Cue Integration using Affine Arithmetic and Gaussians

Siome Goldenstein[1], Christian Vogler[1], and Dimitris Metaxas[2]

[1] CIS Department - University Of Pennsylvania
200 S 33rd Street, Philadelphia PA 19104, USA,
{siome,cvogler}@graphics.cis.upenn.edu
[2] CS Department - Rutgers University
110 Frelinghuysen Road, Piscataway, NJ 08854-8019
dnm@cs.rutgers.edu

Abstract. In this paper we describe how the connections between *affine forms*, *zonotopes,* and Gaussian distributions help us devise an automated cue integration technique for tracking deformable models. This integration technique is based on the confidence estimates of each cue. We use affine forms to bound these confidences. Affine forms represent bounded intervals, with a well-defined set of arithmetic operations. They are constructed from the sum of several independent components. An n-dimensional affine form describes a complex convex polytope, called a zonotope. Because these components lie in bounded intervals, *Lindeberg's theorem*, a modified version of the *central limit theorem*, can be used to justify a Gaussian approximation of the affine form.
We present a new expectation-based algorithm to find the best Gaussian approximation of an affine form. Both the new and the previous algorithm run in $O(n^2m)$ time, where n is the dimension of the affine form, and m is the number of independent components. The constants in the running time of new algorithm, however, are much smaller, and as a result it runs 40 times faster than the previous one for equal inputs. We show that using the *Berry-Esseen theorem* it is possible to calculate an upper bound for the error in the Gaussian approximation. Using affine forms and the conversion algorithm, we create a method for automatically integrating cues in the tracking process of a deformable model. The tracking process is described as a dynamical system, in which we model the force contribution of each cue as an affine form. We integrate their Gaussian approximations using a *Kalman filter* as a *maximum likelihood estimator*. This method not only provides an integrated result that is dependent on the quality of each on of the cues, but also provides a measure of confidence in the final result. We evaluate our new estimation algorithm in experiments, and we demonstrate our deformable model-based face tracking system as an application of this algorithm.

keywords: "statistical cue integration", "deformable model tracking", "affine arithmetic", "visual motion"

1 Introduction

One of the most difficult problems in tracking parameterized deformable models is the integration of multiple cues, such as point tracking, edge tracking, and optical flow. 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 automated integration of cues to yield the best possible parameter estimate of the model is a difficult and open research problem.

In this paper we discuss a novel statistical approach to cue integration that is based on the interrelationships between *affine forms*, their manipulation via affine arithmetic, *Gaussian probability distributions*, and *zonotopes*. We demonstrate how known results and techniques from different areas of literature can be integrated and we develop a new method for conversion between affine forms and Gaussians. We demonstrate how to use these results and this method for automated cue integration that avoids making assumptions about the probability distribution of the noise in each of the cues.

In a deformable model framework, each cue (e.g., edges, optical flow) is mapped into parameter space as *generalized forces* that act on the model and change its parameters through a dynamical system. Each cue, in turn, is typically the sum of a large number of local image contributions, such as the positions of various edges from an edge tracker. We use affine forms to represent the support of the local image contributions, while avoiding making assumptions about the actual shape of their probability distribution functions. We use affine arithmetic to sum them up.

Affine forms and affine arithmetic were developed in the nineties as an alternative to classical interval arithmetic. Affine arithmetic provides tighter bounds than interval arithmetic in cascaded operations. Unlike interval arithmetic [1, 2], it also preserves information about mutual dependencies between results. Since then it has been used in numerical applications [3, 4], electrical engineering [5], computer graphics [6, 7], and computer vision [8].

Gaussian probability distributions are a widely-used tool in engineering [9, 10], as they have several desirable properties: preservation of linearity, compactness of representation via the mean and covariance matrix, and several convergence theorems, notably the central limit theorem. Given certain conditions that we discuss in this paper, we can use Lindeberg's theorem [9, pp 262] to show that the sum of the local image contributions making up a cue can be approximated by a Gaussian-distributed random variable, whose support is represented by an affine form. Moreover, we discuss how to bound the error in the approximation.

In [8] a geometry-inspired heuristic was developed to obtain a Gaussian approximation of an affine form. In this paper we develop an improved method to estimate the Gaussian distribution from an affine form, which is approximately

40 times faster. The new method ensures that the estimated Gaussian distribution has the same first-and second-order moments as the respective affine form. Consequently, this estimate is accurate, as long as the conditions for Lindeberg's theorem hold true. In addition, we relax the assumption made in [8] that the distributions of the contributions making up a cue had to be part of the same parametric family.

Zonotopes are convex volumes formed through the Minkowsky sum of line segments. They have been known in the geometry literature for more than a decade [11]. They appear, among other things, in polytope and point interaction [12], support vector machines [13], and in dynamical systems [14]. We show that the region defined by an affine form is a zonotope, and we demonstrate how zonotope theorems affect the algorithmic complexity of converting affine forms to Gaussian probability distributions.

The rest of the paper is organized as follows: We start with a short discussion of previous work, then provide an overview on affine forms and affine arithmetic. We then discuss the requirements of approximating affine forms with Gaussians, the bounds of the error of this approximation, and a new algorithm to find this approximation. We then connect affine forms to zonotopes to provide insights into the computational complexity of the conversion from affine forms to Gaussians. Finally, we describe how to apply this algorithm to integrate cues in a deformable model framework. In particular we present results of this integration technique in deformable model-based face tracking.

1.1 Previous Work

Cue integration is not a new issue. In [15] a two-cue integration algorithm is presented based on the use of constraints, in which optical flow is defined to be the constraining (i.e., most important) cue, and edges to be the secondary cue. This framework requires an a priori user-based definition of which cue is the most important one. A voting approach for disambiguation of cue information, along with a very thorough review and comparison of several methods, is proposed in [16]. In this paper, we describe a method for automated cue integration that is general enough to merge contributions of cues that are structurally very dissimilar. Unlike previous work, our approach avoids making a priori assumptions about the distribution of noise in cues, and it weights each cue's contribution dynamically depending on how much noise it contains.

There are several general statistical approaches designed for tracking, estimation, and prediction. The *Kalman filter* [17], for example, treats the parameters, as well as the observations, as multivariate Gaussians and also uses a linear predictive model. Another example, *Particle filter* [18, 19] techniques, which are also known in computer vision as *condensation* [20, 21], propagate the evolution of non-Gaussian sampled distributions through nonlinear operations. Unfortunately, the necessary number of samples of the distribution grows exponentially with the dimension of the parameter vector. Particle filters also require knowledge of the observations' distributions.

Our cue integration method cannot be directly compared with the previous examples, since we do not represent our parameters as random variables. Instead, we use the statistics of the cues only to combine them in an optimal way. Our algorithm does not assume any particular distribution in the observations. It just estimates their bounded support. Unlike condensation whose complexity grows exponentially with the dimension of the parameter space, the complexity of our approach is polynomial in the dimension of the parameter space.

2 Affine Arithmetic

Affine arithmetic is a numeric technique similar to *interval arithmetic,* in the sense that it propagates regions, instead of numbers, across arithmetic operations. The atom of affine arithmetic is called an *affine form.* An affine form $\hat{a}$ is represented as:

$$\hat{a} = a_0 + \sum_{i=1}^{m} a_i \varepsilon_i \tag{1}$$

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 $[-1 \ldots 1]$. The quantity a_0 is called the *central value* (mean), and the ε_i are called the *noise variables.* Each noise variable ε_i represents an independent component of the total uncertainty. In $\mathbb{R}^1$, $\hat{a}$ represents an interval and in $\mathbb{R}^n$ a convex polytope, whose number of faces depends on n and m.

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

$$\hat{z} = \alpha\hat{x} + \beta\hat{y} + \zeta, \tag{2}$$

are calculated exactly, where $\hat{x}$, $\hat{y}$, and $\hat{z}$ are affine forms represented by

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

α, β, and ζ are real constants. The definition of this operation is

$$z_0 = \alpha x_0 + \beta y_0 + \zeta \quad \text{and} \quad z_i = \alpha x_i + \beta y_i. \tag{3}$$

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 $a_0 = s$.

Although in this paper we only need the affine operation specified in Equation 2, other operations, including non-affine ones, are also possible. A thorough description of how to do operations like reciprocation, multiplication, exponentiations, trigonometry, or how to create a new operation, can be found in [22].

An affine form that is the result of an operation on other affine forms shares its noise variables with the affine forms of the operands. As a result, and in

contrast to interval arithmetic, affine forms preserve interdependencies between values from intermediate computations. After a series of cascading operations, affine arithmetic usually provides tighter bounds than interval arithmetic.

As an example, consider a two-dimensional affine form f_{cj} as follows:

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

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 variables 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.

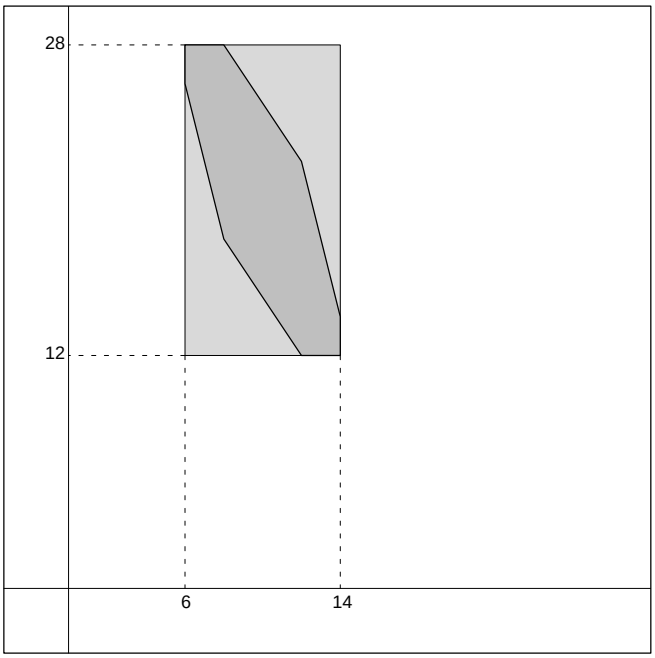


Fig. 1. Region defined by the two-dimensional affine form of Equation 4. In dark gray we see the region of the affine form, while in light gray is the region of the interval counterpart. Source: "Self-Validated Numerical Methods and Applications", Stolfi and Figueiredo, 1997 (used with permission).

3 Gaussians that Approximate Affine Forms

In this section we see how to connect Gaussian distributions to affine forms. We show how we can use a modified version of the *central limit theorem* to justify the approximation of an affine form with a Gaussian distribution, and how we can bound the error of the approximation.

We use affine forms to represent regions of uncertainty in a variable. From [22]: "At any *stable instant* in an AA computation, there is a single assignment of values from $U = [-1, 1]$ to each of the noise variables in use at that time that makes the value of every affine form equal to the value of the corresponding quantity in the ideal computation." In other words, the affine form represents the domain, or support, of the underlying random variable.

All noise variables are independent; thus the affine form is the sum of many independent random variables, whose support is a bounded one-dimensional segment embedded in $\mathbb{R}^n$. Each of the noise variables has an unknown probability distribution, so we cannot assume that they are identically distributed. Hence, we cannot apply the central limit theorem immediately. We can, however, use the multivariate version of *Lindeberg's theorem* [9, pp 262+]. It is an extension to the classical central limit theorem. In its one-dimensional form it tells us that for mutually independent one-dimensional random variables $X_1, X_2, \ldots$ with distributions $F_1, F_2, \ldots$ such that

$$E(X_k) = 0, \qquad \mathrm{Var}(X_k) = \sigma_k^2,$$

if the *Lindeberg condition* [9, pp 518+] is satisfied, the normalized sum

$$S_n^* = (X_1 + \cdots + X_n)/s_n,$$

where $s_n^2 = \sigma_1^2 + \cdots + \sigma_n^2$, tends to the normal distribution $\mathcal{R}$ with zero expectation and unit variance. Intuitively, the Lindeberg condition itself ensures that individual variances σ_k^2 are small if compared to their sum s_n^2 — no single random variable dominates the sum. This theorem can be generalized to multivariate distributions, as per [9, pp 262+].

We ensure that $E[\varepsilon_k] = 0$ by constructing the affine forms such that they are symmetric around the estimates of the local contributions in each cue; see Section 5 for further discussion. We ensure that the Lindeberg condition is satisfied by having enough local contributions with bounded uncertainties. Unfortunately, this theorem does not tell us how many noise variables are necessary in order for the Gaussian to be a good approximation. For estimating the error in the approximation, we need another theorem, the *Berry-Esseen theorem* [9, pp 544]:

Let the X_k be independent variables such that

$$E[X_k] = 0, \qquad E[X_k^2] = \sigma_k^2, \qquad E[|X_k^3|] = \rho_k,$$

and

$$s_n^2 = \sigma_1^2 + \cdots + \sigma_m^2, \qquad r_n = \rho_1 + \cdots + \rho_m.$$

Then

$$|F_m - \mathcal{R}| \leq 6\frac{r_n}{s_n^3}, \tag{5}$$

where F_m is the distribution of the normalized sum $(X_1 + \cdots + X_m)/s_n$, and $\mathcal{R}$is the normal distribution with zero mean and unit variance.

Since the support of each noise variable ε_k is $[-1, 1]$ and $E[\varepsilon_k] = 0$, the third moment $\rho_{\varepsilon_k} < 0.25$ exists. Hence, we can use the result of Equation 5 to provide an upper bound for the error along the principal axes of a Gaussian approximation of an affine form $\hat{a}$.

3.1 Estimation of the Gaussian Distribution

We have shown that the approximation of an affine form with a Gaussian distribution is justified, and that we can compute how closely the affine form represents a Gaussian distribution. We now show how to compute this approximation.

The Gaussian distribution that approximates $\hat{a}$ with

$$\tilde{\hat{a}} = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Lambda_c}|}} e^{-\frac{1}{2}(\mathbf{a}-\mu_c)^\top \mathbf{\Lambda}_c^{-1}(\mathbf{a}-\mu_c)} \tag{6}$$

is completely parameterized by a mean vector μ and a covariance matrix $\mathbf{\Lambda}$.

The mean vector is

$$\begin{aligned} \mu_c = E\left[\hat{a}\right] &= E[\mathbf{a}_0] + \sum_{i=1}^{m} E[\mathbf{a}_i \varepsilon_i] \\ &= \mathbf{a}_0 + \sum_{i=1}^{m} \mathbf{a}_i E[\varepsilon_i]. \end{aligned} \tag{7}$$

Since we ensure that $E[\varepsilon_i] = 0$,

$$\mu_c = \mathbf{a}_0. \tag{8}$$

The estimation of the covariance matrix Λ is not immediately obvious. Because affine forms represent convex polytopes, a geometric approach springs to mind. We presented such an algorithm in previous work [8], and now describe its main properties briefly.

3.2 Geometric Algorithm for the Gaussian Estimation

This algorithm first calculates the eigenvectors of Λ, and then it calculates the eigenvalues. It assumes that the principal axes of the Gaussian distribution are the same as the axes of the minimum-volume hyperparallelopiped that bounds the polytope. In order to find this hyperparallelopiped, it starts with an orthonormal basis of $\mathbb{R}^n$. Each step rotates two of the basis vectors in their plane and finds the minimum-area bounding rectangle of the affine form projected onto that plane. The rotation preserves the orthonormality of the basis and reduces

the total volume of the bounding hyperparallelopiped. This procedure eventually reaches a minimum, since each step never increases the volume. In practice, we apply the rotation once for every pair of vertices.

To find the eigenvalue associated with a given eigenvector $\mathbf{v}$ we project $\hat{a}$ onto $\mathbf{v}$, and obtain a one-dimensional affine form. The eigenvalue is

$$\sigma_v^2 = \sum_{j=1}^{m} (a_j \cdot \mathbf{v})^2 E[\varepsilon_j] = \sum_{j=1}^{m} (a_j \cdot \mathbf{v})^2 \sigma_{\varepsilon_j}^2. \tag{9}$$

This equation can be further simplified to

$$\sigma_v^2 = \sigma_\varepsilon^2 \sum_{j=1}^{m} (a_j \cdot \mathbf{v})^2 \tag{10}$$

if we assume that all noise variables have the same variance σ_ε^2. Note that using the same variance does not imply that the noise variables are identically distributed. Another option is to choose the eigenvalues such that a fixed percentage of the Gaussian is contained within the bounding hyperparallelopiped, by using $\mathcal{Q}$, the tabulated tails of Gaussian distributions.

This geometric algorithm runs in $\mathcal{O}(n^2 m)$ time and uses $\mathcal{O}(nm)$ space, where n is the dimension of the affine form, and m is the number of noise variables. This algorithm has three serious shortcomings: First, there is no guarantee that it will converge to the global minimum of the hyperparallelopiped's volume. Second, the assumption that the minimum-volume hyperparallelopiped is always aligned with the optimum principal axes of the Gaussian distribution is not valid. We show a counterexample for two dimensions in Figure 6. Third, the algorithm is complicated to implement.

We now present a novel, much simpler algorithm that also provides much better estimates of the principal axes of the Gaussian.

3.3 Expectation Algorithm for the Gaussian Estimation

Instead of interpreting the affine form geometrically, the new algorithm takes advantage of the expectation properties of the random variables. Using the definition of the covariance matrix Λ and Equation 8:

$$\Lambda_{\hat{a}} = E\left[(\hat{a} - a_0)(\hat{a} - a_0)^\top\right]. \tag{11}$$

Each element λ_{ij} of Λ is

$$\lambda_{ij} = E\left[(\hat{a} - a_0)_i (\hat{a} - a_0)_j\right] = E\left[\left(\sum_{k=1}^{m} a_{k\,i}\varepsilon_k\right)\left(\sum_{l=1}^{m} a_{l\,j}\varepsilon_l\right)\right],$$

where $a_{k\,i}$ is the ith component of the vector a_k in $\hat{a}$, and $(\hat{a} - a_0)_i$ is the one-dimensional affine form corresponding to the ith component of $(\hat{a} - a_0)$.

Expanding the sum we observe that, because the ε are mutually independent and have zero mean, the cross terms are zero:

$$\lambda_{ij} = \sum_{k=1}^{m} a_{k\,i} a_{k\,j} E\left[\varepsilon_k^2\right] = \sum_{k=1}^{m} a_{k\,i} a_{k\,j} \sigma_{\varepsilon_k}^2, \tag{12}$$

or, if assuming a common σ_ε^2 as in Equation 10,

$$\lambda_{ij} = \sigma_\varepsilon^2 \sum_{k=1}^{m} a_{k\,i} a_{k\,j}. \tag{13}$$

We build Λ using Equation 12 or 10. Note that both equations are just a multiplication of an n-by-m matrix with its transpose, where the $\sigma_{\varepsilon_k} a_k$ form the columns of the matrix.

With a standard implementation of a matrix multiplication, the expectation algorithm has the same complexity as the geometric algorithm, $\mathcal{O}(n^2 m)$, where n is the dimension of the affine form and m is the number of noise variables. Computing a single λ_{ij}, however, is much cheaper than the rotation of a pair of basis vectors in the geometric algorithm, so in practice, the expectation algorithm runs approximately 40 times faster. In addition, unlike in the geometric algorithm, there are no data dependencies in the computation of Λ, so it is fully parallelizable.

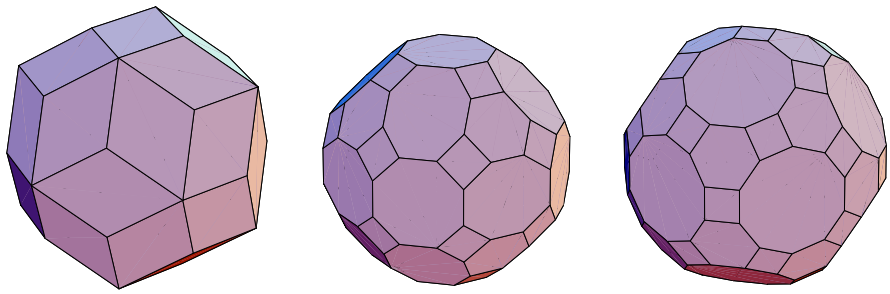
Beside the speed difference and simplicity of implementation, the expectation algorithm's most compelling advantage is that it provides an optimal estimate of the principal axes of the Gaussian distribution if the conditions of Lindeberg's theorem are satisfied. The reason is that if these conditions are satisfied, Lindeberg's theorem tells us that the affine form indeed represents a Gaussian probability distribution. Furthermore, the Gaussian estimated from Equations 12 or 10 has both the same first-order and second-order moments as the affine form. We show an example of this estimator's accuracy in Figure 6(b).

So far we have shown that the upper bound for converting an affine form to a Gaussian approximation is $\mathcal{O}(n^2 m)$. The question remains whether it is possible to use the geometric properties of affine forms to devise a better algorithm that improves this bound. We now discuss this question by connecting affine forms to zonotopes.

4 Zonotopes and Affine Forms

Zonotopes are a special type of convex polytopes obtained through the *Minkowsky sum* of line segments centered on the origin [11, 23]. Constructing the zonotope via the Minkowsky sum is equivalent to constructing the boundary of an affine form centered around the origin: Each component a_j from an affine form $\hat{a}$ (Equation 1) represents half the segment in the zonotope formulation, because $\varepsilon_j \in [-1, 1]$, so the full line segment goes from $-a_j$ to $+a_j$. The number of segments in the zonotope defines its *degree*, and is the same as the number of used noise variables in the affine form.

Zonotopes in three-dimensions are called zonohedra; some examples are shown in Figure 2.



(a) (b) (c)

Fig. 2. Three examples of Zonohedra (three dimensional zonotopes). 2(a) Rhombic Triacontahedron, 2(b) Truncated Small Rhombicuboctahedron, and 2(c) Truncated Icosidodecahedron. Source: "Zonohedra and Zonotopes", Eppstein, 1995 (used with permission).

We can provide bounds on the number of faces and points in a zonotope by noting that the points of the zonotope are the convex hull of all the points generated by the consecutive Minkowsky sum of the line segments. Each line segment adds two more points for each existing point, so m line segments yield 2^m points. Based on [11, p. 23], the number of points in the convex hull of the 2^m points is $\mathcal{O}(m^{n-1})$, immediately leading to $\mathcal{O}(m^{n-1})$ faces in a zonotope.

These bounds make it abundantly clear that all geometric algorithms that attempt to estimate the Gaussian distribution from the faces or boundary points of the affine form's region are doomed to fail. Although such algorithms would work well for two and three dimensions, the complexity explodes beyond these dimensions. In fact, such algorithms would have a complexity of at least $\mathcal{O}(m^{n-1})$, rendering them impractical for the dimensions we encounter in typical parameter vectors of deformable models. We conclude that any efficient algorithm for geometric processing of an affine form can only use the information in the vectors multiplying the noise variables, but not the information in the surface of the convex polytope represented by the affine form.

Zonotopes are used in several applications. In [14, 24] they are used for computing bounds of the orbits of dynamical systems. These papers introduce an interesting procedure (*cascade algorithm*) to reduce the degree of a zonotope. In [13] the zonotopes are explored in training *support vector machines*, and in [12] they are connected to the problem of finding the centroid of points with weights lying in a bounded interval.

5 Application: Cue Integration in Deformable Model Tracking

We now describe how to use affine forms and their approximations by Gaussian to integrate multiple cues in a deformable model tracking framework. The advantage of using affine arithmetic is that we avoid making assumptions about the exact probability distributions of the noise in each cue. Furthermore this approach enables us to weight the cues dynamically, depending on how reliable each one is, as opposed to choosing the most important cue a priori as in [15].

Each cue models local 2D image contributions as two-dimensional affine forms. The cue's generalized force is the sum of these local contributions, after we project them in the n-dimensional parameter space, using the Jacobian of the deformable model at each point. We approximate the generalized force, an n-dimensional affine form, with a Gaussian (section 3), and integrate all Gaussians using a *maximum likelihood estimator*. There is a more detailed explanation of some of these steps in [8].

We apply our cue integration technique in tracking, where, based on image observations, we recover the model's parameters as it evolves over time. This is not a normal inverse problem since the changes in the model between observations are small. We define the problem inductively. Using the correct model parameters of the previous observation, we recover the parameters that follow the model's evolution and match them to the new observation.

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

$$\dot{\mathbf{q}} = \mathbf{K}\mathbf{q} + f_g, \tag{14}$$

where $\mathbf{K}$ is a *stiffness matrix*, and f_g is a *generalized force*. We use numerical integration to solve this system, starting at the value of $\mathbf{q}$ at the end of the previous frame. The system converges to the closest point where the internal and external forces reach an equilibrium [25].

Different cues can be structurally different. Sometimes they come from distinct images or cameras, sometimes they affect disjoint sets of points. In the latter scenario, these different cues complement each other. For example, a point tracker cue works best in regions with complex texture, while a shape from shading cue works best in regions without texture. Cues can even come from three-dimensional data (like a range scanner). For these reasons, it is much better to integrate cues via the generalized forces, rather than via the image forces [8].

In our method, each cue c creates a generalized force $f_{g,c}$, through applying multiple image forces simultaneously at points on the model:

$$f_{g,c} = \sum_j \mathbf{B}_j^\top f_{cj}, \tag{15}$$

where $\mathbf{B}_j^\top$ is the projected model Jacobian at point j, and f_{cj} is the image force that cue c applies at point j.

When multiple cues interact, some collaborate, and some conflict. We need to combine them into an unified generalized force, and apply it to the dynamical system in Equation 14. We use two-dimensional affine forms to model the image forces, which describe in the image, how each force can vary. For example, an image force from an edge detector have more confidence along the gradient than along the tangential direction. Figure 3 illustrates an individual image force in an edge detector. Since $\mathbf{B}$ is a 2-by-n matrix, $\mathbf{B}^\top f_{cj}$ is just a set of affine operations over an affine form, so Equation 15 results in a n-dimensional affine form that represents the cue's generalized force.

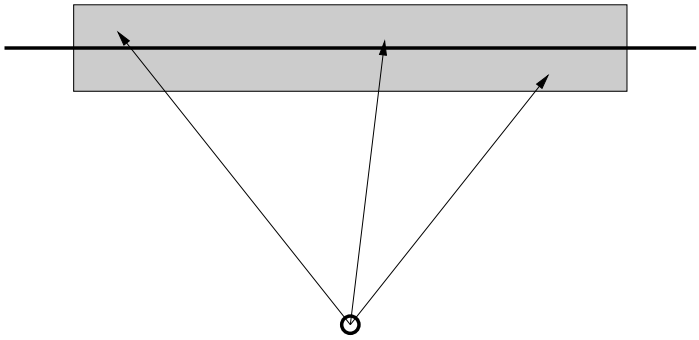


Fig. 3. Affine form for the image force in an edge detector. The region along the normal (gradient of the edge potential field) is smaller than the region along the edge, representing different confidences along these axes.

We assume independence between the image forces the different points in Equation 15. Thus we ensure that all noise variables in the affine form describing $f_{g,c}$ are independent. Using the techniques from section 3, we approximate $f_{g,c}$ with Gaussian. Thus, each cue provides a Gaussian probability density distribution of its generalized force.

We now have reduced the cue integration problem to Gaussian integration. This problem can be solved with a Gaussian *maximum likelihood estimator*. We use a static version of the *Kalman filter* [26] to solve it optimally. The Kalman filter estimates a new Gaussian distribution that optimally takes into account all the available information. We use the mean of this Gaussian as the generalized force f_g, and the covariance matrix as a measure of the estimate's robustness.

6 Validation and Experiments

We implemented the new expectation-based Gaussian estimation method in our deformable face tracking system, as described in [8]. We observed no degradation

of tracking. Some snapshots can be seen in Figures 4 and 5. There was an overall speedup of approximately 100 percent. The model we used in these sequences has 192 points, and there are 31 parameters to control its shape and motion.

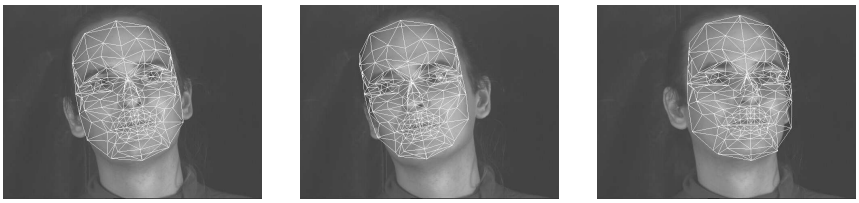


Fig. 4. Real images: Tracking of face rotation and translation with statistical methods

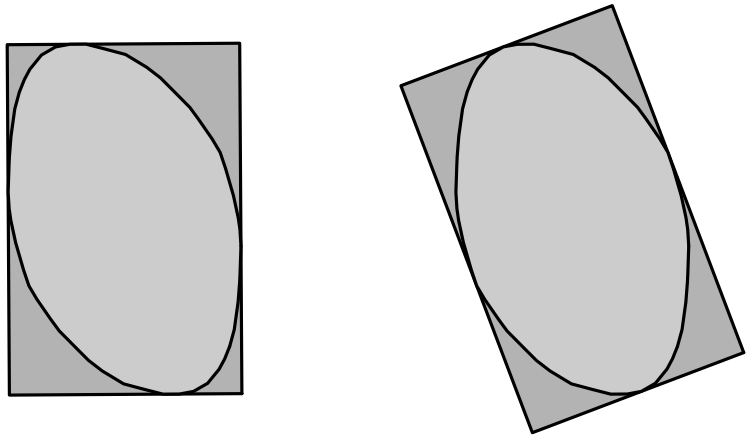
Fig. 5. Real images: Tracking of raising eyebrows with simultaneous head tilting with statistical methods

While inspecting the results, we compared the bounding boxes' volumes of the affine forms along the covariance matrices' axes. The geometric algorithm estimation consistently generated smaller volumes. Nevertheless, the minimal bounding box is not the best criterion to choose, because the volume is not necessarily minimal along the affine form's principal axes. The expectation method consistently estimates a match closer to the the desired orientation. In two-dimensions it is easy to visualize that he minimum volume bounding box may not correspond to the desired orientation axis: in Figures 6(a) and 6(b) we construct an affine form with 27 randomly generated noise variables centered around the origin. We can see that the minimum volume bounding box is not aligned along the principal components of the affine form.

7 Conclusions

In this paper, we studied the mathematics of affine arithmetic and its application to the problem of cue integration. We saw that affine forms, zonotopes, and Gaussian distributions are closely related, and explored this fact to develop a new algorithm to estimate a Gaussian from an affine form. Unlike condensation, this algorithm scales well with the dimension of the parameter space.

Within this framework, a cue must be able to recognize regions of confidence in the image space, and map them into affine forms. These image regions are



(a) (b)

Fig. 6. Bounding box of affine form along the axis of the estimated covariance matrices. In 6(a) we see the bounding box (with volume 253.697) along the axis of the Gaussian estimated by the geometric-based algorithm, in dark gray, against the affine form, in light gray. In 6(b) we see the bounding box (with volume 261.161) along the axis of the Gaussian estimated using our new expectation-based algorithm, in dark gray, against the same affine form, in light gray. Clearly, the fit in 6(b) is better, even though it does not minimize the volume.

converted into parameter regions, using affine arithmetic, and then summed up. The final cue contribution has a large number of noise variables, since each local image contribution has at least two noise variables. Hence, in conjunction with Lindeberg's Theorem, we can justify the assumption that the cue is well represented as a Gaussian distribution in parameter space. In addition, using Berry-Esseen's theorem, we have a way to estimate how good a given affine form's Gaussian approximation is.

Using the properties of zonotopes, we saw that any attempt to convert an affine form to a Gaussian using the geometric information on the boundary would not be computationally efficient.

We introduced a new expectation-based method for the Gaussian approximation that does not rely on any geometric information. Our new method directly constructs the covariance matrix of the affine form using expectation properties. Our previous geometric method obtained the set of axes that minimized the volume of the bounding box parallel to it. We showed that this criterion is not what we look for. Our new expectation algorithm has also superior computational efficiency. It is at least 40 times faster than the older method, and it is easier to implement and maintain. In addition, our expectation inspired method

is fully parallelizable, since there are no data dependencies in the calculation of every element of the covariance matrix.

References

1. Moore, R.: Interval Analysis. Prentice-Hall (1966)
2. Moore, R.: Methods and Applications of Interval Analysis. SIAM (1979)
3. Messine, F., Mahfoudi, A.: 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)
4. Kashiwagi, M.: An all solution algorithm using affine arithmetic. In: NOLTA'98 (1998 International Symposium on Nonlinear Theory and its Applications). (1998)
5. Egiziano, L., Femia, N., Spagnuolo, G.: New approaches to the true worst-case evaluation in circuit tolerance & sensitivity analysis: Part ii – calculation of the outer solution using affine arithmetic. In: COMPEL (6th Workshop on Computer in Power Electronics). (1998)
6. Figueiredo, L., Stolfi, J.: Adaptive enumeration of implicit surfaces with affine arithmetic. Computer Graphics Forum **15** (1996) 287–296
7. de Figueiredo, L.H.: Surface intersection using affine arithmetic. In: Graphics Interface. (1996) 168–175
8. Goldenstein, S., Vogler, C., Metaxas, D.: Affine arithmetic based estimation of cue distributions in deformable model tracking. In: Proceedings of CVPR. (2001) 1098–1105
9. Feller, W.: An Introduction to Probability Theory and Its Applications. Volume II. John Willey & Sons (1971)
10. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. McGraw-Hill (1991)
11. Edelsbrunner, H.: Algorithms in combinatorial geometry. Springer-Verlag (1987)
12. Bern, M.W., Eppstein, D., Guibas, L.J., Hershberger, J.E., Suri, S., Wolter, J.D.: The centroid of points with approximate weights. In: Proc. 3rd Eur. Symp. Algorithms. Number 979 in Lect. Notes in C. S., Springer-Verlag (1995) 460–472
13. Bern, M.W., Eppstein, D.: Optimization over zonotopes and training support vector machines. In Dehne, F., Sack, J.R., Tamassia, R., eds.: Proc. 7th Worksh. Algorithms and Data Structures. Number 2125 in Lect. Notes in C. S., Springer-Verlag (2001) 111–121
14. Kuhn, W.: Rigorously computed orbits of dynamical systems without the wrapping effect. Computing (1998) 47–67
15. de Carlo, D., Metaxas, D.: Optical flow constraints on deformable models with applications to face tracking. IJCV **38** (2000) 99–127
16. Brutigam, C.G.: A model-free voting approach to cue integration. PhD thesis, Dept. of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology) (1998)
17. Maybeck, P.: Stochastic Models, Estimation, and Control. Academic Press (1979)
18. Smith, A., Gelfand, A.: Bayesian statistics without tears: A sampling-resampling perspective. American Statistician **46** (1992) 84–88
19. Gordon, N.J., Salmon, D.J., Smith, A.F.M.: A novel approach to nonlinear/nongaussian bayesian state estimation. IEEE Proceedings on Radar Signal Processing (1993) 107–113

20. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Proceedings of ECCV. (1996) 343–356
21. Isard, M., Blake, A.: CONDENSATION: conditional density propagation for visual tracking. International Journal of Computer Vision **29** (1998) 5–28
22. Stolfi, J., Figueiredo, L.: Self-Validated Numerical Methods and Applications. 21° Colóquio Brasileiro de Matemática, IMPA (1997)
23. Eppstein, D.: Zonohedra and zonotopes. Technical Report 95-53, Dept. of Information & Computer Science, U. C. Irvine (1995)
24. Kuhn, W.: 125–134. In: Zonotope Dynamics in Numerical Quality Control. Springer (1998)
25. Metaxas, D.: Physics-based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging. Kluwer Academic Publishers (1996)
26. Plessis, R.: Poor man's explanation of Kalman filtering or how I stopped worrying and learned to love matrix inversion. Technical report (1967)