{gi,j}, \tag{11}$$

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

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

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

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

$$\Lambda = \begin{bmatrix} \square & & \\ & \square & \\ & & \square \end{bmatrix}, \quad (13)$$

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

The parameter independence assumption holds the key to coping with the unobservability phenomenon. It is simplistic from a mathematical point of view, because there certainly is overlap between points that are affected by different parameters; for instance, the rigid body transformation parameters affect every point on the 3D model, whereas the eyebrow parameters affect only a limited region on the forehead. From an engineering point of view, however, this assumption has the effect of decoupling the rigid body transformations and various groups of facial deformations from one another. In practice, this effect is desirable, because individual tracked points often provide conflicting information for the estimation of the rigid and non-rigid parameters—such as a subject moving his eyebrows upward, while simultaneously moving his head downward.

3.2.2. Generalization of the estimation of the distribution of $\vec{f}_g$

The estimates of the mean and covariance in Eqs. (11) and (12) are not robust, because their breakdown point is zero. Hence, a single outlier can affect the estimates arbitrarily. As a result, they suffer from the *masking problem*, where multiple outliers can hide one another's existence to the point that none of them are detected [27]. Therefore, robust statistical estimators for calculating the mean and covariance of the generalized forces are indicated. Unfortunately, Eqs. (11) and (12) cannot be easily adapted to an arbitrary statistical estimator. However, the point of view in Eq. (13) provides a clue: each of the blocks in the matrix is independent from the others, and consequently each block can be calculated independently on the subspace that it affects.

In the following discussion we assume, without loss of generality, that the parameters have been arranged to conform to the block-diagonal pattern in Eq. (13). Formally, we define an equivalence relation over the parameters:

$$q_j \equiv_P q_k \quad \text{iff} \quad S_j = S_k. \quad (14)$$

Let $\mathcal{EP}_l$ be the l^{th} equivalence class of $\equiv_P$ and define

$$\mathcal{Q}_l := \{ j | q_j \in \mathcal{EP}_l \} \quad (15)$$

to be the indices of the parameters belonging to each equivalence class. Let

$$\mathcal{S}_l := S_j, \quad \text{where} \quad q_j \in \mathcal{EP}_l; \quad (16)$$

that is, the set of nodes for which the parameters in that particular equivalence class are observable. Similarly, let

$$\mathcal{F}_l := \{ \vec{f}_{gi} | p_i \in \mathcal{S}_l \}; \quad (17)$$

that is, the set of forces that affect the parameters in the equivalence class $\mathcal{EP}_l$. Then the projection of the generalized forces into the subspace generated by an equivalence class is a set of $|\mathcal{Q}_l|$-dimensional vectors, with $|\mathcal{Q}_l|$ being the size of the equivalence class:

$$\mathcal{EP}_l(\{\vec{f}_{gi}\}) := \left\{ \begin{bmatrix} \vdots \\ | \\ f_{g1,j} \\ \vdots \end{bmatrix}, \ldots, \begin{bmatrix} \vdots \\ | \\ f_{gi,j} \\ \vdots \end{bmatrix}, \ldots \middle| j \in \mathcal{Q}_l \wedge \vec{f}_{gi} \in \mathcal{F}_l \right\}, \quad (18)$$

where $f_{gi,j}$ is the jth component of $\vec{f}_{gi}$, as before. Intuitively, this set consists of only the observable forces, restricted to the components that affect the parameters in that equivalence class. With these definitions, the estimate of the mean can be rewritten as

$$\vec{\mu} = \begin{bmatrix} \text{mean}(\mathcal{P}_1(\{\vec{f}_{gi}\})) \\ \vdots \\ \text{mean}(\mathcal{P}_l(\{\vec{f}_{gi}\})) \end{bmatrix}, \quad (19)$$

where l is the number of equivalence classes in $\equiv_P$, and mean$(\cdot)$ is an arbitrary statistical estimator for the mean. Likewise, the estimate of the covariance matrix can be rewritten as

$$\Lambda = \begin{bmatrix} \boxed{\text{cov}\left(\mathcal{P}_1\left(\{\vec{f}_{gi}\}\right)\right)} & & 0 \\ & \boxed{\ddots} & \\ 0 & & \boxed{\text{cov}\left(\mathcal{P}_l\left(\{\vec{f}_{gi}\}\right)\right)} \end{bmatrix}, \quad (20)$$

where cov$(\cdot)$ is an arbitrary statistical estimator for the covariance. Each of these cov$(\cdot)$ entries in Eq. (20) corresponds to a block in Eq. (13).

3.2.3. Robust estimation of the distribution of $\vec{f}_g$

We are now in a position to apply robust estimators to the problem of estimating the mean and covariance matrix of the generalized forces, even in the presence of parameter unobservability. Because we operate in a high-dimension, our choice of estimators is limited by statistical and computational efficiency concerns. For instance, the popular median absolute deviation from the median (MAD) [28] is defined for 1D data. The field of statistics has come up with various approaches to computing the median in higher dimensions, such as [29,30], but they are neither equivariant under affine transformations, nor computationally efficient. Chakraborty and Chaudhuri proposed an alternative approach that is computationally efficient and affine equivariant [31],

which would be a suitable, albeit very complicated, median estimator for our purposes.

If we would like to avoid the complexities inherent in median-based estimators, we can turn to robust mean-based estimators. In particular, Rousseeuw and Leroy proposed the minimum volume ellipsoid (MVE) and the minimum covariance determinant (MCD) estimators [24]. Of these two, the MCD has the more desirable statistical properties, and has already been tested in computer vision applications [32]. Moreover, there exists a fast algorithm, called FAST-MCD, for computing both the robust mean and covariance, with implementations available in Fortran, Matlab, and S-PLUS [33].

We choose the FAST-MCD algorithm for estimating the robust mean and covariance, because it is both simpler and computationally more efficient than the median-based approaches. Conceptually, this algorithm operates by selecting a subset containing h of the N sample points and computing the mean and covariance of this subset. The desired subset is the one for which the determinant of the associated covariance matrix is at a minimum. The breakdown value of this estimator is $(N-h-\dim)/N$ (where $N/2 \leq h \leq N$, and dim is the dimension of the data set) [33], so we can make it as robust as we like, with breakdown points between 0 and 50%. The flip side of a high breakdown point is the reduced statistical efficiency. If we raise the breakdown value too high, with too few samples, the estimate of the mean and the covariance will be inaccurate. Hence, aiming for a high breakdown point is possible only with a lot of sample points, often more than are available from the generalized forces.

Fortunately, there are two factors that alleviate concerns about the tradeoff between statistical efficiency and the breakdown point. First, because we apply the estimator to each equivalence class separately, and assemble the mean and covariance matrix from the pieces as per Eqs. (19) and (20), dim stays relatively small. Second, in typical circumstances, we expect that the majority of the forces generated by the computer vision algorithms (Section 2) have reasonable values, and so the percentage of outliers is typically much smaller than 50%.

The complexity of FAST-MCD is quadratic in the dimension of the sample space, and takes at least time $O(N \log N)$ to find the minimum covariance determinant subset. In addition, the setup of the initial guesses for the subsets takes at least 500 iterations of complexity $O(N)$ [33]. Consequently, compared to the non-robust estimator, its run time is at least by a factor of $O(\log N)$ slower, plus the large constant arising from the 500+ initial iterations. It is, therefore, likely that using the MCD estimator will run into computational efficiency concerns sooner than the simple, non-robust estimator, but on an asymptotic scale their complexity is similar.

3.2.4. 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 jth component $\vec{f}_{gi,j} = 0$, and the distance between 0 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 k observable parameters, we can build a projection matrix P_i, with dimensions $k \times n$, composed of only 0s and 1s, that projects a force from the n-dimensional parameter space into the k-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[1]

$$(\vec{f}_{gi} - \vec{\mu})^\top P_i^\top P_i \Lambda^{-1} P_i^\top P_i (\vec{f}_{gi} - \vec{\mu}) \leq \text{threshold}. \tag{21}$$

It is tempting, but incorrect, to choose a constant value for the threshold experimentally. Instead, the choice is governed by statistical considerations. For sufficiently large sample sets, the Mahalanobis distances are χ^2-distributed.

A common choice for the cutoff is to choose the threshold such that anything beyond the 97.5% quantile of the distribution's tails is rejected as an outlier [24]:

$$\chi^2_{\text{dof}}(\text{threshold}) = p, \tag{22}$$

$$\chi^2_k(\text{threshold}) = 0.975, \tag{23}$$

where $\text{dof} = k$ denotes the degrees of freedom, or equivalently the dimension k of the subspace. Thus, the threshold depends on the dimension of the subspace and must be chosen adaptively for each generalized force $\vec{f}_{gi}$.

4. Validation with 3D face tracking

Recall that the image space-based rejection method involves a linearization, whereas the parameter space-based rejection methods do not. All other factors equal, we therefore, would expect the parameter space-based rejection methods to do better. In addition, between the two parameter space-based rejection methods, we would expect the one in conjunction with the MCD estimator for calculating the mean and covariance of the generalized forces to perform best overall, because of robustness concerns. To test this hypothesis, we implemented the following three methods in our existing 3D face tracking system:

- Image space-based rejection using a predictive Kalman filter (Section 3.1).
- Parameter space-based rejection using the simple,

[1] Note that this formula conceptually describes the computation of the observable subspace Mahalanobis metric. From an algorithmic point of view, it is faster to compute the Mahalanobis metrics separately for each equivalence class $\mathcal{EP}_l$ and generalized force $\vec{f}_{gi}$, using cov $(\mathcal{P}_l(\{\vec{f}_{gi}\}))$ and mean $(\mathcal{P}_l(\{\vec{f}_{gi}\}))$ for all $\mathcal{E}\wp_l$ that are affected by a particular force $\vec{f}_{gi}$. Accumulating the separate metrics from all $\mathcal{E}\wp_l$ for each $\vec{f}_{gi}$ gives the same result as Eq. (21), but drastically reduces the number of matrix inversions and projections.

non-robust estimator for the mean and covariance of the forces (Section 3.2.1).
- Parameter space-based rejection using the MCD estimator for the mean and covariance of the forces (Section 3.2.3).

4.1. Experimental set-up

For a quantitative evaluation of these three methods, we produced a video sequence at 60 Hz at a 640×480 resolution, where we physically drew markers on the subject's face (Fig. 4). We downsampled this sequence to 15 Hz at a 320×240 resolution. At this low frame rate and relatively low level of detail, fast facial movements are prone to causing failures in the low-level computer vision algorithms (Section 2), and thus are more suitable for an evaluation of the efficacy of the outlier rejection methods than higher resolutions or frame rates. We extracted the 2D image position of each marker for all frames in a semi-automated manner based on thresholding, and associated each marker with a node on the deformable model. This association was the same for all experiments and remained constant over the course of each tracking run.

During the tracking experiments, we excluded all regions in the vicinity of the markers from being selected as image features, so that the markers would not provide inappropriately strong hints to the tracking system. After recovering the parameter vector $\vec{q}$ for the 3D face model at each frame, we projected the model nodes that we had associated with the markers back into 2D image space, and compared the projected positions with the actual positions extracted from the frames. Our evaluation criterion is the distance between the projected and actual marker positions, where a distance of zero means perfect tracking.

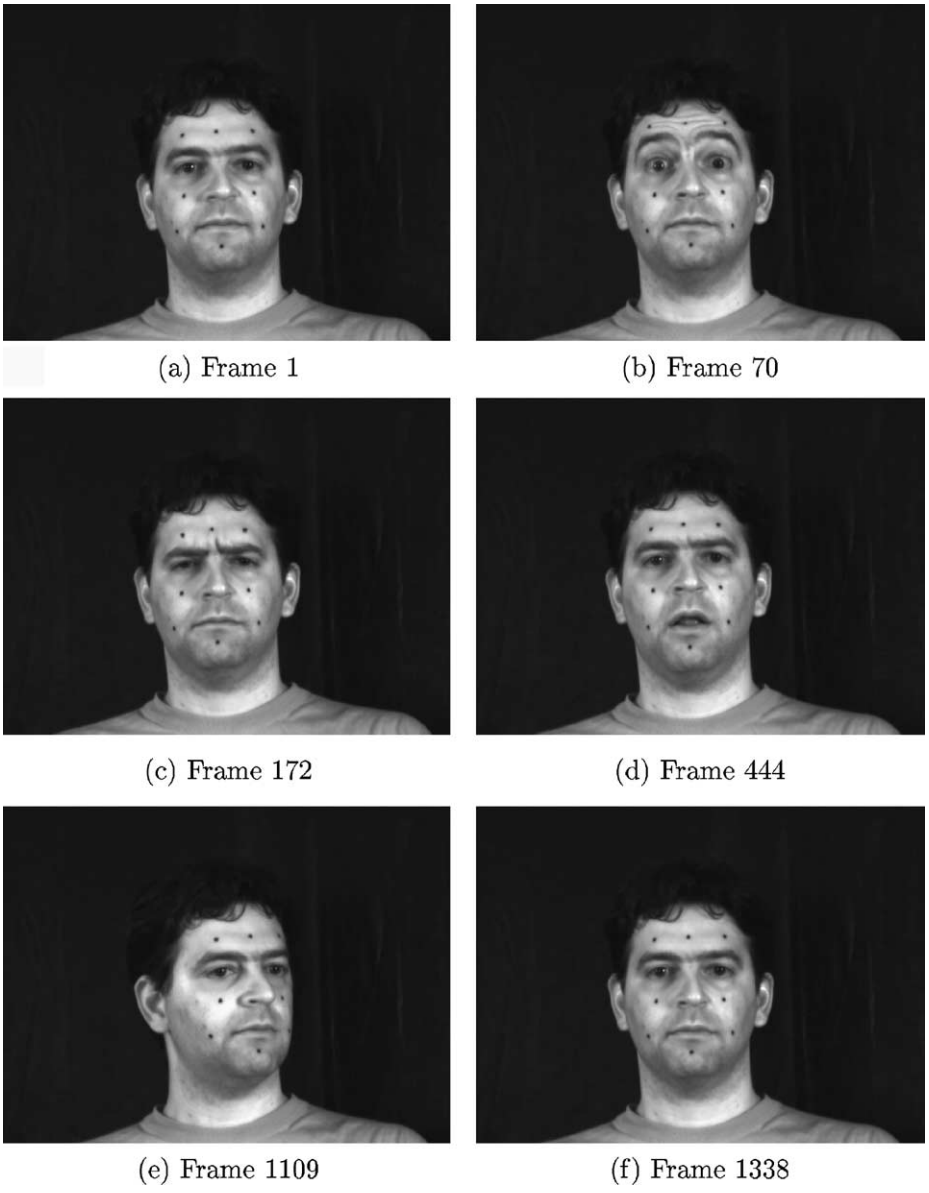

Fig. 4. Subject with eight markers physically drawn on the face, showing some face deformations that we tracked during the validation sequence.

We used an all-purpose deformable face model with 1101 nodes, 2000 triangles, and 11 parameters that controlled both the rigid transformation and the facial deformations (eyebrows, lip stretching, smiling, jaw opening, etc.) of each of these nodes (see Fig. 1 in Section 2). 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 [3,6,34].

For the purposes of the validation experiments, we fitted the model in a semi-automated way: the user manually selected a few dozen model-image correspondences. Fitting then consisted of solving Eq. (3), with the user correspondences as the image forces $\{\vec{f}_i\}$ (Section 2, Eq. (2)), using a finite-element inspired set of shape deformations. Because the definitions of the facial expression deformations are independent of the base mesh [16], the model was ready for tracking immediately after fitting.

To isolate and understand the effect of the outlier rejection techniques, we simplified the tracking procedure as much as possible, leaving only the implementation of the KLT tracker to generate the image forces, and removing tethering procedures. This algorithm selected the model points that were most suitable for tracking, based on the characteristics of the image features [35], and then tracked them to obtain their displacements $\vec{f}_i$ at each new frame. For all experiments, we set the maximum number of iterations (Eq. (3)) to 600. The tracking speed was an average of 0.5 s per frame on a Pentium 4 running at 2.4 GHz.

4.2. Experimental results

We compared the effects of no outlier rejection, image space-based rejection (Section 3.1), simple non-robust parameter space-based rejection (Section 3.2.1), and robust MCD parameter space-based rejection (Section 3.2.3). Fig. 5 shows the comparative results of the best configuration options for these four techniques, and the respective percentages of removed outliers. Using no outlier rejection at all fares the worst, with the error quickly growing out of control. Image space-based rejection fares somewhat better, keeping track for 500 more frames than no rejection, but eventually loses track, as well. Simple parameter space-based rejection yields a significant improvement over image space-based rejection, and replacing the simple mean and covariance estimator with the robust MCD estimator provides another boost. Thus, the validation results agree with the hypothesis formulated at the beginning of this section.

The differences in the percentages of removed outliers across the three different rejection methods are revealing. In the case of image space-based rejection, the spikes in the rejection percentages coincide with the moment when the tracking system starts to drift, starting at frame 2000. Such large spikes occur when the Kalman filter's prediction of the model

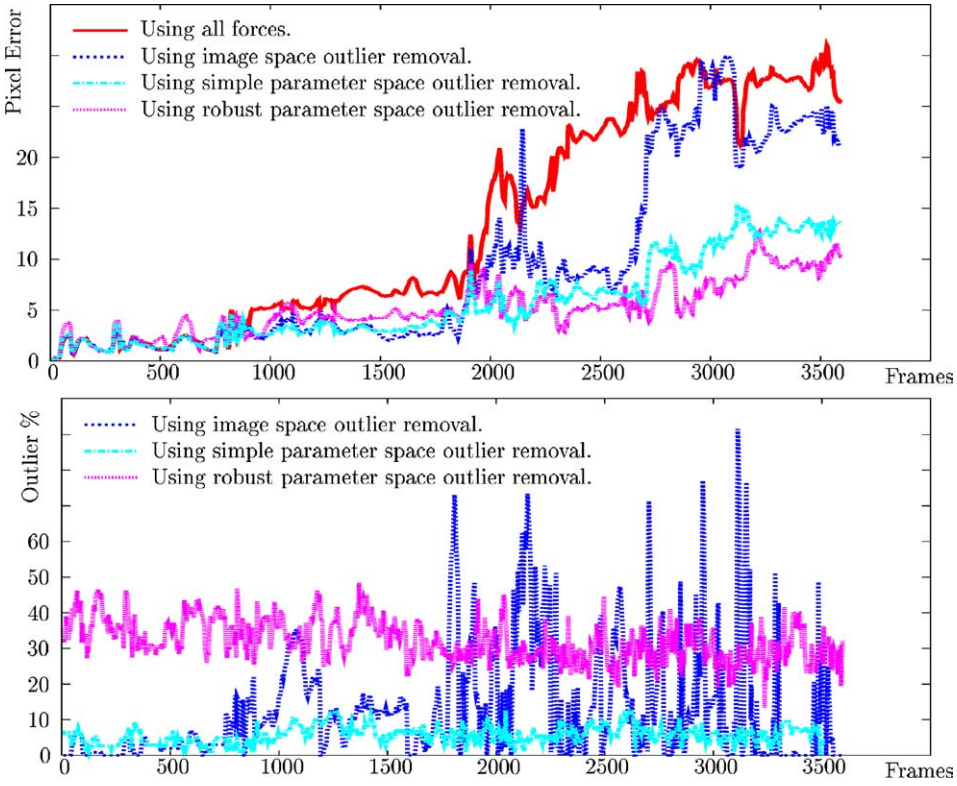

Fig. 5. *Top*: mean errors of visible markers, with best to worst: parameter space MCD (small dashed purple), parameter space simple (dashed cyan), image space (dotted blue), no outlier rejection (solid red). *Bottom*: percentage of forces dropped by each of the three outlier rejection methods. The average number of forces in each frame, before outlier removal, was 120.

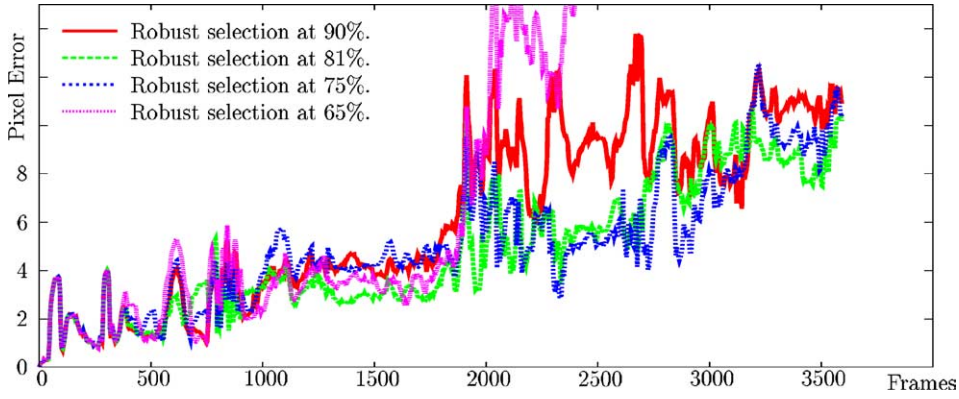

Fig. 6. Mean errors of visible markers.

(Section 3.1) disagrees strongly with the actual tracked estimate of the model, which is likely to happen when the system starts to lose track. The non-robust parameter space rejection percentages stay low throughout the entire tracking sequence, even when the system starts to lose track, which is an indicator of the masking problem mentioned in Section 3.2.2. The MCD rejection percentages are surprisingly high and uniform throughout the entire sequence, which cannot be explained by a high number of outliers alone. We discuss the reasons for this result further in Section 4.3.

As we discussed in Section 3.2.3, the MCD estimator poses a tradeoff between robustness and the accuracy of the estimate, via the selection of the percentage of points that are used to calculate the mean and covariance. More specifically, smaller percentages are statistically more robust; that is, the resulting estimate is less sensitive to large numbers of outliers, but require a larger number of initial points. Conversely, larger percentages require fewer points, but the resulting estimate is more sensitive to large numbers of outliers. To test the effect of the selection percentage, we ran a series of MCD estimator experiments, where this percentage constituted the only variable. Fig. 6 shows several representative results. As the selection percentage decreases[2], tracking accuracy first improves, reaching an optimum around 75–80%, and then decreases sharply. This behavior is also in line with the paper describing the FAST-MCD algorithm, which suggests 75% as a good compromise between accuracy and robustness.

4.3. Examination of outlier behavior

To understand the behavior of our robust parameter space outlier detector better, we apply it to tracking a natural sign language speaker telling a story. In this particular stretch of time, our subject is saying: "I remember that a while ago I was driving on highway 40 …" The face is frequently occluded by the moving hands, making it very hard to track.

[2] Note that 100% is equivalent to the simple, non-robust estimator.

Like in the validation experiments in Section 4, we only use the KLT point tracker to select the points to track, and to find the associates image forces. Without outlier rejection, the system loses track after the first pass of the hand in front of the face. Recently, some researchers dealt with this kind of problem by employing known images of the model to reacquire tracking [36]. MCD-based outlier rejection handles this occlusion without inducing any drift in the face model, and without requiring any kind of recovery.

In Fig. 7, we show three snapshots of the tracking sequence: before, during, and just after the hand passes the face. On the left we show the tracked model overlaid over the input image, and on the right we show all points that have been selected by the KLT tracker for generating image forces. The points denoted by green squares are accepted by the MCD estimator, whereas the ones denoted by red circles are rejected. In particular, in the second and third row, it correctly rejects the points that were pulled astray by the occluding movement of the hand. In essence, there is an outlier wavefront that moves along the hand.

Note that even without occlusion, our method rejects a large number of points that it considers unfit. In general, MCD-estimated outliers tend to be clustered in regions, which indicates that at any given moment, depending on position, visibility, and movement, certain regions in the face yield more accurate estimates of the facial parameters than others. For instance, in the first row, the system rejects points along the left contour in low-contrast regions, where a normal point tracker does not fare well at estimating the rigid part of the head movement. In the third row, the contrast along the contour is better, but the points are still rejected, because the estimates do not fit well into the overall downward movement of the face at this frame. Thus, the effect of the MCD estimator is not only to reject outliers, but also to select a subset of points that form the 'best' tracking hypothesis; that is, the hypothesis with the lowest covariance matrix. This behavior also explains why in the validation experiments in Section 4.2 the MCD outlier rejection percentage was consistently above 30%.

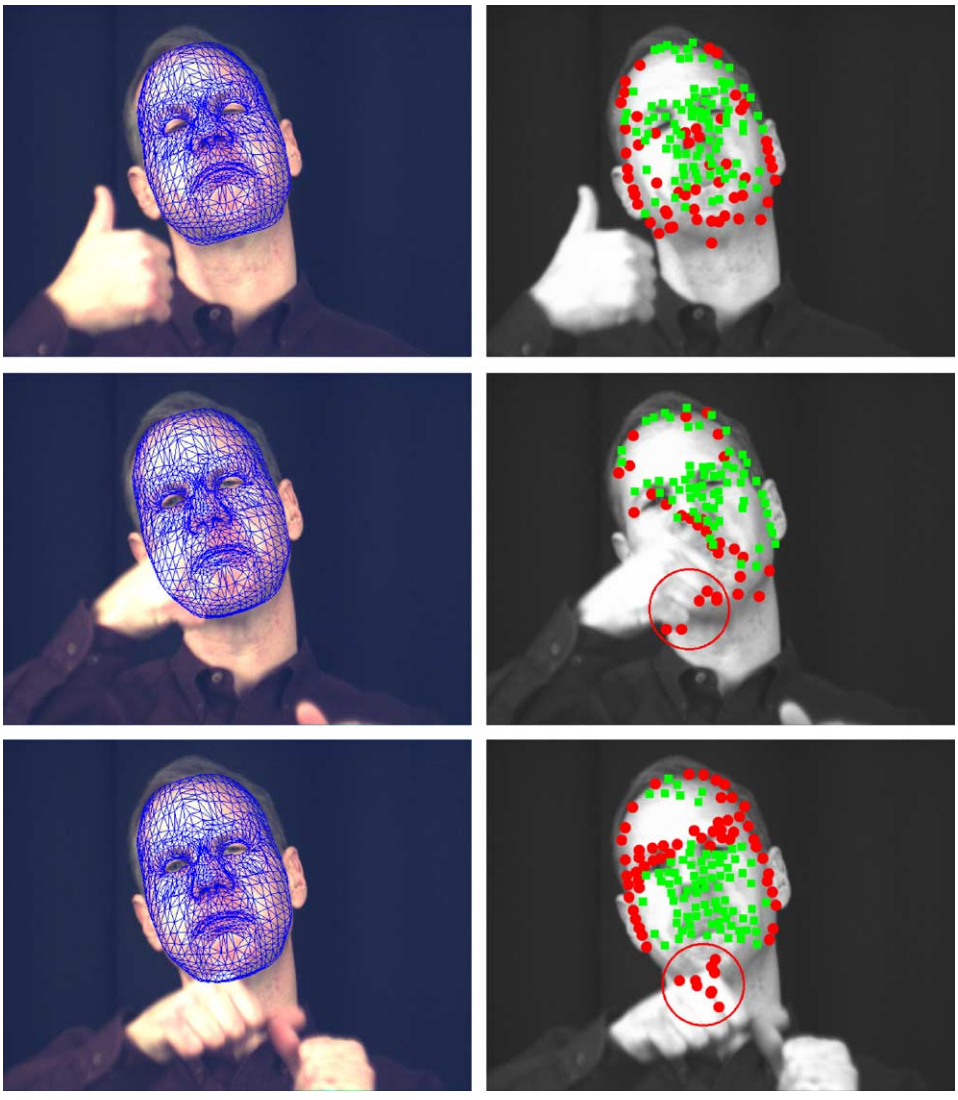

Fig. 7. Outlier behavior in a tracking sequence with occlusions. Red circles denote outliers detected by the MCD algorithm; green squares denote good points. The tracking process is able to survive the occlusion of the face by the hands by detecting the encircled outliers in the bottom two rows. Source of tracking sequence: National Center for Gesture and Sign Language Resources, http://www.bu.edu/asllrp/ncslgr.html

5. Conclusions

In this paper, we have introduced three approaches for rejection of the outlier forces generated from low-level computer vision algorithms. The first approach uses a multidimensional Gaussian prediction, and the second one builds a covariance matrix from the available data. Finally, the third one extends the second one with a robust statistical estimator. In all 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.

The third approach adds robust estimators for the necessary mean and covariance matrices of the second method. It has a

better breakdown point, at the cost of an extra complicated-to-implement procedure, and the need for more points to achieve accurate estimates. The robust estimator's computational complexity is somewhat higher than the non-robust estimator's, because of an extra factor of O(log N) and larger constants, but the former's asymptotic behavior is sufficiently similar to the latter's to be suitable for most deformable tracking applications.

Overall, these factors and the validation results demonstrate the clear superiority of the parameter space methods over the image space method. The robust parameter space approach is slightly better than its simpler counterpart, at the cost of a more complex procedure and some application limitations.

Finally, it is important to point out that, although tempting, it is wrong to provide a manually determined value for the final Mahalanobis threshold. The χ^2 distribution can estimate the threshold cutoff point based on the desired rejection percentage. This process is automatic, and takes into account the effect of different dimensions on the tails of the distributions.

Acknowledgements

The research in this paper was supported by NASA Cooperative Agreements 9-58 with the National Space Biomedical Research Institute, CNPq, FAPESP, FAEPEX-Unicamp 1679/04, research scientist funds by the Gallaudet Research Institute, NSF CNS-0427267, and NSF CNS-0428231.

References

[1] P. Meer, C.V. Stewart, D.E. Tyler, Robust computer vision: an interdisciplinary challenge, in: Computer Vision and Image Understanding, vol. 78, 2000, pp. 1–7.
[2] D. Samaras, D. Metaxas, P. Fua, Y. Leclerc, Variable albedo surface reconstruction from stereo and shape from shading, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2000, pp. 480–487.
[3] D. DeCarlo, D. Metaxas, M. Stone, An anthropometric face model using variational techniques, in: Proceedings of the SIGGRAPH, 1998, pp. 67–74.
[4] M. Brand, R. Bhotika, Flexible flow for 3D nonrigid tracking and shape recovery, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2001, pp. 315–322.
[5] H. Murase, S.K. Nayar, Visual learning and recognition of 3-d objects from appearance, International Journal of Computer Vision 14 (1995) 5–24.
[6] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: SIGGRAPH, 1999, pp. 187–194.
[7] F. Pighin, R. Szeliski, D. Salesin, Resynthesizing facial animation through 3D model-based tracking, in: Proceedings of International Conference of Computer Vision, 1999, pp. 143–150.
[8] A. Romdhani, T. Vetter, Efficient, robust and accurate fitting of a 3D morphable model, in: Proceedings of International Conference of Computer Vision, 2003, pp. 59–66.
[9] M. Cascia, S. Sclaroff, 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) (2000) 322–336.
[10] M. Black, Y. Yacoob, Recognizing facial expressions in image sequences using local parameterized models of image motion, International Journal of Computer Vision 25 (1) (1997) 23–48.
[11] F. Pighin, R. Szeliski, D. Salesin, Modeling and animating realistic faces from images, International Journal of Computer Vision 50 (2) (2002) 143–169.
[12] L. Brown, 3D head tracking using motion adaptive texture-mapping, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2001, pp. 998–1005.
[13] H. Tao, T. Huang, Visual estimation and compression of facial motion parameters: elements of a 3D model-based video coding system, International Journal of Computer Vision 52 (2) (2002) 111–125.
[14] Z. Wen, T. Huang, Capturing subtle facial motions in 3D face tracking, in: Proceedings of International Conference of Computer Vision, 2003, pp. 1343–1350.
[15] D. de Carlo, D. Metaxas, Optical flow constraints on deformable models with applications to face tracking, International Journal of Computer Vision 38 (2) (2000) 99–127.
[16] S. Goldenstein, C. Vogler, D. Metaxas, Statistical cue integration in DAG deformable models, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7) (2003) 801–813.
[17] S. Goldenstein, C. Vogler, D. Metaxas, 3D facial tracking from corrupted movie sequences, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2004.
[18] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.
[19] P. Torr, A. Zisserman, MLESAC: a new robust estimator with application to estimating image geometry, Computer Vision and Image Understanding 78 (1) (2000) 138–156.
[20] P. Torr, C. Davidson, IMPSAC: a synthesis of importance sampling and random sample consensus, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (3) (2003) 354–365.
[21] R. Marona, Robust m-estimators of multivariate location and scatter, The Annals of Statistics 4 (1976) 51–67.
[22] H. Chen, P. Meer, Robust computer vision through kernel density estimation, in: Proceedings of European Conference of Computer Vision, 2002, pp. 236–250.
[23] H. Chen, P Meer, Robust regression with projection based m-estimators, in: Proceedings of International Conference of Computer Vision, 2003, pp. 878–885.
[24] P. Rousseeuw, A. Leroy, Robust Regression and Outlier Detection, Wiley, London, 1987.
[25] P. Torr, D. Murray, The development and comparison of robust methods for estimating the fundamental matrix, International Journal of Computer Vision 24 (3) (1997) 271–300.
[26] S. Goldenstein, Statistical cue estimation for model-based shape and motion tracking, PhD thesis, University of Pennsylvania, 2002.
[27] R.R. Wilcox, Applying Contemporary Statistical Techniques, Elsevier, Amsterdam, 2003.
[28] F.R. Hampel, The influence curve and its role in robust estimation, Journal of the American Statistical Association 69 (1974) 383–393.
[29] R.Y. Liu, On a notion of data depth based on random simplices, Annals of Statistics 18 (1990) 405–414.
[30] B.M. Brown, Statistical use of spatial median, Journal of the Royal Statistical Society Series B 45 (1983) 25–30.
[31] B. Chakraborty, P. Chaudhuri, A note on the robustness of multivariate medians, Statistics and Probability Letters 45 (1999) 269–276.
[32] P.J. Rousseeuw, S.V. Aelst, Positive-breakdown robust methods in computer vision, in: K. Berk, M. Pourahmadi (Eds.), Computing Science and Statistics, vol. 31, Interface Foundation of North America, Inc., Fairfax Station, VA, 1999, p. 451.
[33] P.J. Rousseeuw, K.V. Driessen, A fast algorithm for the minimum covariance determinant estimator, Technometrics 41 (1999) 212–223.
[34] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. Salesin, Synthesizing realistic facial expressions from photographs, in: Proceedings of the SIGGRAPH, 1998, pp. 75–84.
[35] J. Shi, C. Tomasi, Good features to track, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 1994, pp. 593–600.
[36] M. Dimitrijevic, S. Ilic, P. Fua, Accurate face models from uncalibrated and ill-lit video sequences, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2004, pp. 1034–1041.