Affine arithmetic based estimation of cue distributions in deformable model tracking

Siome Goldenstein Christian Vogler Dimitris Metaxas
V.A.S.T. Laboratory - University of Pennsylvania
200 S 33rd Street, Philadelphia, PA 19104 USA
{siome,cvogler,dnm}@graphics.cis.upenn.edu

Abstract

In this paper we describe a statistical method for the integration of an unlimited number of cues within a deformable model framework. We treat each cue as a random variable, each of which is the sum of a large number of local contributions with unknown probability distribution functions. Under the assumption that these distributions are independent, the overall distributions of the generalized cue forces can be approximated with multidimensional Gaussians, as per the central limit theorem.

Estimating the covariance matrix of these Gaussian distributions, however, is difficult, because the probability distributions of the local contributions are unknown. We use affine arithmetic as a novel approach toward overcoming these difficulties. It lets us track and integrate the support of bounded distributions without having to know their actual probability distributions, and without having to make assumptions about their properties. We present a method for converting the resulting affine forms into the estimated Gaussian distributions of the generalized cue forces. This method scales well with the number of cues.

We apply a Kalman filter as a maximum likelihood estimator to merge all Gaussian estimates of the cues into a single best fit Gaussian. Its mean is the deterministic result of the algorithm, and its covariance matrix provides a measure of the confidence in the result. We demonstrate in experiments how to apply this framework to improve the results of a face tracking system.

1. Introduction

Deformable models are an important technique in computer vision for robust tracking tracking nonrigid motion, such as human faces, and full human body movements. These types of motions have important applications particularly in surveillance and human computer interaction, as well as virtual environments.

The shape of a deformable model is characterized by its parameterization. Each point on the surface of a model is uniquely determined by this parameterization. As long as this surface is C_1-differentiable with respect to the model parameters, it is possible to determine what effect moving a point on the surface has on the model parameters. Typically, deformable model frameworks take advantage of this fact by using various low-level computer vision cues, such as edge tracking and optical flow, to act on points or regions of the deformable model, and consequently on the model parameters based on Lagrangian dynamics. Cues provide information on how the shape and the position of the deformable model change over time.

As long as only one cue is used at a time, estimation of the model parameters is a straightforward process. The picture changes dramatically, however, when multiple cues act on a model at the same time. Due to the noise inherent in most low-level computer vision cues, different cues will exhibit different degrees of reliability at different points on the model surface. Even worse, often the distribution of the noise is unknown, thus making it difficult to capture it with a probability distribution. As a result, the optimal integration of cues to yield the best possible parameter estimate of the model is a difficult and open research problem.

Previous approaches integrated the cues either by using a direct sum of the cues, or through the design of hard constraints [4] that subjugate some cues to others. A direct sum ignores that some cues may be more reliable than others at a given point in time, whereas a hard constraint causes problems if the dominant cue is unreliable or changes over time. In this paper we describe a new method for combining the cues dynamically in an optimal manner within a statistical framework.

In the deformable model framework, the cues are mapped into parameter space as *generalized forces* that act on and change the model parameters. The cues, in turn, typically are the sum of a large number of local image contributions, such as the positions of various edges from an

edge tracker. If we model these underlying contributions as random variables, the cues, and hence the generalized forces, will also be random variables. We present a way to estimate these generalized forces' probability distributions without any prior knowledge about the cues' and local contributions' probability distributions.

Since computer vision deals with discrete domains, it is reasonable to assume that the values of the local contributions lie in bounded regions [15, 12]. These regions provide the support of each contribution's probability distribution. We use these regions to represent the contributions, instead of representing them with normal numbers. To describe the regions mathematically we use *affine forms. Affine arithmetic* defines the normal algebraic operations, such as sums and multiplications, over these affine forms (Sec. 3), which allow us to obtain the multidimensional affine forms of the generalized forces from the affine forms of the individual contributions (see Sec. 2 and Eq. 2).

Because the number of independent contributions to a cue is typically very large, we can apply the *central limit theorem*, which states that the distribution of the sum can be represented as a Gaussian distribution. We show in this paper how to estimate the parameters of this Gaussian distribution from the generalized forces expressed as multidimensional affine forms.

The mean of the Gaussian distribution is the most likely estimate of that cue, whereas the covariance matrix describes the confidence of the cue. This description leads to a seamless integration into a *Kalman filter* framework as a maximum likelihood estimator of the combined generalized force from all cues.

The rest of the paper is organized as follows: We discuss related work, then provide the mathematical background for deformable models and affine arithmetic. We then describe our novel approach to converting a multidimensional affine form into a Gaussian distribution that can be used in the Kalman filter framework. We demonstrate in experiments on synthetic images and real face tracking data that the statistical approach is more robust than the simple direct sum of the cues.

1.1. Related Work

Deformable Models have been used in a variety of areas and applications. In computer vision for tracking and shape estimation [8, 4, 24], in computer graphics for synthesis and simulation [6] and in medical applications for reconstruction, modeling and diagnosis [22, 2]. Most of these approaches have been deterministic; that is, they did not address the statistical uncertainties inherent in tracking images, and in fitting the models to a particular shape or image.

Throughout the years different statistical approaches have been explored for computer vision applications. Among others, there are Condensation [14, 11], and Kalman Filters for tracking and for predicting motion [3, 23]. Such approaches either do not scale well with the problem size, or make assumptions about the shape and the characteristics of the probability distribution functions. However, often nothing is known about them except their bounds.

Interval arithmetic [19, 20] manipulates such bounds directly. It has been largely used in numerical analysis and optimization [9], and computer graphics [25, 10]. This approach suffers from overestimation of bounds, and the complete loss of information on how bounds in multidimensional intervals are correlated. More recently, affine arithmetic has been developed to overcome these shortcomings [1, 26]. It has previously been applied to numerical optimization [5, 17, 13].

In this paper we apply affine arithmetic to embed deformable models within a statistical framework. This approach allows us to avoid making assumptions about the probability distribution functions, unlike most previous statistical approaches to computer vision, and it scales well with the number of parameters used in the deformable model description.

2. Deformable Model Tracking

Fundamentally, a deformable model framework is a Lagrangian dynamics system parameterized by a vector $\mathbf{q}$ [18]:

$$\dot{\mathbf{q}} = \mathcal{F}(\mathbf{q}). \tag{1}$$

As the tracking process evolves, we integrate $\dot{\mathbf{q}}$ with the Euler integration method.

We obtain the coordinates of each point on a deformable model through a series of linear and non-linear operations applied over its parameters $\mathbf{q}$. We convert contributions from a 2D visual cue i applied to a point $\mathbf{p}_j$ on the deformable model to generalized forces $\mathbf{f}_g^{c,j}$, which act on the model parameters. The conversion to generalized forces is obtained through the application of the matrix $\mathbf{B}_j$, which is the projection of the model Jacobian at point $\mathbf{p}_j$ to image space. The sum of these gives us the generalized force $\mathbf{f}_{g,c}$ for cue c:

$$\mathbf{f}_{g,c} = \sum_j \mathbf{f}_{g,cj} = \sum_j \mathbf{B}_j^\top \mathbf{f}_{cj}, \tag{2}$$

where $\mathbf{f}_{cj}$ is the image force from cue c at point $\mathbf{p}_j$ in image space. If these image forces are independent random variables, and the number of elements in this sum is large, the central limit theorem [21] (CLT) states that a a multidimensional Gaussian is a good representation of $\mathbf{f}_{g,c}$, even

if the probability distributions of the image forces are unknown.

$$\mathbf{f}_{g,c} = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Lambda_c}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^\top \mathbf{\Lambda}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)} \tag{3}$$

where $\boldsymbol{\mu}_c$ is the mean vector and $\mathbf{\Lambda}_c$ is the covariance matrix. Usual proofs of the CLT require i.i.d. random variables, but if these variables' distributions satisfy extra conditions (third absolute moment bounded and $|f_k'| < M$), then only independence is necessary [7]. In our application, the domains of the distributions are bounded, which implies that the third absolute moments are bounded. Even if we relax the condition of the derivative on the distributions, it is still possible to bound the error of the approximated Gaussian distributions [7].

To integrate the contribution of the cues with the deformable model, we need to combine the $\mathbf{f}_{g,c}$ into the single generalized force $\mathbf{f}_g$ in the best possible way:

$$\mathbf{f}_g = \mathcal{F}\left(\mathbf{f}_{g,1}, \mathbf{f}_{g,2}, \cdots\right). \tag{4}$$

In a deterministic framework, $\mathcal{F}$ is simply a weighted sum of the $\mathbf{f}_{g,c}$. In a statistical framework, we use a maximum likelihood estimator, as described in Sec. 5.

In the next section we describe the tools that will allow us to estimate the mean vector $\boldsymbol{\mu}_c$ and the covariance matrix $\mathbf{\Lambda}_c$ that statistically describe each $\mathbf{f}_{g,c}$.

3. Affine Arithmetic

To model the visual cues (point tracker, optical flow, etc.) properly as random variables we need to know the probability distribution function of their values. In the general case, this is a complicated problem that might need strong knowledge of the application. The assumptions made while estimating the distributions might not also translate well for different applications.

To get around these problems we model only the region of the values that the cue's image forces take; that is, the support of their probability distributions. Affine arithmetic provides the framework to manipulate these regions. Calculating the generalized forces operates on regions, instead of real numbers.

The result of applying these operations to the cue's image forces is a region in the model parameter space representing $\mathbf{f}_{g,c}$. Because of the large number of individual image forces, it is then possible to estimate the parameters $\boldsymbol{\mu}_c$ and $\mathbf{\Lambda}_c$ of the Gaussian that represent $\mathbf{f}_{g,c}$ based on the properties of this region, as described in Sec. 4.

3.1. Affine forms and the mathematical operations

The basic atom of the affine arithmetic is called an *affine form*. An affine form $\hat{a}$ represents an interval and is represented as:

$$\hat{a} = a_0 + a_1\varepsilon_1 + a_2\varepsilon_2 + \cdots + a_m\varepsilon_m. \tag{5}$$

In $\mathbb{R}^1$ the coefficients a_i are real numbers, whereas in $\mathbb{R}^n$ they are n-dimensional vectors. The ε_i are symbolic real variables whose values are unknown, but guaranteed to lie in the interval $[-1 \ldots 1]$ with $E[\varepsilon_i] = 0$. The quantity a_0 is called the *central value* (mean), and the ε_i are called the *noise symbols*. Each noise symbol ε_i represents an independent component of the total uncertainty. By scaling ε_i with the a_i from Eq. 5, it is possible to obtain arbitrarily large intervals of uncertainty.

As an example, consider a two-dimensional vector describing a cue's image force j from a visual cue c, $\mathbf{f}_{cj}$. It can be described as an affine form as follows:

$$\hat{\mathbf{f}_{cj}} = \begin{pmatrix}\hat{f}_x \\ \hat{f}_y\end{pmatrix} = \begin{pmatrix}10 \\ 20\end{pmatrix} + \begin{pmatrix}2 \\ -3\end{pmatrix}\varepsilon_1 + \\ + \begin{pmatrix}1 \\ 0\end{pmatrix}\varepsilon_2 + \begin{pmatrix}0 \\ 1\end{pmatrix}\varepsilon_3 + \begin{pmatrix}-1 \\ 4\end{pmatrix}\varepsilon_4 \tag{6}$$

This representation, shown in Figure 1, describes a vector whose mean is at $(10, 20)^\top$. If $\hat{f}_x$ and $\hat{f}_y$ were independent, their spanned intervals would be $[6 \ldots 14]$ and $[12 \ldots 28]$, respectively (plotted as the light gray on Figure 1). However, because $\hat{f}_x$ and $\hat{f}_y$ share the noise symbols ε_1 and ε_4, their variations are not independent. In fact, $\mathbf{f}_{cj}$ has to lie in the dark region of Figure 1.

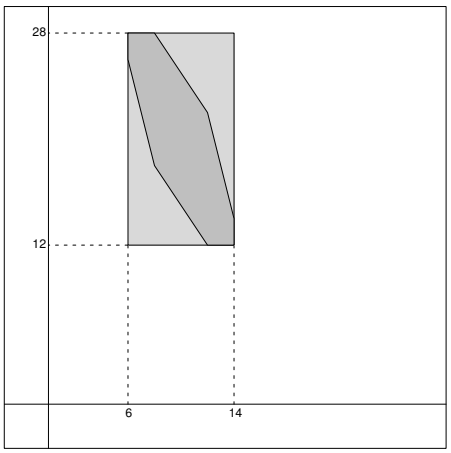


Figure 1. Joint range of two partially dependent quantities in AA.

Interval arithmetic [19] is another way to track intervals; each interval is just represented as the pair $[a \ldots b]$. Affine arithmetic has a significant advantage in that it provides tighter bounds. In addition, unlike interval arithmetic, it preserves the information on the correlation between the components of $\mathbf{f}_{g,c}$. In Figure 1 the light gray region shows

the best possible bound that could be represented by interval arithmetic.

For each operation on real numbers we have to define a counterpart for affine forms. Affine operations like

$$\hat{z} = \alpha\hat{x} + \beta\hat{y} + \zeta, \tag{7}$$

are calculated exactly, where

$$\hat{x} = x_0 + \sum_{i=1}^{m} x_i\varepsilon_i \quad \hat{y} = y_0 + \sum_{i=1}^{m} y_i\varepsilon_i \quad \hat{z} = z_0 + \sum_{i=1}^{m} z_i\varepsilon_i,$$

where $\hat{x}$, $\hat{y}$, and $\hat{z}$ are affine forms, and α, β, and ζ are real constants. The definition of this operation is

$$z_0 = \alpha x_0 + \beta y_0 + \zeta \quad \text{and} \quad z_i = \alpha x_i + \beta y_i. \tag{8}$$

Note that any operation defined on two affine forms also defines this operation on an affine form and a scalar, because a scalar s is trivially represented by the affine form $a_0 = s$. For this reason, the operations necessary to convert an image force to a generalized force in Eq. 2 are still valid when the image forces are affine forms.

Non-affine operations require a more careful analysis. For each operation we have to determine an affine approximation for the valid range. We then introduce an extra independent noise symbol is to carry the introduced error, thus keeping the interval valid. A thorough description on how to do the appropriate operations (reciprocate, multiplications, exponentials, trigonometric, etc.) can be found in [26].

We now show how to obtain the Gaussian parameters $\boldsymbol{\mu}_c$ and $\boldsymbol{\Lambda}_c$ from an affine form describing the generalized cue force $\mathbf{f}_{g,c}$, so that we can use it in a maximum likelihood estimator (MLE).

4. Obtaining a Gaussian random variable from an affine form

After using affine arithmetic on the image forces, we obtain an affine form for the generalized cue force $\mathbf{f}_{g,c}$ from Eq. 2, where

$$\hat{\mathbf{f}}_{g,c} = \mathbf{a}_0 + \sum_{i=1}^{m} \mathbf{a}_i\varepsilon_i. \tag{9}$$

Since we have summed many independent elements (i.e., the image forces), we know from the central limit theorem that a Gaussian distribution will adequately approximate $\mathbf{f}_{g,c}$. We need to estimate the mean vector $\boldsymbol{\mu}_c$ and the covariance matrix $\boldsymbol{\Lambda}_c$ of this distribution.

The mean vector is

$$\begin{aligned}\boldsymbol{\mu}_c = E\left[\hat{\mathbf{f}}_{g,c}\right] &= E[\mathbf{a}_0] + \sum_{i=1}^{m} E[\mathbf{a}_i\varepsilon_i] \\ &= \mathbf{a}_0 + \sum_{i=1}^{m} \mathbf{a}_i E[\varepsilon_i],\end{aligned} \tag{10}$$

but since all noise symbols ε_i are equally distributed around the origin,

$$\boldsymbol{\mu}_c = \mathbf{a}_0. \tag{11}$$

We break the estimation of the covariance matrix $\boldsymbol{\Lambda}_c$ into two problems: to determine the eigenvectors and the eigenvalues of this matrix. The eigenvectors are the major axes of the region defined by the affine form. The eigenvalues are the variances of the affine form along these axes.

4.1. Eigenvectors of $\boldsymbol{\Lambda}_c$

To find the eigenvectors of $\boldsymbol{\Lambda}_c$ we look for the minimum-volume hyperrectangle[1] bounding the solid described by the affine form. This hyperrectangle can be represented by an affine form with exactly n noise symbols (where n is also the size of the deformable model's parameter vector $\mathbf{q}$), all of which point along orthogonal direction vectors. In normalized form these vectors are the desired eigenvectors.

Given an orthonormal basis $\{\mathbf{w}_i\}$ of $\mathbb{R}^n$, there is an affine form $\mathbf{w}$ that represents the hyperrectangle oriented along the axes $\mathbf{w}_i$, and which contains the solid described by $\hat{\mathbf{f}}_{g,c}$. Its description is

$$\mathbf{w} = \mathbf{w}_0 + \sum_{i=1}^{m} \alpha_i\mathbf{w}_i\varepsilon_i, \tag{12}$$

where α_i is the sum of the absolute values of the projections of the $a_i, i \geq 1$ (from Eq. 5) over $\mathbf{w}_i$:

$$\alpha_i = \sum_{j=1}^{m} |\mathbf{a}_i \cdot \mathbf{w}_j|. \tag{13}$$

The problem is now reduced to finding $\{\mathbf{w}_i\}$ that minimizes

$$\prod_{i=1}^{n} \alpha_i.$$

We can rotate vectors $\mathbf{w}_k$ and $\mathbf{w}_l$ by an angle θ without destroying the orthonormality of $\{\mathbf{w}_i\}$:

$$\mathbf{w}_k' = \mathbf{w}_k \cos\theta - \mathbf{w}_l \sin\theta \tag{14}$$

$$\mathbf{w}_l' = \mathbf{w}_k \sin\theta + \mathbf{w}_l \cos\theta. \tag{15}$$

[1] a higher-dimensional rectangle

This operation only changes α_k and α_l. To minimize the volume along this rotation operation it is necessary to find θ such that $\alpha'_k\alpha'_l$ (the α corresponding to $\mathbf{w}'_k$ and $\mathbf{w}'_l$) is at a minimum. Our algorithm is the application of this local optimization over all pairs of vectors starting from an arbitrary orthonormal basis $\{\mathbf{e}_i\}$:

Initializes: $\{\mathbf{w}_i\} \leftarrow \{\mathbf{e}_i\}$
for $k = 1$ **to** $n - 1$ **do**
  for $j = k + 1$ **to** n **do**
    Find θ that minimizes $\alpha'_k\alpha'_l$
    Rotate $\mathbf{w}_k$ and $\mathbf{w}_l$ by θ
    $\alpha_k \leftarrow \alpha'_k, \alpha_l \leftarrow \alpha'_l$
  end for
end for

4.2. Eigenvalues of $\mathbf{\Lambda}_c$

The eigenvalues λ_i of $\mathbf{\Lambda}_c$ are the variances σ_i^2 along the axes of the eigenvectors $\{\mathbf{w}_i\}$:

$$\lambda_i = E\left[\left((\hat{\mathbf{f}}_{g,c} - \boldsymbol{\mu}_c) \cdot \mathbf{w}_i\right)^2\right].$$

By plugging in Eqs. 9 and 11

$$\lambda_i = E\left[(\sum_{j=1}^{m} \mathbf{w}_i \cdot \mathbf{a}_j\varepsilon_j)^2\right],$$

but ε_j are by definition independent, so

$$\begin{aligned}\lambda_i &= \sum_{j=1}^{m} E\left[(\mathbf{w}_i \cdot \mathbf{a}_j\varepsilon_j)^2\right] \\ &= \sum_{j=1}^{m} (\mathbf{w}_i \cdot \mathbf{a}_j)^2 E\left[\varepsilon_j^2\right]\end{aligned} \tag{16}$$

We can bound λ_i from above by

$$\lambda_i \leq \max_j(\sigma^2_{\varepsilon_j}) \sum_{j=1}^{m} (\mathbf{w}_i \cdot \mathbf{a}_j)^2. \tag{17}$$

If all noise symbols ε_j are identically distributed (since they are by definition independent, this would imply IID), we can simplify Eq. 16 by noting that $E[\varepsilon_j] = 0$:

$$\lambda_i = E\left[\varepsilon^2\right] \sum_{j=1}^{m} (\mathbf{w}_i \cdot \mathbf{a}_j)^2 = \sigma^2_\varepsilon \sum_{j=1}^{m} (\mathbf{w}_i \cdot \mathbf{a}_j)^2. \tag{18}$$

5. Merging the Cues

To obtain the probability distribution of the generalized force $\dot{\mathbf{q}}$, we integrate the Gaussian distributions of the cues' generalized forces that we obtained in the previous section with a MLE. In the case of Gaussian distributions, the optimal MLE is a simple, non-predictive Kalman filter[16]. Its input are Gaussian distributions, and it yields another Gaussian distribution whose mean $\boldsymbol{\mu}$ is the maximum likelihood estimate of $\dot{\mathbf{q}}$, and whose covariance matrix $\mathbf{\Lambda}$ is an estimate of its confidence.

Each cue is described by the mean vector $\boldsymbol{\mu}_c$ and the covariance matrix $\mathbf{\Lambda}_c$. We iteratively apply the Kalman MLE over each one of the cues. After each iteration the Kalman MLE holds the best estimate of $\hat{\boldsymbol{\mu}}_c$ and $\hat{\mathbf{\Lambda}}_c$ for the cues already processed.

Initializes: $\hat{\boldsymbol{\mu}}_1 \leftarrow \boldsymbol{\mu}_1, \quad \hat{\mathbf{\Lambda}}_1 \leftarrow \mathbf{\Lambda}_1$
for $c = 2$ **to** nc **do** {where nc = number of cues}
  $b_c \leftarrow \hat{\mathbf{\Lambda}}_{c-1}(\hat{\mathbf{\Lambda}}_{c-1} + \mathbf{\Lambda}_c)^{-1}$
  $\hat{\boldsymbol{\mu}}_c \leftarrow \hat{\boldsymbol{\mu}}_{c-1} + b_c(\boldsymbol{\mu}_c - \hat{\boldsymbol{\mu}}_{c-1})$
  $\hat{\mathbf{\Lambda}}_c \leftarrow (\mathbf{I} - b_c)\hat{\mathbf{\Lambda}}_{c-1}$
end for
$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_{nc}, \quad \mathbf{\Lambda} \leftarrow \hat{\mathbf{\Lambda}}_{nc}$

6. Application and Experiments

As an application we augmented our deformable model tracking system to use affine arithmetic inside the cue models, and then to convert it to Gaussians using the method described in the previous sections. We represent the parameters of the deformable model as a vector of scalars, which we integrate with the maximum likelihood estimate $\boldsymbol{\mu}$ of $\dot{\mathbf{q}}$ in the Lagrangian dynamic system, obtained by converting the affine forms of the cues' image forces to affine forms of generalized forces, summing them up to form the cues' generalized forces $\mathbf{f}_{g,c}$, and converting them to Gaussian distributions. Currently we do not use the confidence estimate in the covariance matrix $\mathbf{\Lambda}$ during the integration process; however, future work should make use of it.

In the next subsections we describe, for each cue, how we have constructed the initial random variables that will be propagated to the final Gaussians, and present the results of our method.

6.1. Point Tracker

The point tracker tracks high-contrast points on the deformable model in image space. The criterion for choosing a point's position in the next frame is the minimum *sum of squared differences* (SSD) over a patch within a small region around the point's position in the current frame. The distribution of the SSDs in the region around the point provides us with information on the confidence of the tracked point's position.

Intuitively, the smaller the difference between any two SSDs, the higher is the uncertainty between the two pixels

$\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_j$ corresponding to these two SSDs. We express this relationship in the affine form that describes the position of the tracked point $\mathbf{p}_i$ at pixel $\hat{\mathbf{p}}_i$ by extending its bounds in the x and y directions in image space to cover both points. Rather than choosing an arbitrary cutoff value for the difference between the SSDs to decide whether to extend the bounds to include a point, we attenuate the magnitude of the bounds with a decaying exponential function. We describe the position of $\hat{\mathbf{p}}_i$ with the affine form

$$\hat{\mathbf{p}}_i = \mathbf{Pt}_i + b_x \begin{pmatrix} 1 \\ 0 \end{pmatrix} \varepsilon_1 + b_y \begin{pmatrix} 0 \\ 1 \end{pmatrix} \varepsilon_2,$$

where $\mathbf{Pt}_i$ is the point tracker's estimate for $\mathbf{p}_i$, and b_x and b_y are the bounds

$$\begin{pmatrix} b_x \\ b_y \end{pmatrix} = \max_j \{ ||(\mathbf{Pt}_i - \hat{\mathbf{p}}_j)|| \, g \, e^{-s(\log SSD_j - \log SSD_i)} \}, \tag{19}$$

where SSD_i is the SSD at the tracked point's position, SSD_j is the SSD for point $\mathbf{p}_i$ at image coordinates $\hat{\mathbf{p}}_j$, and g and s depend on the size of the patch used for computing the SSD, and the region searched. The cue's image force is the difference between the affine form $\hat{\mathbf{p}}_i$ and the point $\mathbf{p}_i$'s position in the previous frame.

6.2. Edge Tracker

For this paper, we have modeled a simple statistical edge tracker. The mean of the force is obtained through the gradient of a potential field provided by a smoothed edge detector.

The affine form has two components, one along the direction of the mean, and one perpendicular to it. The first component is inversely proportional to the magnitude of the gradient of the field — the lower the potential field values, the lower is the confidence, and thus the higher the variance has to be. The second component, in the direction perpendicular to the gradient of the potential field, is $k_1 x + k_2$, where x is the first component, and $k_1 > 1$ and $k_2 > 0$ are constants.

The rationale is that the uncertainty along the edge is always higher than the uncertainty perpendicular to the edge, since there is no information on whether the edge moved along this direction.

6.3. Optical Flow

We implemented a statistical version of a simple optical flow [27]. Assuming a constant flow in a patch around the desired point, the estimated optical flow $\mathbf{v}$ is a least squares problem, and we solve it through the two-dimensional system

$$A^\top W^2 A \, \mathbf{v} = A^\top W^2 \, \mathbf{b} \tag{20}$$

where

$$A^\top W A = \begin{bmatrix} \sum_i \nabla x_i^2 w_i^2 & \sum_i \nabla x_i w_i^2 \nabla y_i \\ \sum_i \nabla x_i w_i^2 \nabla y_i & \sum_i \nabla y_i^2 w_i^2 \end{bmatrix},$$

$$A^\top W^2 \, \mathbf{b} = \begin{bmatrix} \sum_i \nabla x_i w_i^2 \nabla t_i \\ \sum_i \nabla y_i w_i^2 \nabla t_i \end{bmatrix}$$

The magnitude of the spatial gradient is a good measure of the reliability of the optical flow estimate at that point. This magnitude can, through an inverse relation, determine the how large is the confidence region in the direction of the optical flow.

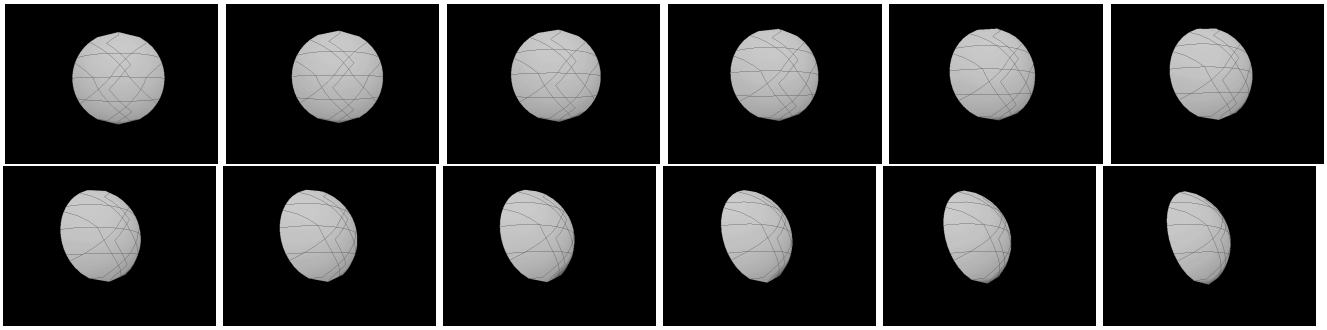
The eigenvalues of $A^\top W^2 \, \mathbf{b}$ have a direct relation with the gradient; they tell us about the properties of that image's neighborhood, and how the aperture problem affects the results. Two high eigenvalues mean a very good estimate, two low eigenvalues mean that the optical flow is not well defined, one high and one low eigenvalue mean that we have a lower confidence in the direction perpendicular to the estimated flow.

We use the higher eigenvalue to estimate one uncertainty bound along the direction of the flow, which is inversely proportional to the larger eigenvalue. We estimate another uncertainty bound along the direction perpendicular to the flow based on the ratio of the eigenvalues.

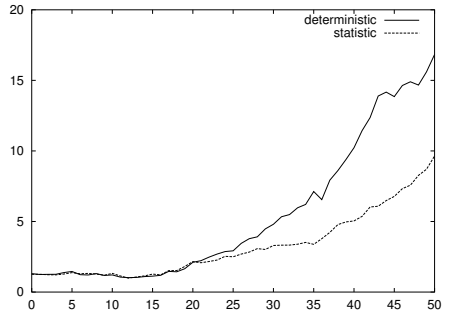
6.4. Results

In this section we show two examples, one synthetic and one with real data. For the synthetic example we build a model of a wedge, and rendered images from two different points of view. We tracked these two images with two different point trackers as described in Section 6.1, one for each point of view. Because of the low amount of texture in the images, these present as difficult a test for a point tracker-based cue as can be. In figure 2(a) we show some snapshots of one of the image sequences, as well as the plot of the maximum error in pixels for each camera in both the deterministic and statistic methods.

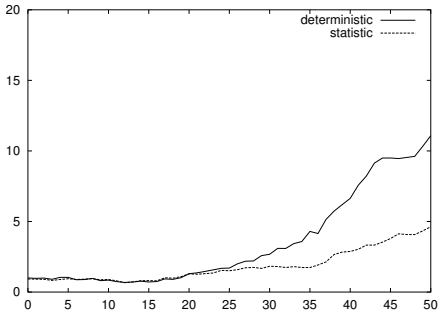
For the real example we have constructed a low resolution face model that has 31 parameters defining the facial anthropomorphic proportions. These parameters are responsible for the fitting of the model to an individual. On top of these we have added dynamic motion parameters to capture the deformations from facial expressions. In Figures 6.4 and 6.4 we show two real tracking sequences obtained from our system. Our method enabled us to track over 300 frames in the face sequence with a single camera.



(a)



(b)



(c)

Figure 2. Synthetic example. 2(a) Snapshots of one image sequence. 2(b) and 2(c) compare the deterministic against the statistic for each one of the cameras.

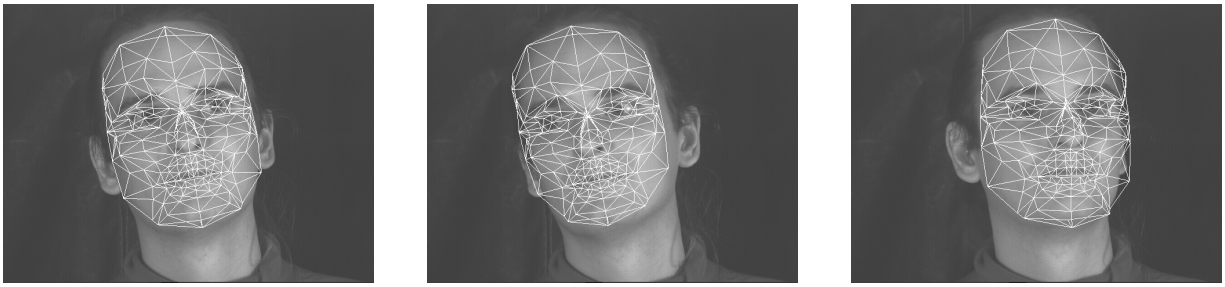


Figure 3. Real images: Tracking of face rotation and translation with statistical methods

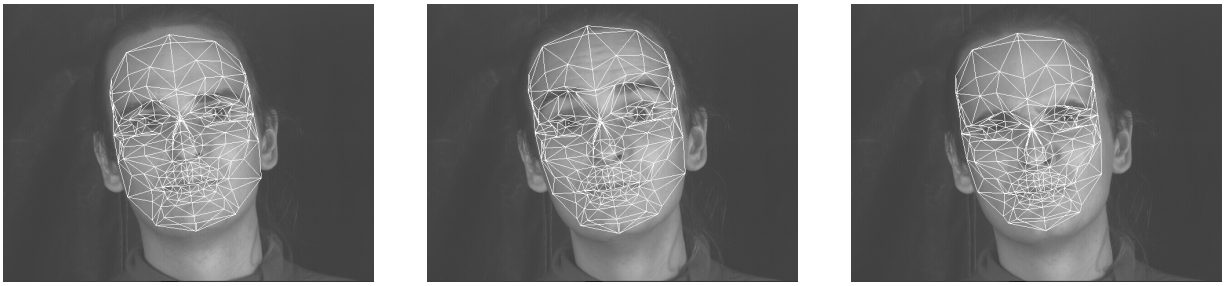


Figure 4. Real images: Tracking of raising eyebrows with simultaneous head tilting with statistical methods

7. Conclusions and Future Work

In this paper we introduced *affine arithmetic* to propagate the boundaries of random variables. We can use this method without prior knowledge of the probability distribution functions, as long as there is a way to obtain an estimate of the random variable's bounds.

Our framework for integration of cues is elegant and adaptive. It allows the dynamic combination of cues that have no interaction with one another; that is, they may come from different cameras, or even different sources of data. The framework scales well with the number of the cues available.

There are still some important open problems to be addressed. If the initial affine form does not cover the entire domain of the distribution of the random variable, part of the probabilities will be discarded. On the other hand, if the interval spanned by the affine form is too large, there will be a loss of correlation information because non-affine operations over affine forms are approximated by local affine ones [26]. More work is required to understand the tradeoff between too-large and too-small regions.

In the short term, integrating a shape from shading cue [24] within the statistical framework should improve the process for the initial fitting of deformable models. In addition, using the Kalman MLE's covariance estimate in a predictive filter may further improve the tracking results.

Acknowledgments

This work was supported in part by an NSF Career Award NSF-9624604, NSF EIA-98-09209, and AFOSR F49620-98-1-0434. The first author was supported by CNPq - Brazilian's "*Conselho Nacional de Desenvolvimento Cientfico e Tecnolgico*". We would like to thank the authors of [26] for allowing us to use Figure 1 and the example that it depicts.

References

[1] M. Andrade, J. Comba, and J. Stolfi. Affine arithmetic. In *Abstracts of the International Conference on Interval and Computer-Algebraic Methods in Science and Engineering (INTERVAL)*, pages 36–40, 1994.

[2] F. Azar, D. Metaxas, and M. Schnall. A 3d deformable model of the breast for predicting mechanical deformations under plate compression during interventional procedures. In *Biomedical Engineering Society Annual Meeting*, 2000.

[3] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *PAMI*, 8(1):90–99, Jan. 1986.

[4] D. de Carlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *IJCV*, 38(2):99–127, July 2000.

[5] L. H. de Figueiredo, R. Van Iwaarden, and J. Stolfi. Fast interval branch-and-bound methods for unconstrained global optimization with affine arithmetic. Technical Report IC-97-08, Institute of Computing, Univ. of Campinas, June 1997.

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

[7] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume II. John Willey & Sons, 1971.

[8] P. Fua and Y. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 16(1):35–56, September 1995.

[9] E. Hansen. *Global Optimization using Interval Analysis*. M. Dekker, 1988.

[10] C. Hu, T. Maekawa, E. C. Sherbrooke, and N. M. Patrikalakis. Robust interval algorithm for curve intersections. *Computer-aided Design*, 28(6-7):495–506, 1996.

[11] M. Isard and A. Blake. C-conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, August 1998.

[12] G. Kamberova and M. Mintz. Minimax rules under zero-one loss for a restricted location parameter. *Journal of Statistical Planning and Inference*, 2(79):205–221, 1999.

[13] M. Kashiwagi. An all solution algorithm using affine arithmetic. In *NOLTA'98 (1998 International Symposium on Nonlinear Theory and its Applications)*, 1998.

[14] M. I. M. and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. of ECCV*, 1998.

[15] R. Mandelbaum, G. Kamberova, and M. Mintz. Stereo depth estimation: a confidence interval approach. In *Proc. of ICCV*, 1998.

[16] P. Maybeck. *Stochastic Models, Estimation, and Control*. Academic Press, 1979.

[17] F. Messine and A. Mahfoudi. Use of affine arithmetic in interval optimization algorithms to solve multidimensional scaling problems. In *IMACS/GAMM International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics*, 1998.

[18] D. Metaxas. *Physics-based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*. Kluwer Academic Publishers, 1996.

[19] R. Moore. *Interval Analysis*. Prentice-Hall, 1966.

[20] R. Moore. *Methods and Applications of Interval Analysis*. SIAM, 1979.

[21] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[22] J. Park, D. Metaxas, and A. Young. Deformable models with parameter functions: Application to heart-wall modeling. In *Proc. of CVPR*, pages 437–442, 1994.

[23] R. Rao. Robust kalman filters for prediction, recognition, and learning. Technical report, Univ. Rochester, 1996.

[24] D. Samaras, D. Metaxas, P. Fua, and Y. Leclerc. Variable albedo surface reconstruction from stereo and shape from shading. In *Proc. of CVPR*, pages 480–487, 2000.

[25] J. M. Snyder. Interval analysis for computer graphics. In *Proc. of SIGGRAPH*, pages 121–130, 1992.

[26] J. Stolfi and L. Figueiredo. *Self-Validated Numerical Methods and Applications*. 21º Colóquio Brasileiro de Matemática, IMPA, 1997.

[27] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.