International Journal of Computer Vision, 11:2, 127–145 (1993)

A Framework for Spatiotemporal Control in the Tracking of Visual Contours

ANDREW BLAKE, RUPERT CURWEN, AND ANDREW ZISSERMAN
Department of Engineering Science, University of Oxford, Parks Rd, Oxford OX1 3PJ, UK

Received November 4, 1992; Revised May 21, 1993.

Abstract

There has been a great deal of research interest in contour tracking over the last five years. This article combines themes from tracking theory—elastic models and stochastic filtering—with the notion of affine invariance to synthesize a substantially new and demonstrably effective framework for contour tracking.

A mechanism is developed for incorporating a shape template into a contour tracker via an affine invariant coupling. In that way the tracker becomes selective for shape and therefore able to ignore background clutter. Affine invariance ensures that the effect of varying viewpoint is accommodated. Use of a standard statistical filtering framework allows uncertainties to be treated systematically, which accommodates object flexibility and unmodeled distortions such as the deformation of a silhouette under motion.

The statistical framework also facilitates a further development. In place of heuristically determined spatial scale for feature search, both spatial scale and temporal memory are controlled automatically and in a way that is responsive to the tracking process. Typically, the tracker operates initially in a coarse scale/short memory mode while it searches for a feature. Then spatial scale diminishes to allow more precise localization while memory (temporal scale) lengths to take advantage of motion coherence. All system parameters are determined by natural assumptions and desired tracking performance, leaving none to be fixed heuristically.

Versions of the tracker have been implemented at video rate, both on SUN 4 and in parallel, using a network of 11 transputers. The theoretically established properties of automatic control of spatiotemporal scale and of affine invariance are demonstrated using the implemented tracker.

1 Introduction

This article is concerned with the principles of tracking curves in motion, at video rate. This has many potential applications, for instance in biomedical image analysis—for example, (Ayache et al. 1992)—in surveillance—for example, (Sullivan 1992)—and in autonomous vehicle navigation—for example, (Dickmanns & Graefe 1988). Earlier versions of our tracker have been used in the control of a robot arm, supporting closed-loop tracking (Curwen & Blake 1992) and various aspects of hand-eye coordination (Cipolla & Yamamoto 1990; Blake et al. 1991; Blake et al. 1992; Cipolla & Blake 1992; Blake, 1992). A review of existing methods in curve tracking can be found in (Blake & Yuille 1992). Our aim here is to set out a framework for contour tracking as a relatively autonomous process, in which tracking behavior is determined as a mathematical consequence of some natural assumptions about geometry and uncertainty.

The framework has evolved from the principles of the snake of Kass et al. (1987), which is an elastic model for shapes in motion that can be coupled to image features. The elastic framework has been shown to be approximately equivalent to a Kalman filter that can be derived from statistical assumptions about curves and their motion (Szeliski & Terzopoulos 1991). In fact such an elastic system can be equivalent to the steady state of a Kalman filter. Here, the Kalman filter formalism will be developed further.

It is efficient to represent curves parametrically with a low-dimensional basis rather than using a pixel-based

representation, or a fine polygonal chain. Such a basis has been used for tracking solid models (Terzopoulos & Metaxas 1991) and for image curves using B-splines (Menet et al. 1990; Cipolla & Yamamoto 1990) and other parameterizations (Scott 1987). The B-spline representation is used in this article, though most of the results apply to any reasonable curve basis.

Another theme that is related to the snakes idea has been the representation of geometric prior information which can be incorporated into the tracker by means of a template. Templates—parameterized shapes—have been used effectively in nondynamic shape-fitting processes (Fischler & Elschlager 1973; Grenander et al. 1991; Lipson et al. 1990; Bennett & Craw 1991; Yuille et al. 1992). Some include statistical learning of shape variations (Grenander et al. 1991; Bennett & Craw 1991). Bookstein (1988) derives one method of elastic matching, with thin-plate splines, that allows affine transformations freely while allowing other deformations with some "reluctance."

On a practical note it has been shown recently that curve trackers can run at video rate without special hardware. Although it was originally thought that convolution hardware was needed for low-level image processing to support tracking, this has proved not to be the case. It has been demonstrated (Inoue & Mizoguchi 1985; Thompson and Mundy 1987; Dickmanns & Graefe 1988; Harris 1992; Curwen & Blake 1992; Lowe 1992; Wang & Brady 1992) by several researchers that tracking of rigid or deforming bodies is possible, often at frame rate and with modest hardware.

Several significant advances are reported here. First, templates are coupled into the dynamics of a real-time tracker with allowance for spatiotemporal uncertainty—both elastic deformability and temporal noise. Second, the statistical basis of the tracker is used to control spatiotemporal scale automatically and in a way that fits the progress of the tracking task. This is a considerable advance on previous approaches in which the spatial scale for feature search was set by hand. Mathematical analysis elucidates the operation of the spatiotemporal scaling mechanism and establishes that the tracker behaves stably. Third, the coupling of template to tracker is made invariant to affine transformations of the template. This allows both for 3-D rigid motion of a planar shape and for uncertainty in camera calibration. Fourth, the template mechanism is extended, representing the template by a *subspace* of the tracker's state-space. It is shown that this mechanism subsumes the affine-invariant 2-D template and generalizes it to full 3-D rigid motion of a nonplanar shape. Variations on the structure of the subspace allow for constrained tracking—for example panning—and surprisingly for exploring, simultaneously and dynamically, more than one object-hypothesis.

2 State Space Representation

The tracking model assumes that the moving feature is a contour $(X(s, t), Y(s, t))$ which can be expressed parametrically in terms of time-varying control points $(X_n(t), Y_n(t))$, $n = 1, \ldots, M$. State vectors are defined in terms of $\mathbf{X}$, $\mathbf{Y}$ where $\mathbf{X} = (X_1, \ldots, X_M)^T$ and similarly for $\mathbf{Y}$. (Note that the notation will omit explicit reference to s, t where appropriate.) The aim of this article is to develop a tracker which is an estimate $(\hat{X}(s, t), \hat{Y}(s, t))$, expressed in terms of estimated state vectors $\hat{\mathbf{X}}(t)$, $\hat{\mathbf{Y}}(t)$. The estimate is updated continually by reference to a visual feature $(X_f(s, t), Y_f(s, t))$ which is measured by searching in the vicinity of $(\hat{X}(s, t), \hat{Y}(s, t))$ as in figure 1.

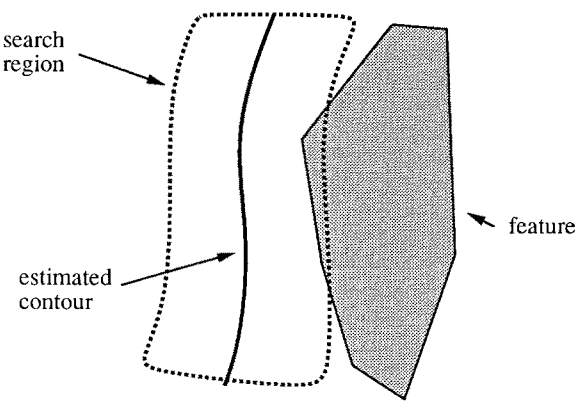

Fig. 1. The basic tracker is an estimated contour updated continuously using features that fall within its search region.

2.1 B-spline

The mathematical framework of this presentation largely applies to any parametric representation for curves. Specifically we will refer to the parametric B-spline representation, of which quadratic and cubic are particularly useful. A B-spline curve $(X(s), Y(s))$ of degree d is defined parametrically for $0 \leq s \leq N$, where $M = N$ for closed curves and $M = N + d$ for open ones (with appropriate variations where multiple knots are used to vary curve continuity):

$$X(s) = H(s)\mathbf{X} \tag{1}$$

where

$$H(s + n) = \mathbf{s}^T B_{n+1} G_{n+1}$$
$$0 \leq s \leq 1,\ 0 \leq n < N \tag{2}$$

$\mathbf{s}^T = (1, s, \ldots, s^d)$, B_n is a standard B-spline matrix (Faux & Pratt 1979; Bartels et al. 1987) and G_n *is a* $d \times M$ matrix that simply selects d consecutive control points:

$$G_n\mathbf{X} = (X_n, \ldots, X_{n+d})^T \quad 1 \leq n \leq N$$

Note that, for a closed curve, control point indexes are evaluated modulo M. The definition for $Y(s)$ is similar.

2.2 *State Space Metric*

Uncertainty in state space will be treated in terms of "Mahalanobis distance" (Rao 1973) via a norm $\| \ldots \|$ on $\mathbf{X}$ or $\mathbf{Y}$ which is compatible with true Euclidean distance measure in the image plane. We therefore define the norm so that

$$\|\mathbf{X}\|^2 = \int_{s=0}^{N} X(s)^2 \, ds$$

or, equivalently,

$$\|\mathbf{X}\|^2 = \mathbf{X}^T \underline{\mathcal{H}} \mathbf{X} \tag{3}$$

where the "metric" matrix $\underline{\mathcal{H}}$ is

$$\underline{\mathcal{H}} = \int_0^N H(s)^T H(s) \, ds \tag{4}$$

which, from (2), is

$$\underline{\mathcal{H}} = \sum_{n=1}^{N} G_n^T B_n^T \underline{\mathcal{S}} B_n G_n \tag{5}$$

a form that is convenient for practical computation of $\underline{\mathcal{H}}$, and where $\underline{\mathcal{S}}$ is the invertible, banded matrix

$$\underline{\mathcal{S}} = \int_0^1 \mathbf{s}\mathbf{s}^T \, ds \tag{6}$$

For instance, for quadratic B-splines,

$$\underline{\mathcal{S}} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}.$$

The $\underline{\mathcal{H}}$-matrixes are sparse, becoming more so as the number N of spans increases. For closed contours the $\underline{\mathcal{H}}$-matrix is a sparse circulant, for instance, for a quadratic spline with $N = 8$,

$$\underline{\mathcal{H}} = \begin{pmatrix} 0.55 & 0.217 & 0.008 & 0 & 0 & 0 & 0.008 & 0.217 \\ 0.217 & 0.55 & 0.217 & 0.008 & 0 & 0 & 0 & 0.008 \\ 0.008 & 0.217 & 0.55 & 0.217 & 0.008 & 0 & 0 & 0 \\ 0 & 0.008 & 0.217 & 0.55 & 0.217 & 0.008 & 0 & 0 \\ 0 & 0 & 0.008 & 0.217 & 0.55 & 0.217 & 0.008 & 0 \\ 0 & 0 & 0 & 0.008 & 0.217 & 0.55 & 0.217 & 0.008 \\ 0.008 & 0 & 0 & 0 & 0.008 & 0.217 & 0.55 & 0.217 \\ 0.217 & 0.008 & 0 & 0 & 0 & 0.008 & 0.217 & 0.55 \end{pmatrix} \tag{7}$$

For an open curve, $\underline{\mathcal{H}}$ is banded, pentadiagonal for quadratic splines, as shown here for the case $N = 4$:

$$\underline{\mathcal{H}} = \begin{pmatrix} 0.2 & 0.117 & 0.017 & 0 & 0 & 0 \\ 0.117 & 0.333 & 0.208 & 0.008 & 0 & 0 \\ 0.017 & 0.208 & 0.55 & 0.217 & 0.008 & 0 \\ 0 & 0.008 & 0.217 & 0.55 & 0.208 & 0.017 \\ 0 & 0 & 0.008 & 0.208 & 0.333 & 0.117 \\ 0 & 0 & 0 & 0.017 & 0.117 & 0.2 \end{pmatrix} \tag{8}$$

and heptadiagonal for cubic splines.

Lastly, given the norm $\| \ldots \|$ it is natural also to define a compatible inner product $< \ldots >$ such that:

$$< \mathbf{X}, \mathbf{X}' > = \mathbf{X}^T \underline{\mathcal{H}} \mathbf{X}' \tag{9}$$

3 Visual Features

The feature $[X_f(s, t), Y_f(s, t)]$ is defined by searching along fixed lines radiating from the current estimate $[\hat{X}(s, t), \hat{Y}(s, t)]$. The lines may be unit normal vectors to the curve or, if a constrained tracker is required, along fixed parallel lines. Search occurs on a specified spatial scale (figure 1) defined by a search window, selecting a point of maximum contrast or other visual measure, as appropriate for the underlying object model. In our implementations, contrast is used. In

practice, of course, $[X_f(s, t), Y_f(s, t)]$ is not observed in its entirety but at sampled points s_i along the contour. Furthermore, in the interests of computational speed, contrast is examined at only three different points on each normal: on the curve and at the two extremes of an interval which is initially close to the full width of the search window. If one of the extremes has the highest contrast it is retained as the current $[(X_f(s, t), Y_f(s, t)]$. Otherwise the interval is halved in length and the process is repeated.

If the sampling is dense and uniform in s, it is a reasonable abstraction to model the measurement continuously and this will allow sufficient analysis to give some insight into the operation of our estimator as a control system. Similarly, although measurements are made at discrete times, they can usefully be regarded as continuous-time for the purpose of analysis. Then, assuming that sensor error is unbiased, homogeneous, isotropic, and Gaussian, we have the following conditional p.d.f. (probability density function) for the measurement process:

$$p\{[X_f(s), Y_f(s)]|[X(s), Y(s)]\}$$

$$\propto \exp -\frac{1}{2r}\int_0^N \{[X(s) - X_f(s)]^2$$

$$+ [Y(s) - Y_f(s)]^2\}\, ds \qquad (10)$$

where r is a measurement variance constant (strictly, variance spectral density—see (Gelb 1974)). As before, $[X(s), Y(s)]$ is the *true* underlying position of the curve, as distinct from the *estimated* position $[\hat{X}(s), \hat{Y}(s)]$.

Now the square error integral above can be reexpressed, using the fact that $X(s) = H(s)\mathbf{X}$, and completing the square:

$$\int_0^N (X(s) - X_f(s))^2\, ds$$

$$= \|\mathbf{X} - \mathbf{X}_f\|^2 - \|\mathbf{X}_f\|^2 + \int X_f(s)^2\, ds \qquad (11)$$

where $\mathbf{X}_f$ is the least-squares B-spline approximation to the feature,

$$\mathbf{X}_f = \underline{\mathcal{H}}^{-1}\int_0^N H(s)^T X_f(s)\, ds \qquad (12)$$

and similarly for Y terms. Now since the only term on the right of (11) that depends on $\mathbf{X}$ is $\|\mathbf{X} - \mathbf{X}_f\|^2$, the other terms being effectively constant, the conditional p.d.f. (10) can be expressed directly in terms of Mahalanobis distances:

$$p\{[X_f(s), Y_f(s)]|[X(s), Y(s)]\}$$

$$\propto \exp -\frac{1}{2r}(\|\mathbf{X}_f - \mathbf{X}\|^2 + \|\mathbf{Y}_f - \mathbf{Y}\|^2) \qquad (13)$$

and this depends on the feature $(X_f(s), Y_f(s))$ only via its B-spline approximation $(\mathbf{X}_f, \mathbf{Y}_f)$.

We can now regard a feature as a time-varying measurement $(\mathbf{X}_f, \mathbf{Y}_f)$ in the joint state-space for the $\mathbf{X}$, $\mathbf{Y}$ processes. From (13) and (3), each of $\mathbf{X}_f$ and $\mathbf{Y}_f$ has covariance matrix

$$R = r\underline{\mathcal{H}}^{-1} \qquad (14)$$

This abstraction of the sensor will be used for now to obtain analytic insights into performance. Later, we will outline modifications that are required for real discrete measurements and due to the "aperture problem" (Horn 1986) which allows only the normal component of displacement of $[X_f(s), Y_f(s)]$ to be measured.

4 Stochastic Dynamical Model

A simple dynamical model is based on the assumption of uniform 2-D motion with an additive Gaussian noise process representing randomly varying force applied continuously over time. In an augmented state space of vectors $(\mathbf{X}, \dot{\mathbf{X}})^T$ this is expressed as a stochastic differential equation (Gelb 1974)

$$\frac{d}{dt}\begin{pmatrix}\mathbf{X}\\ \dot{\mathbf{X}}\end{pmatrix} = \begin{pmatrix}\dot{\mathbf{X}}\\ \mathbf{0}\end{pmatrix} + \begin{pmatrix}\mathbf{0}\\ \mathbf{w}\end{pmatrix} \qquad (15)$$

where $\mathbf{w}(t)$ is a zero mean, temporally uncorrelated Gaussian noise process. A similar equation applies for $\mathbf{Y}$, independently of the $\mathbf{X}$ process provided the noise process is assumed to be isotropic. Assuming an isotropic and homogeneous Gaussian distribution, and following similar reasoning to that used above for the measurement process, the covariance spectral density matrix for $\mathbf{w}$ is simply $q\underline{\mathcal{H}}^{-1}$ where q is a variance (spectral density) constant.

4.1 Tracking Filter

Under this simple model, one can build a standard continuous Kalman filter for the estimated contour $(\hat{\mathbf{X}}, \hat{\mathbf{Y}})$ of the form

$$\frac{d}{dt}\begin{pmatrix}\hat{\mathbf{X}}\\ \dot{\hat{\mathbf{X}}}\end{pmatrix} = \begin{pmatrix}\dot{\hat{\mathbf{X}}}\\ \mathbf{0}\end{pmatrix} + K(\mathbf{X}_f - \hat{\mathbf{X}}) \qquad (16)$$

where $K(t)$ is the Kalman gain matrix, defined in terms of the covariance of the measurement process and the covariance $P(t) = E[(\mathbf{X} - \hat{\mathbf{X}})(\mathbf{X} - \hat{\mathbf{X}})^T]$ of the current state estimate:

$$K = P H^T R^{-1} \tag{17}$$

where H is a measurement matrix:

$$H = (\mathbf{I} \quad \mathbf{0}) \tag{18}$$

representing the fact that the contour position $\mathbf{X}$ is measured, but not the velocity $\dot{\mathbf{X}}$.

4.2 Shape Template

It remains to specify initial conditions for the filter. This is done by initializing the estimator to some fixed-shape template which is defined to be a B-spline expressed as control point vectors $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. Assuming, similarly to the measurement model, a spatially homogeneous and isotropic Gaussian prior distribution for the state $\mathbf{X}$, $\mathbf{Y}$ then, by similar reasoning, the prior distribution in the augmented state-space has covariance

$$P(0) = \begin{pmatrix} \alpha_1 \underline{\mathcal{H}}^{-1} & 0 \\ 0 & \alpha_2 \underline{\mathcal{H}}^{-1} \end{pmatrix}$$

which serves as an initial condition for the Riccati equation (Gelb 1974) that specifies the evolution of $P(t)$ in a Kalman filter:

$$\frac{dP}{dt} = FP + PF^T + Q - K R K^T \tag{19}$$

where, for constant-velocity dynamics, the matrix F is

$$F = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix} \tag{20}$$

and I, 0 are the $M \times M$ identity and zero matrices, and where

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & q \underline{\mathcal{H}}^{-1} \end{pmatrix} \tag{21}$$

is the plant-noise covariance spectral-density matrix.

Now substituting a solution for $P(t)$ of the form

$$P(t) = \begin{pmatrix} p_{11} \underline{\mathcal{H}}^{-1} & p_{12} \underline{\mathcal{H}}^{-1} \\ p_{21} \underline{\mathcal{H}}^{-1} & p_{22} \underline{\mathcal{H}}^{-1} \end{pmatrix} \tag{22}$$

into (19) and using (17) and the definitions (14) and (18) gives

$$\frac{d}{dt} \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 2p_{21} - \frac{p_{11}^2}{r} & p_{22} - \frac{p_{11}p_{12}}{r} \\ p_{22} - \frac{p_{11}p_{12}}{r} & q - \frac{p_{12}^2}{r} \end{pmatrix} \tag{23}$$

with initial conditions

$$p_{11}(0) = \alpha_1, \quad p_{22}(0) = \alpha_2, \quad p_{12}(0) = p_{21}(0) = 0$$

The Kalman gain in (17) is simply (the $\underline{\mathcal{H}}$ terms cancelling):

$$K = \begin{pmatrix} k_1 \mathrm{I} \\ k_2 I \end{pmatrix} \quad \text{where} \quad \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} p_{11} \\ p_{21} \end{pmatrix} \tag{24}$$

4.3 Tracking Performance

The diagonal form of the Kalman gain means that the second-order dynamics in state space (16) degenerate into a set of identical, independent second-order systems on each control point. This simple case arises because of the homogeneity of the measurement model. Nontrivial modal structure is created when measurements are inhomogeneous, for instance when part of the contour fails to lock onto a visual feature. However, the homogeneous case is useful for analysis as a guide to the stability and accuracy properties of the tracker. In the steady state, for instance, from (24) and taking $d/dT \equiv 0$ in (23),

$$\begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} \sqrt{2}(q/r)^{1/4} \\ (q/r)^{1/2} \end{pmatrix} \tag{25}$$

Now, in the steady state, the second-order systems have damping constant $\beta = k_1/2$ and natural frequency $\omega = \sqrt{k_2}$ so that, from (25), $\omega = \sqrt{2}\beta$, just underdamped relative to the critical damping condition $\beta = \omega$.

This means that tracking behavior is good—fast but stable—regardless of the setting of the covariance parameters q, r. Furthermore, tracking is also accurate in the steady state. In fact tracking error is zero, in the steady state, for a constant-velocity target, since

$$\hat{\mathbf{X}} = \hat{\mathbf{X}}_f = \mathbf{V}t \quad \text{with} \quad \mathbf{V} \text{ constant}$$

is a solution of (16).

5 Automatic Control of Spatiotemporal Scale

Automatic control of spatial scale for search is achieved by applying a *validation gate* (Bar-Shalom & Fortmann

1988) in a spatially distributed fashion. In the isotropic case (isotropy is broken if measurements comprise only the normal component of displacement—see later), the filters for the **X** and **Y** processes are identical. A circular feature-search window (elliptical in the anisotropic case) of radius $2\rho(s, t)$ is constructed around each point on the curve where ρ^2 is the positional variance of the current estimator at s:

$$\rho(s)^2 = H(s)\, P\, [H(s)]^T \tag{26}$$

Feature search is then performed along the normal to the estimated contour, and within the window. The circle $\rho(s, t)$ can be pictured as sweeping along the estimated contour to form an enclosing search region as in figure 2.

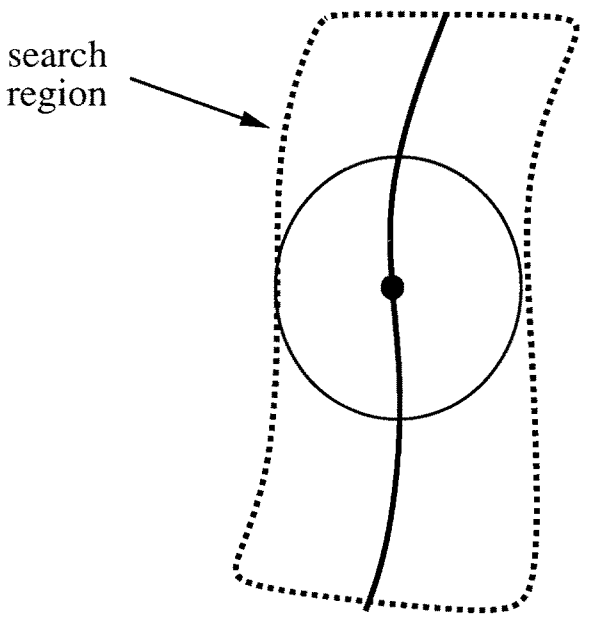

Fig. 2. The search region is formed by sweeping an ellipse of uncertainty along the estimated contour.

When P is also homogeneous, as in the previous section, its Riccati equation can be solved analytically to give some insight into the operation of the scale-control mechanism. The radius varies as $\rho(s)^2 \propto p_{11}(t)$ and $p_{11}(t)$ varies according to the differential equation (23) when a feature is present. In fact it is also true that $\rho(s) \approx \sqrt{p_{11}}$, so the variation of ρ over time is governed by the Riccati equation (23).

In the absence of a feature, when lock is lost (and assuming the whole contour is unlocked), no measurements are made. This can be incorporated into the Riccati equation (23) by regarding the measurement covariance as infinite, and as $r \to \infty$, (23) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 2p_{21} & p_{22} \\ p_{22} & q \end{pmatrix} \tag{27}$$

which can be solved exactly to give

$$p_{11}(t) = p_{11}(0) + p_{12}(0)t + p_{22}(0)t^2 + \frac{1}{3}qt^3$$

so that asymptotically the search scale ρ grows as $t^{3/2}$. If at some time t, assumed large so that (23) can be solved approximately, the whole feature locks on again, the search scale will contract again approximately as $\rho \propto 1/\sqrt{t}$ until, asymptotically, it reaches the steady state scale of $\rho_\infty = 2^{1/4}(qr^3)^{1/8}$. This behavior is illustrated for a tracking contour with simulated data in figure 3 and with real data in figure 4.

5.1 Automatic Control of Temporal Scale

The Riccati mechanism also takes care automatically of temporal scale, the effective memory of the tracker. The Kalman gain K has the dimensions of inverse time and, in a multivariate system, governs the duration of the negative exponential memory of each mode of the tracker. In the simple case we are using for analysis, modes are degenerate and all have time constant $\tau = 1/\beta$, where β is the damping constant defined above. Taking τ as the characteristic time of the Kalman filter's negative exponential memory and, using the definitions of β, k_1 above,

$$\tau = \frac{2r}{p_{11}} \approx \frac{2r}{\rho^2}$$

so that temporal scale varies as the inverse square of spatial scale. In the absence of a feature, temporal scale shortens so that feature acquisition and locking can occur rapidly. Once locked, temporal scale lengthens, allowing motion coherence to be exploited.

In the practical and general case that the contour is only partially locked onto a feature, spatial scale will be inhomogeneous, being larger where the contour is not locked. Consequently, temporal scale should also be inhomogeneous. The contour should react more rapidly over segments with greater spatial scale and this is illustrated in figure 5.

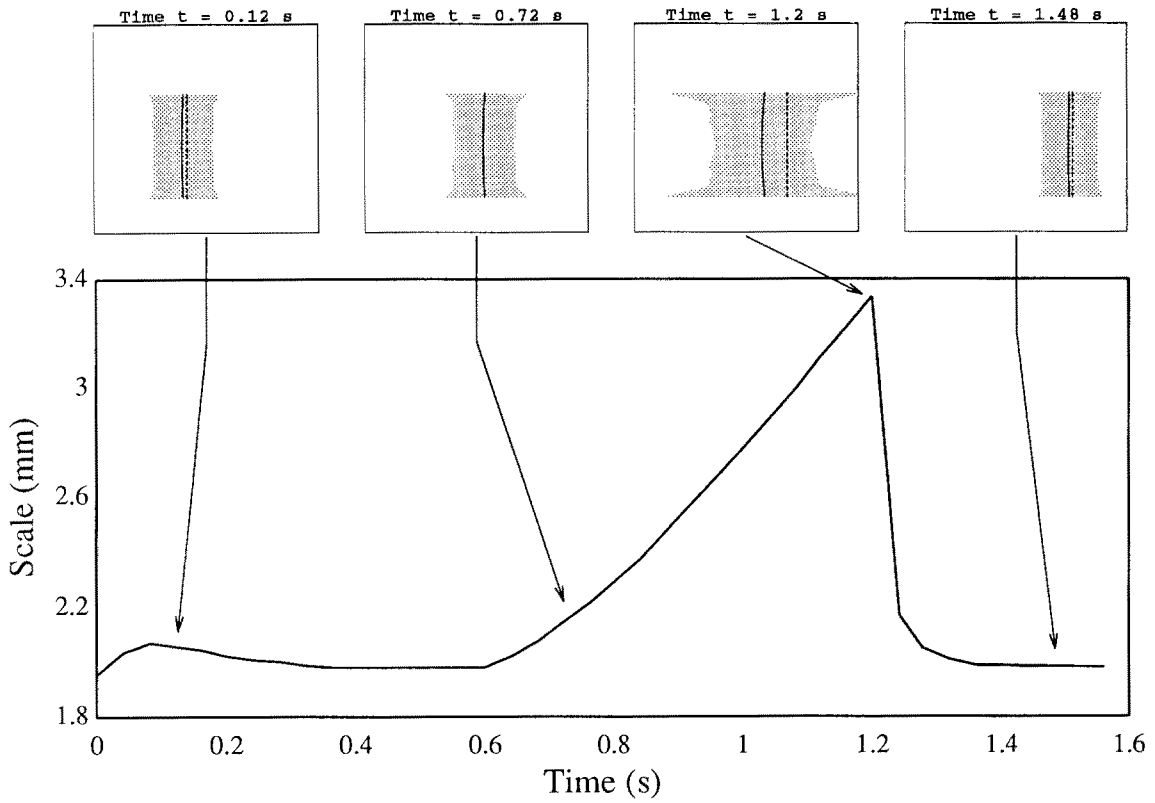

Fig. 3. The time-course of spatial scale is illustrated here by applying the tracker to simulated data. Spatial scale decays to a steady-state value as tracking proceeds. When the feature is lost, at $t = 0.6$ s, spatial scale increases and theory predicts a $t^{3/2}$ growth. Then the feature is recaptured at $t = 1.2$ s and spatial scale decreases again (theoretically inversely with $\sqrt{t}$), once again reaching a steady state. Tracking parameters for this simulation were $r = 1.0$ mm^2.s and $q = 66$ mm^2.s^{-3}, giving a steady state scale of $\rho_\infty = 2.05$ mm, in a total field of view of width 8 mm on the image plane.

5.2 *Setting of System Parameters*

The assumptions of homogeneity and isotropy in system and measurement uncertainties dramatically reduced the numer of unspecified covariance parameters from $O(M^2)$ to just a few, because all covariance submatrixes turned out to be multiples of $\mathcal{H}^{-1}$, the inverse of the metric matrix. In fact there are just four covariance parameters to specify: r, q, α_1, α_2. Of these, α_1, α_2 will prove to be less important; they govern the initial strength of influence of the template. Later, however, the "persistent template" is introduced, which has a continuing influence, not just an initial one, governed by an additional parameter $\bar{r}$ (see Section 7).

Measurement covariance r is fixed, in principle, by the sensor characteristics, and might reflect a typical noise variance of a fraction of one (pixel)^2 in image measurement. In practice, this is unrealistic. The measurement process outlined earlier, searching along normals from the current estimated curve, is crude in the interests of speed and its error is of the order of the width of the search window, which may be much greater than one pixel. Ideally, then, we should set $r \propto \rho^2$ so that r is time-varying. This is an attractive idea, but it remains to solve the covariance equation (no longer a Riccati equation since R is a function of P) for this case. In the meantime the simpler case is considered in which r is constant.

Rather than setting the four parameters explicitly, it is more natural to fix an equivalent set of four which correspond directly to operational characteristics of the tracker. Initial and steady-state values of the spatial scale ρ for search are set by q, α_1 as follows:

$$\rho_0 = \sqrt{\alpha_1} \text{ and } \rho_\infty = 2^{1/4}(qr^3)^{1/8} \tag{28}$$

The variation of temporal scale with spatial scale

$$\tau = 2\frac{r}{\rho^2} \tag{29}$$

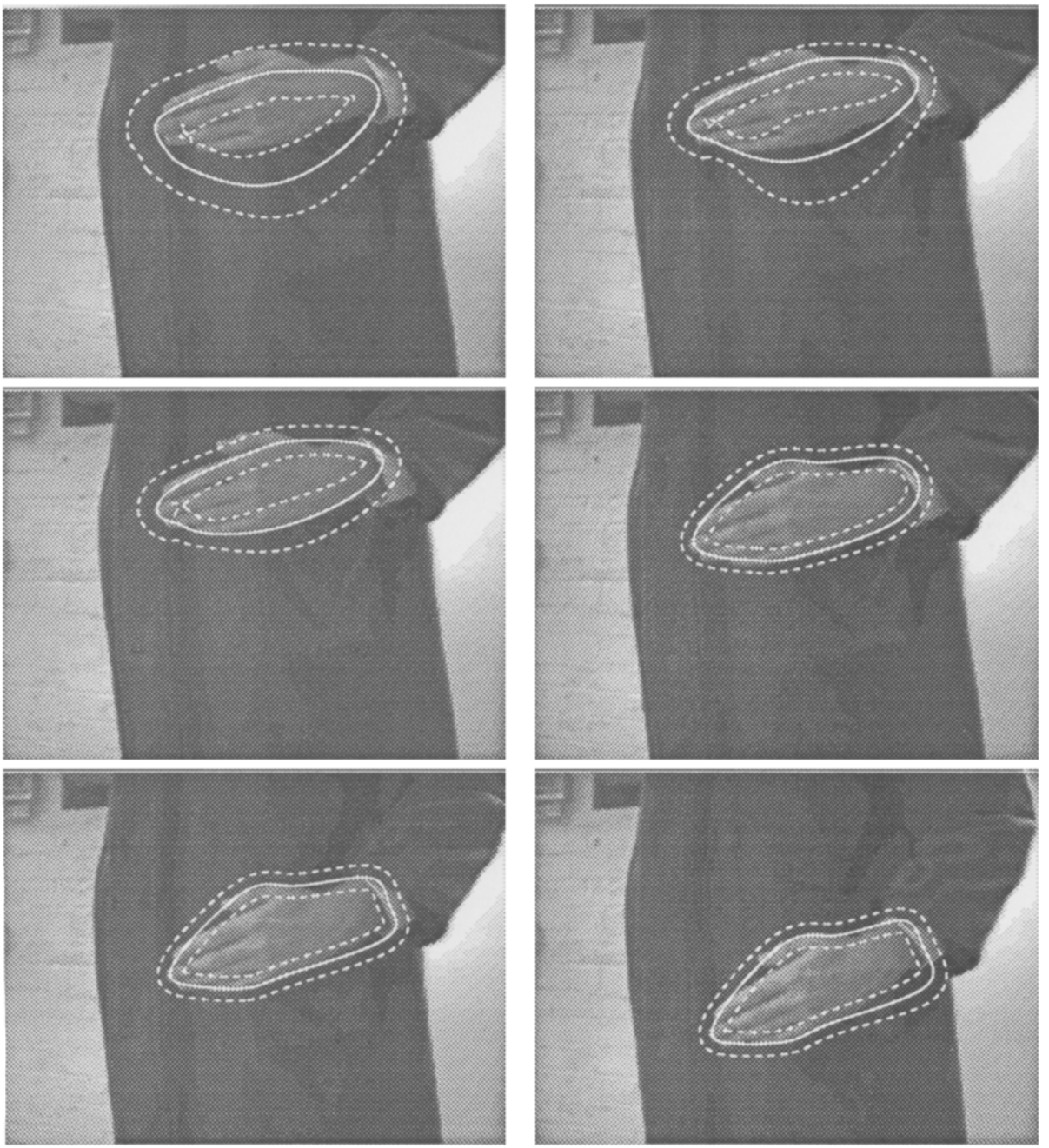

Fig. 4. Frames are shown here at times $t = 0, 0.04, 0.08, 0.32, 0.56, 0.8$ seconds (raster order) in the tracking of a moving hand. The solid white line shows the estimated contour and the dotted line is the boundary of the search region. Initially, the lower edge of the hand is not locked on; the search region expands locally until the contour does lock on. Over time the search region reaches a steady state and the continuing motion of the hand is successfully tracked.

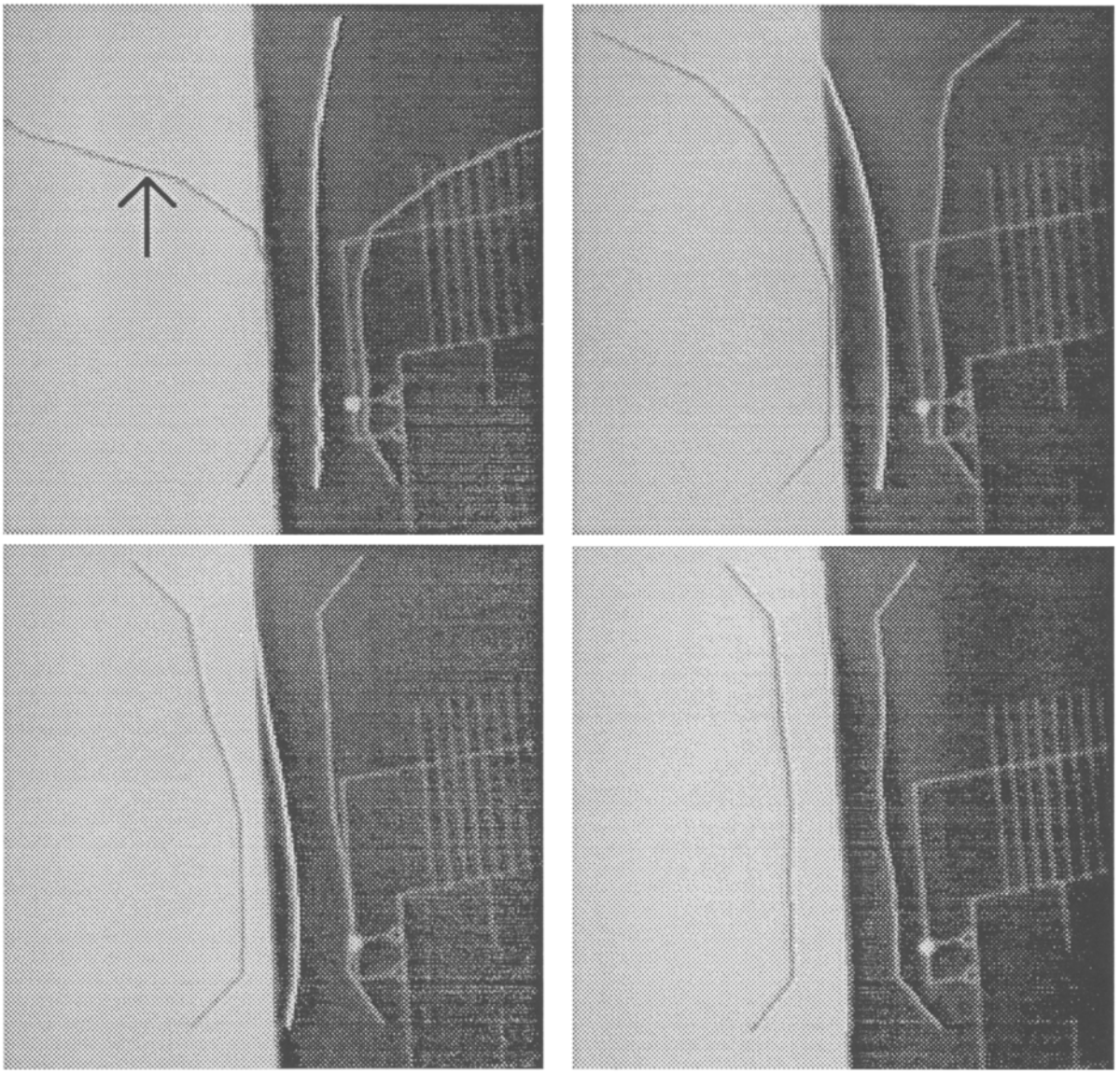

Fig. 5. This simulation demonstrates that portions of a snake with greater spatial uncertainty have shorter temporal memory, as predicted theoretically. The estimated contour (white line) is initially close to a feature; the search region (bounded by grey lines, arrowed in the first frame) is initialized to be larger at the top than the bottom. Temporal scale should therefore be shorter at the top and in the second frame the contour is indeed attracted to the feature more rapidly near the top. Subsequently the remainder of the snake locks on. (Frames are approximately at times $t = 0, 0.04, 0.08, 2.0$ seconds).

is fixed at the choice of r and it is most natural to choose r for the desired steady-state temporal scale

$$\tau_\infty = 2\,\frac{r}{\rho_\infty^2} \tag{30}$$

which governs the degree to which coherence of motion is exploited once the contour is fully locked. (Of course, if the maneuver causes lock to be lost, τ rapidly becomes very short as ρ increases; the coherence assumption is canceled and the contour is reactive, ready to follow the maneuver.)

The remaining parameter α_2 can be shown to determine a bound on the rate at which spatial-search scale grows, at time $t = 0$, in the absence of any features to track:

$$\dot{\rho} < \sqrt{\alpha_2} \tag{31}$$

So all four free parameters of the system are fixed in terms of the desired operational characteristics of the tracker.

6 Affine Invariant Shape Memory

So far, the template $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ has been set up to influence the initial conditions of the tracker by a direct probabilistic coupling. The estimated contour is initialized to the template shape, with a homogeneous and isotropic allowance for uncertainty via the covariance matrix $P(0)$. However, for 3-D tracking it is highly desirable to accommodate specially those deformations that occur as a result of projective effects. This could be done at any of several levels.

1. Allowing 2-D rotation.
2. Allowing 2-D affine transformations; this is sufficient to accommodate 3-D transformations of a planar contour, under affine projection.
3. Allowing 3-D affine transformations, sufficient to accommodate 3-D transformations of a space curve under affine projection.
4. Modelling 3-D rigid transformations of a 3-D model and perspective projection (Harris 1992).
5. Allowing full planar projective transformation.
6. Allowing 3-D projective transformations.

The last three of these are most general but require an extended Kalman filter to cope with nonlinearity. The first is least general but most tightly constrained, and is also nonlinear. The second and third are covered by the approach to be described here. An affine approximation is made to the camera projection (Horn 1986) with the benefit that the Kalman filter turns out to be linear and so to enjoy well understood convergence properties. Error in the approximation is absorbed by the general mechanism for uncertainty incorporated in the filter.

6.1 Affine Invariance

Let us first consider case 2 above: general 2-D affine invariance. It is in some respects appealing to try to develop some affinely invariant shape measure $I(\mathbf{X}, \mathbf{Y})$ which could then be used in an error-measure of the form

$$\|I(\mathbf{X}, \mathbf{Y}) - I(\bar{\mathbf{X}}, \bar{\mathbf{Y}})\|_I \tag{32}$$

with some suitable norm $\| \ldots \|_I$. Suitable candidates for I might be found, for instance, in (Mundy & Zisserman 1992). This would have the desired effect of making the template $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ an affinely invariant shape model because it was accessed only via $I(\ldots)$.

There are disadvantages, though, in relying entirely on an affine invariant $I(\ldots)$. Firstly affine invariants are nonlinear functions and that introduces nonlinearity into the tracking filter, whereas a linear filter is attainable (see below). Secondly it may be difficult to construct an $I(\ldots)$ that is not only invariant but also *complete*, in the sense that the error measure (32) should be sensitive to *all* nonaffine differences between curve and template. Thirdly, the overwhelming disadvantage relates to the modeling of sensor error. Use of the error measure (32) would appear to have the advantage of allowing the (affine) camera to be uncalibrated, because $(\mathbf{X}, \mathbf{Y})$ is also accessed via $I(\ldots)$. Our tracking framework rests firmly on modeling of sensor noise for which the *true* X, Y sensing frame is required to compute the Mahalanobis distance (13), not just some frame that is affinely related to (X, Y). This means that $(\mathbf{X}, \mathbf{Y})$ must be available explicitly, not just via $I(\ldots)$ and the camera must be calibrated, at least approximately.

Aiming for an error measure that is invariant to affine transformations in both curve and template is therefore not only needlessly ambitious but actually undesirable. Instead we should aim for a measure that is affinely invariant only to the template $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and not to the curve $\mathbf{X}$, $\mathbf{Y}$. Such a measure is derived next.

6.2 Invariant Template

Continuing to consider the 2-D affine case 2 above, we will exploit the fact that, instead of representing the template $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and the affine transformation A *explicitly*, the space of all possible transformations of the template can be represented as a subspace of $\mathcal{U}$ of the state space. This is mathematically efficient in the 2-D case. Later, it will prove also to be a highly effective generalization to regard this vector subspace $\mathcal{U}$ itself as the model, in place of the template.

Given a template obtained as one training view $(\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1)$, and since A has 6 degrees of freedom as in figure 6, the set $\{A(\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1)\}$ is a 6-dimensional vector subspace. The subspace can conveniently be expressed as a direct product of subspaces for the $\mathbf{X}$, $\mathbf{Y}$ processes respectively:

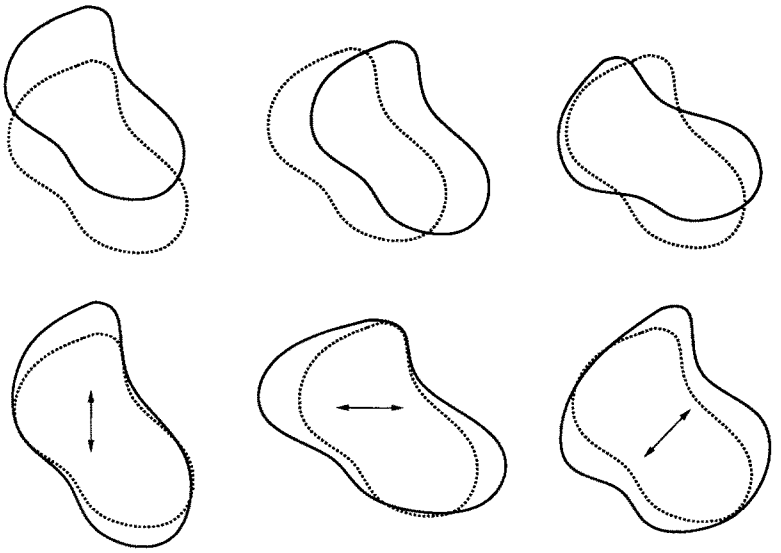

Fig. 6. The six degrees of freedom of a 2-D affine transformation are illustrated here: translation vertically and horizontally, rotation and scaling vertically, horizontally, and diagonally.

$$\underline{\mathcal{U}} = \underline{\mathcal{U}}_X \otimes \underline{\mathcal{U}}_Y$$

so that the typical element of $\underline{\mathcal{U}}$ is $v = (v_X, v_Y)$ where $v_X \in \underline{\mathcal{U}}_X$ and $v_Y \in \underline{\mathcal{U}}_Y$. The bases B_X, B_Y for $\underline{\mathcal{U}}_X$, $\underline{\mathcal{U}}_Y$ are

$$B_X = B_Y = \{\mathbf{1}, \bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1\} \tag{33}$$

where $\mathbf{1}$ is the M-vector

$$\mathbf{1} = \frac{1}{\sqrt{M}} (1, 1, \ldots, 1)^T$$

Each basis contains the vector $\mathbf{1}$ to allow translation and the vectors $\bar{\mathbf{X}}_1$, $\bar{\mathbf{Y}}_1$ to allow arbitrary linear functions of the template, including planar rotation, scaling, and foreshortening. We will assume, in general, that the bases are orthonormal with respect to the earlier defined inner product $< \ldots >$. This is achieved in the 2-D affine case by normalizing the position, size, and orientation of the template $(\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1)$ via the following steps:

1. Translate it so that its centroid is at the origin. This achieves $<\mathbf{1}, \bar{\mathbf{X}}_1> = <\mathbf{1}, \bar{\mathbf{Y}}_1> = 0$.
2. Rotate the template through an angle θ given by

$$\tan 2\theta = 2 \frac{<\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1>}{\|\bar{\mathbf{Y}}_1\|^2 - \|\bar{\mathbf{X}}_1\|^2}$$

   and this achieves $<\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1> = 0$.
3. Scale the template vertically by a factor $1/\|\bar{\mathbf{Y}}_1\|$ and horizontally by $1/\|\bar{\mathbf{X}}_1\|$ to achieve $\|\bar{\mathbf{X}}_1\| = \|\bar{\mathbf{Y}}_1\| = 1$.

Now, since already $\|\mathbf{1}\| = 1$, $\{\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1, \mathbf{1}\}$ forms an orthonormal set as required.

6.3 Prior p.d.f.

The next task is to show how the subspace model for shape amounts to incorporating affine invariance of the template into a distance measure and hence into the error measure prior p.d.f. for the curve. Consider the prior p.d.f. for positional uncertainty, whose covariance is the upper left submatrix $P_{11}(0)$ of $P(0)$. (The situation for the other nonzero submatrix, representing motion-uncertainty, is very similar.)

Now $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ is no longer a fixed template, but ranges over the subspace $\underline{\mathcal{U}}$ of affine transformations of the template $(\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1)$. Previously, the positional covariance $P_{11}(0)$ implied a fixed prior p.d.f.

$$p(\mathbf{X}, \mathbf{Y}) \propto \exp\left[-\frac{1}{2\alpha_1} (\|\mathbf{X} - \bar{\mathbf{X}}\|^2 + \|\mathbf{Y} - \bar{\mathbf{Y}}\|^2) \right].$$

Now that same expression is interpreted as a conditional p.d.f. $p(\mathbf{X}, \mathbf{Y} \mid \bar{\mathbf{X}}, \bar{\mathbf{Y}})$. From this conditional p.d.f., the prior p.d.f. $p(\mathbf{X}, \mathbf{Y})$ is computed by obtaining a maximum-likelihood estimate (m.l.e.) (Rao 1973) for $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ as a function of $(\mathbf{X}, \mathbf{Y})$. This is done by maximizing $p(\ldots \mid \ldots)$ above with respect to $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$, as it

ranges over the subspace $\underline{\mathcal{U}}$. The m.l.e. for the template is therefore the $(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \in \underline{\mathcal{U}}$ which minimizes the Mahalanobis distance

$$\|\mathbf{X} - \bar{\mathbf{X}}\|^2 + \|\mathbf{Y} - \bar{\mathbf{Y}}\|^2$$

The solution is

$$\bar{\mathbf{X}} = E_X' \mathbf{X}, \quad \text{where} \quad E_X' = \left(\sum_{\mathbf{v} \in B_X} \mathbf{v}\mathbf{v}^T \right) \underline{\mathcal{H}} \tag{34}$$

and similarly for $\bar{\mathbf{Y}}$—simply the *projection* of the current state $(\mathbf{X}, \mathbf{Y})$ onto the subspace $\underline{\mathcal{U}}$.

The prior p.d.f. for the contour can now be written as

$$p(\mathbf{X}, \mathbf{Y}) \propto \exp \left[-\frac{1}{2\alpha_1} (\|E_X \mathbf{X}\|^2 + \|E_Y \mathbf{Y}\|^2) \right] \tag{35}$$

where $E_X = I - E_X'$ and similarly for E_Y. This means that the prior p.d.f. depends only on the component of the current state $(\mathbf{X}, \mathbf{Y})$ that lies *outside* the subspace $\underline{\mathcal{U}}$.

Given the way in which $\underline{\mathcal{U}}$ is constructed, the term in round brackets in (35) is simply the minimum distance from $(\mathbf{X}, \mathbf{Y})$ to the subspace $\underline{\mathcal{U}}$. Since the definition of $\underline{\mathcal{U}}$ is invariant to an affine transformation of the training view $(\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1)$, the minimum-distance measure is, itself, invariant to affine transformation of the training view, as required.

6.4 Invariant Filter

The implication of the prior p.d.f. $p(\ldots)$ above is that the positional covariance $P_{11}(t)$ of $\mathbf{X}$ is given initially by

$$(P_{11}(0))^{-1} = \frac{1}{\alpha_1} (E_X)^T \underline{\mathcal{H}} E_X \tag{36}$$

so that $P_{11}(0)$ takes the same value $\alpha_1 \underline{\mathcal{H}}^{-1}$ as before *outside* the subspace $\underline{\mathcal{U}}$ but is effectively infinite inside the subspace. This reflects the fact that the template determines initially only the nonaffine component of the estimated curve. In a continuous Kalman filter this would create a singularity problem, in that the initial Kalman gain is unbounded. In the discrete filter, described later, the problem does not arise because the Kalman gain is bounded even though (36) is rank deficient. Provided a feature $(\mathbf{X}_f, \mathbf{Y}_f)$ is observable at time $t = 0$ it will be absorbed via the Kalman gain into the estimate $(\hat{\mathbf{X}}(0), \hat{\mathbf{Y}}(0))$. Within the affine subspace, the initial estimate is simply a copy of the feature:

$$E_X' \hat{\mathbf{X}}(0) = E_X' \mathbf{X}_f$$

whereas outside the subspace it is a linear mixture of template and feature.[1] The initial conditions for the estimate are now determined both inside and outside the subspace and this is reflected in the fact that P_{11} becomes nonsingular.

7 Persistent template

So far, the template or training views have had an influence on the tracker that continually decreased over time so that, in the steady state, no shape memory remains. However, in practice, it is crucial that a shape-specific tracker should continue, throughout its lifetime, to retain some shape memory. It is essential to stability that it should do so, otherwise there is undue disturbance when features are temporarily obscured and the tracker is bumped out of its steady state. The more complex the shape to be tracked, the worse is the instability when lock is lost, and this is illustrated in figure 7. With

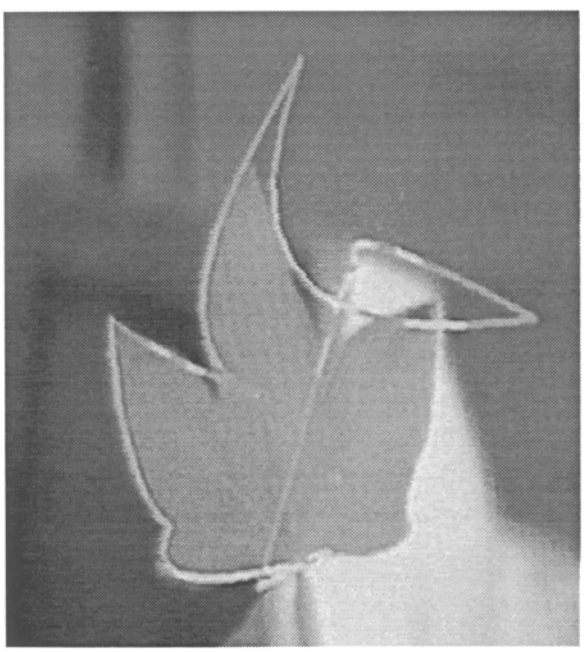

Fig. 7. The persistent template is essential to the stability of the contour tracker, especially with more complex shapes. The figure shows a contour without a persistent template. Over the portion of the contour that has lost lock, it has become too tangled to be able to recover lock subsequently.

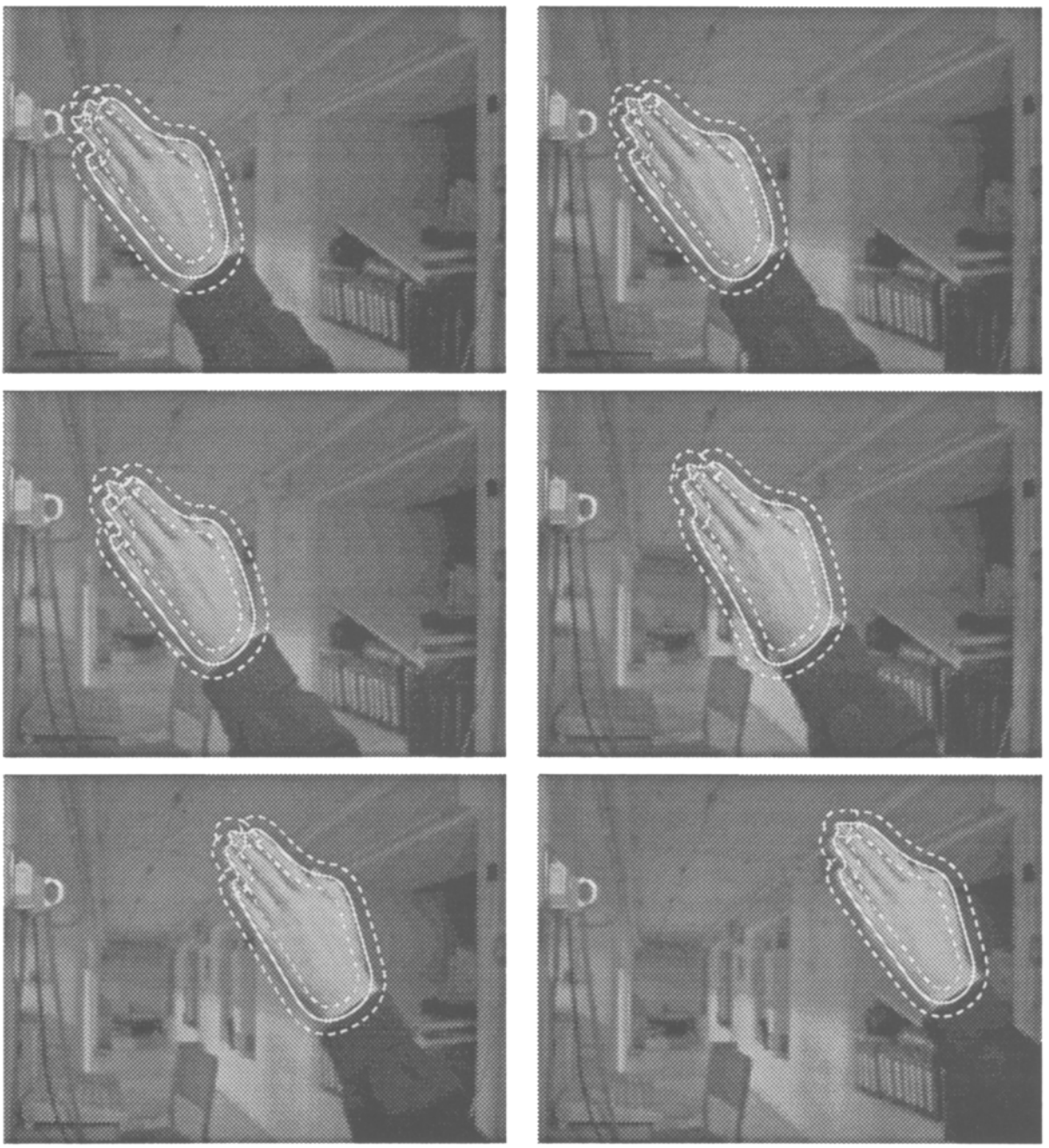

Fig. 8. A hand, accelerating from left to right, sweeps across a cluttered background (total elapsed time: 1.1 s). The template imposes shape specificity without total rigidity. Note that in the fourth frame the lower left corner of the contour is momentarily distracted by the chair but the disturbance is successfully filtered out over time.

persistence however, the tracker remains stable with or without features and retains its ability to track across a cluttered background as figure 8 shows.

The persistent template has the effect of building additional smoothness into the estimate $\hat{\mathbf{X}}$, beyond the implicit smoothness of the B-spline. Probabilistically, it can be regarded as a prior constraint, for the stochastic component of the dynamical model, specified by its mean $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and by the covariance implicit in the Kalman gain $\bar{K}$ defined below.

7.1 Mechanism

The mechanism for persistence of the template is as follows. In addition to the feature $(\mathbf{X}_f, \mathbf{Y}_f)$, a *virtual* input of $\mathbf{0}$ is applied to the filter, but coupled only *outside* the subspace $\underline{\mathcal{U}}$. In the 2-D planar case, for instance, this tends to extinguish those components of shape that are not related by affine transformation to the template. Now the Kalman filter becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\hat{\mathbf{X}}\\ \dot{\hat{\mathbf{X}}}\end{pmatrix} = \begin{pmatrix}\dot{\hat{\mathbf{X}}}\\ \mathbf{0}\end{pmatrix} + K(\mathbf{X}_f - \hat{\mathbf{X}}) - \bar{K}\,E_X\,\hat{\mathbf{X}} \qquad (37)$$

The extra, final term is the virtual input and its associated Kalman gain is $\bar{K} = P\,H^T \bar{R}^{-1}$ where the covariance (spectral density) $\bar{R}$ associated with the persistent template is given by

$$\bar{R}^{-1} = \frac{1}{\bar{r}}\,(E_X)^T\,\underline{\mathcal{H}}E_X \qquad (38)$$

and $\bar{r}$ is a covariance (spectral density). Typically we choose $\bar{r} < r$ so that the real input to the tracker dominates the virtual one.

7.2 Results

The effectiveness of the affine tracker with persistent template is demonstrated by the tracking of a moving hand. The four affine degrees of freedom are independently exercised in figure 9 and successfully tracked. The transformation of the hand outline is not *precisely* affine because the hand is not perfectly planar, has extremal boundaries and may flex a little during motion. However the stochastic allowance for shape uncertainty in the filter ensures that such deviations can be accommodated. When a nonaffine distortion of significant magnitude occurs, however, the tracker correctly ignores it, preferring to maintain its memorized shape (figure 10).

8 Subspaces as Shape Models

As claimed earlier, the subspace $\mathcal{U}$ can itself be regarded as a rather general form of prior model for shape. Varying the structure of the subspaces $\mathcal{U}_X$, $\mathcal{U}_Y$ allows different modelling assumptions to be applied to the tracker.

Camera Rotation. A simple case of a restricted subspace occurs if motion is restricted to pan and tilt of the camera. In that case, image motion is approximately a rigid 2-D translation and the appropriate subspaces are given by the bases

$$B_X = B_Y = \{\mathbf{1}\}$$

so that $\underline{\mathcal{U}}$ has dimension $1 + 1 = 2$ as appropriate for the two degrees of freedom of image-plane translation.

Space Curve. A larger subspace than the six-dimensional one used in the planar, affine case is needed for space-curve motion. Affine transformations of a 3-D space-curve under rigid transformations can be modeled by a subspace $\underline{\mathcal{U}}$, defined as follows. Given views $(\bar{\mathbf{X}}_i, \bar{\mathbf{Y}}_i)$, $i = 1, 2, \ldots$, B_X and B_Y can be constructed from three views, corresponding to the observation of Ullman and Basri (1991) that, under affine projection, any view of an object is a linear combination of three prototype views, so that

$$B_X = \{\mathbf{1}, \bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \bar{\mathbf{X}}_3\} \qquad B_Y = \{\mathbf{1}, \bar{\mathbf{Y}}_1, \bar{\mathbf{Y}}_2, \bar{\mathbf{Y}}_3\}$$

An alternative version, requiring only 1½ views, constructs the bases as

$$B_X = B_Y = \{\mathbf{1}, \bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1, \bar{\mathbf{X}}_2\}.$$

The use of 1½ views here is effectively a form of affine stereo (Koenderink & Van Doorn, 1991). It can be regarded as a single view $(\bar{\mathbf{X}}_1, \bar{\mathbf{Y}}_1)$ together with horizontal stereoscopic disparities $\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1$ which together imply the underlying 3-D structure. Thus the bases B_X, B_Y each, independently, span the space of affine $\mathcal{R}^3 \to \mathcal{R}$ transformations or, jointly, the affine transformations $\mathcal{R}^3 \to \mathcal{R}^2$, as required.

Multiple Space Curves. A further, more ambitious, development of the mechanism would be to include views of more than one shape into the tracker, and then its shape memory would span both shapes, linearly, and with affine invariance. In that case the tracker could track either of two objects without the initial knowledge of which one might appear. Alternatively the views might be from different states of a deformable object in which case, to the extent that the deformation could be approximated by a 3-D affine transformation, the tracker would follow the deforming object.

It is assumed throughout that $\underline{\mathcal{U}}$ is orthonormal which was true in 2-D if the template was normalized.

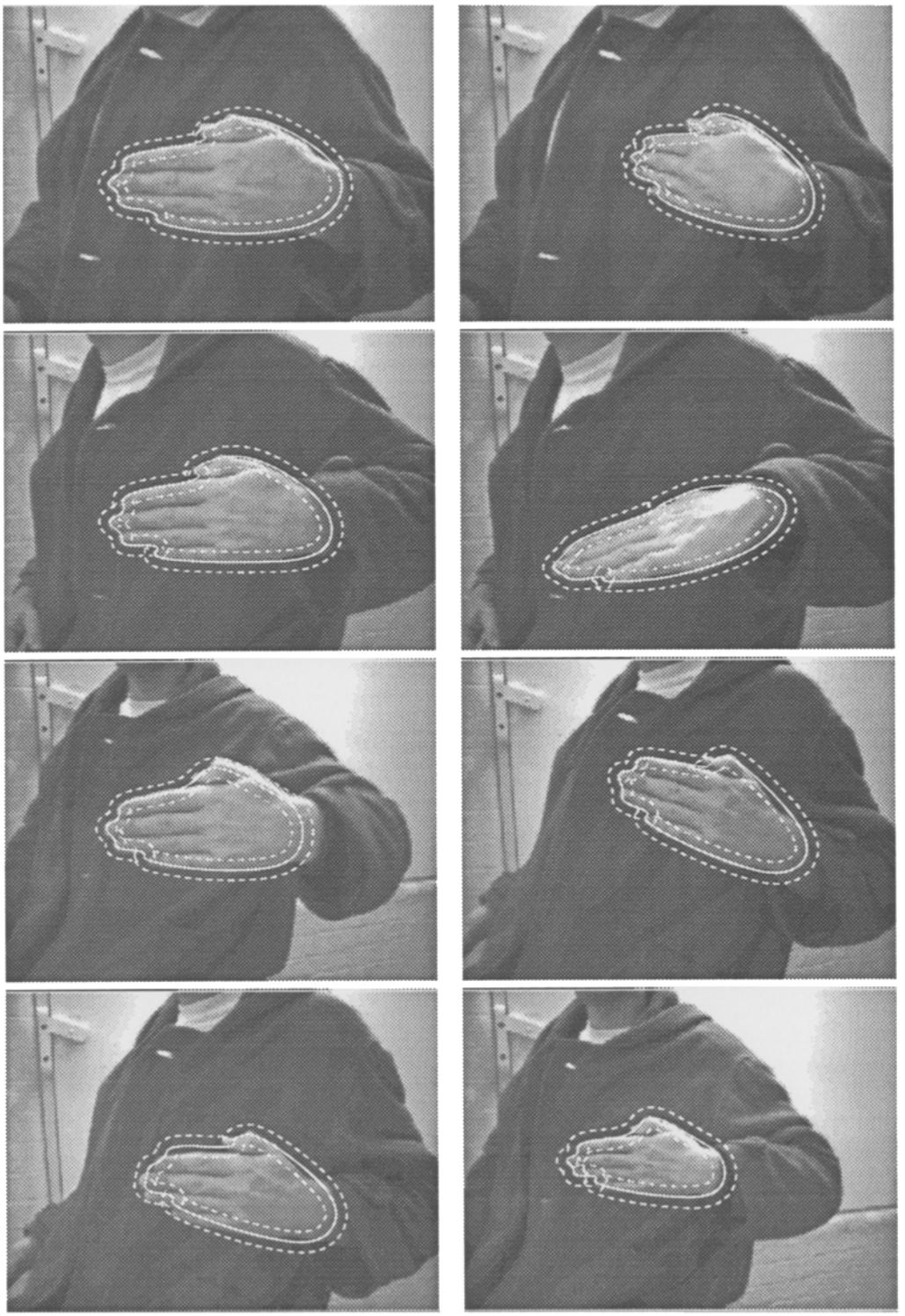

Fig. 9. Handtracking. Exercising. 4 affine degrees of freedom, in raster order: home—slant horizontal—home—slant vertical—home—rotate—home—distance scaling.

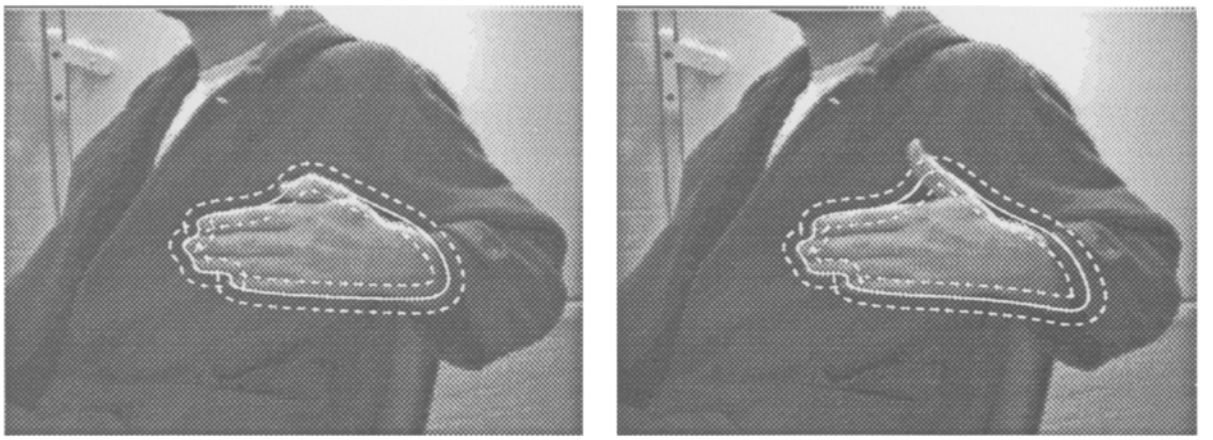

Fig. 10. Hand tracking. A nonaffine distortion, extending the thumb, is correctly ignored by the tracker.

In 3-D, when several training views are used, $\underline{U}$ needs to be orthormalized, for instance by the Gram-Schmidt procedure. Trackers for each of the subspace models are generated using exactly the same structures as for the planar affine case, simply by redefining E_X, E_Y, in (34) in terms of the new subspace $\underline{U}$. The 2-D result about the prior p.d.f. for $(\mathbf{X}, \mathbf{Y})$ being invariant to affine transformations of the training view also extends to the 3-D rigid-body case, now with respect to 3-D affine transformations.

9 Discrete Filter

The continuous Kalman filter model used so far is good for analysis, deriving the scale behavior over time and the steady-state performance. However, in practice, measurements are discrete, synchronized with video frames. It is crucial to the maintenance of real-time performance, that no video frame should be missed. This means that it may not be possible to sample all curve points within a single frame period Δt. Instead it is better to process as many points, chosen randomly with a uniform spatial distribution over the curve, as the frame period allows. Then, as soon as a new video frame is available, $P(t)$ and $K(t)$ are updated to allow for the elapsed time Δt, and the filter continues with the new frame using randomly sampled measurements to update the estimated curve.

9.1 Measurements

In this practical framework, it is convenient to regard a unit measurement not as a set of time-varying B-spline coefficients $(\mathbf{X}_f, \mathbf{Y}_f)$ but as a single (X, Y) point observation at the curve parameter s, taken at a discrete time t. This observation has an associated discrete Kalman gain matrix $K(s, t)$ defined by

$$K(s, t) = \mathbf{P}(t)H^T(s)\frac{1}{\sigma^2} \tag{39}$$

where $H(s)$ is the point-wise measurement matrix defined in (2) and

$$\frac{1}{\sigma^2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{40}$$

is the inverse covariance for the measured point, assumed isotropic. Then $K(s, t)$ is used twice, once in the $\mathbf{X}$ process and once in the $\mathbf{Y}$ process.

For the purpose of setting parameters to obtain desired tracking performance, the standard deviation σ of individual measurements must be related to the measurement variance spectral density r in the continuous filter. Assuming that L measurements are made in each span—that is $N\,L$ measurements in total per video frame—and that they are evenly distributed over the curve parameter s, then it can be shown that

$$\sigma^2 = \frac{Lr}{\Delta t} \tag{41}$$

Once r is fixed to achieve desired filter properties, as earlier in section 5, σ for the discrete filter is therefore also fixed.

It is highly desirable to take into account the aperture problem (Harris & Stennett 1990). In practice, if the point-measurement model is used, the estimated curve does not rotate freely, because the parameterization of

the estimated curve is projected along normals onto the feature curve, and this discourages motion orthogonal to the normals. The problem is cured when displacement only along the normal $\mathbf{n}(s)$ is used. In that case the Kalman gain $K(s, t)$ is applied once to a coupled **X** and **Y** process, the coupling arising because the measurement model is no longer isotropic. In place of the isotropic matrix in (40) the inverse covariance for the normal measurement is

$$\frac{1}{\sigma^2}\,\mathbf{n}(s)\mathbf{n}(s)^T$$

and this can be used to define the Kalman gain for the measurement. However, for the remainder of this section on the discrete filter we will, for simplicity of notation, treat the point-measurement model. The above modifications required to take account of the aperture problem and then quite straightforward to apply.

9.2 Filter

The discrete-time measurement model assumes that observations are made at times $t_k = t_0 + k\Delta t$, where Δt is the interval between video frames. Our notation will use suffix k to refer to a discrete-time quantity at time t_k. The continuous filter described earlier can be transformed into a discrete one, following the treatment given by Gelb (1974).

First, over the interval $t_{k-1} \le t < t_k$ during which no observations are made, the filter (16) becomes simply

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\hat{\mathbf{X}}\\ \hat{\dot{\mathbf{X}}}\end{pmatrix} = \begin{pmatrix}\hat{\dot{\mathbf{X}}}\\ \mathbf{0}\end{pmatrix} \tag{42}$$

which can be integrated directly. Approximating to $O(\Delta t)$, this gives the discrete prediction $\hat{\mathbf{X}}_k$ immediately prior to the time $t = t_k$:

$$\begin{pmatrix}\hat{\mathbf{X}}_k\\ \hat{\dot{\mathbf{X}}}_k\end{pmatrix} = \begin{bmatrix}\mathbf{I} & (\Delta t)\mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{pmatrix}\hat{\mathbf{X}}_{k-1}\\ \hat{\dot{\mathbf{X}}}_{k-1}\end{pmatrix} \tag{43}$$

The Riccati equation (19) is also simplified, in the absence of measurements, to

$$\frac{\mathrm{d}P}{\mathrm{d}t} = F\,P - P\,F^T + Q$$

which, being linear, can also be integrated directly to give

$$\hat{P}_k = \begin{bmatrix}\mathbf{I} & (\Delta t)\mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\hat{P}_{k-1}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ (\Delta t)\mathbf{I} & \mathbf{I}\end{bmatrix} + \Delta t Q \tag{44}$$

and again this has been approximated to $O(\Delta t)$.

Secondly, discrete measurements for time t_k are applied sequentially. A given observation of a point feature $(X_f(s), Y_f(s))$ is applied via the Kalman gain

$$K_k(s) \equiv K(s, t_k) = \hat{P}_k\begin{pmatrix}H(s)^T\\ \mathbf{0}\end{pmatrix}\left[(H(s)\ \mathbf{0})\,\hat{P}_k\begin{pmatrix}H(s)^T\\ \mathbf{0}\end{pmatrix} + \sigma^2\right]^{-1} \tag{45}$$

where $\hat{P}_k$ is the covariance of the state as it was immediately before application of the current measurement—that is, taking into account all measurements for $t = t_k$ up to, but not including, the current measurement. The current measurement is applied to the state by

$$\begin{pmatrix}\hat{\mathbf{X}}_k\\ \hat{\dot{\mathbf{X}}}_k\end{pmatrix} \to \begin{pmatrix}\hat{\mathbf{X}}_k\\ \hat{\dot{\mathbf{X}}}_k\end{pmatrix} + K_k(s)\,(X_f(s, t_k) - H(s)\hat{\mathbf{X}}_k) \tag{46}$$

The state covariance is updated by a discrete Riccati equation,

$$\hat{P}_k \to [\mathbf{I} - K_k(s)\,(\mathrm{H}(s)\ \ \mathbf{0})]\,\hat{P}_k \tag{47}$$

where $H(s)$ is the point-wise measurement matrix defined earlier in (2).

9.3 Persistent Template

In an exact discretization the persistent template (37) should be treated as a continuous measurement, integrated over the time interval $t_{k-1} \le t \le t_k$, as part of the prediction phase. When Δt is small this can be approximated by one discrete virtual observation per timestep. Allowing for invariance over the subspace $\underline{\mathcal{U}}$, as earlier, the virtual observation of the template is applied via the discrete Kalman gain

$$\bar{K}_k = \hat{P}_k\,H^T[H\,\hat{P}_k\,H^T + (\Delta t\,\bar{R})I]^{-1} \tag{48}$$

where $\bar{R}$ is the variance spectral density (38) from the continuous filter and

$$H = (\mathbf{I}\ \ \mathbf{0})$$

as in the continuous filter. The template update is

$$\begin{pmatrix}\hat{\mathbf{X}}_k\\ \hat{\dot{\mathbf{X}}}_k\end{pmatrix} \to \begin{pmatrix}\hat{\mathbf{X}}_k\\ \hat{\dot{\mathbf{X}}}_k\end{pmatrix} - \bar{K}_k\,\hat{\mathbf{X}}_k \tag{49}$$

with covariance updated by

$$\hat{P}_k \to [\mathbf{I} - \bar{K}_k\,H]\,\hat{P}_k \tag{50}$$

9.4 Computational Complexity

The implementation of this filter requires $O(M^2)$ floating-point operations for the prediction step. Each observation takes $O(M^3)$ arithmetic operations, due to the $2M \times 2M$ matrix multiplication in (47). Finally the virtual observation for the persistent template requires $O(M^3)$ operations. The expense of each observation may be reduced by using the *Information Filter* or *Inverse Kalman Filter* (Maybeck 1979). This makes prediction the expensive step, giving $O(M^3)$ for prediction and $O(M^2)$ for each observation, which may be preferable in the typical case that there are many (more than M) observations per time-step. Furthermore, the information filter allows an efficient parallel implementation because the observation updates can be applied in an arbitrary order. This contrasts with the Kalman filter described earlier in the section in which the covariance update for one measurement must be calculated before the next measurement can be applied.

9.5 Spatial Search

The prediction step (43) is used to find the expected feature position in each frame of the image sequence, and the corresponding covariance is used to constrain the search for the features within that image. Search along the predicted contour normal is bounded within the uncertainty ellipse, derived from the state covariance, which will contain the feature with a 98% likelihood (2 standard deviations). Anisotropy arising from the aperture problem and the measurement of normal displacements only is taken into account in the construction of the uncertainty ellipse. Full details are given by Curwen (1993).

Features are not necessarily found at all sample points. They may for example be obscured along a portion of the contour, or of insufficient contrast to be registered. In such a case no observation is applied so that no new information is introduced into the state and covariance for that sample point.

10 Conclusions

The value of the statistical basis for contour tracking has been established by elucidating and demonstrating the mechanism for automatic control of spatiotemporal scale. Despite the substantial size of the state space needed to deal with the geometric complexity of the contour, some natural assumptions lead to a system that has few free parameters; and those are fixed, via control-theoretic analysis, to obtain desired dynamic behavior. The incorporation of the template mechanism with its affine invariance has proved to be crucial in procuring shape-selective tracking that is immune to background clutter.

It remains for future work to experiment further with extensions of the subspace mechanism used here to represent affine degrees of freedom. This includes tracking of space curves, tracking with two or more models simultaneously, and integration of the learning of shapes and their associated spatiotemporal uncertainty into the tracking process.

Acknowledgments

The financial support of the SERC, the EEC and Oxford Metrics is gratefully acknowledged. Discussions with M. Brady, R. Brockett, N. Ferrier, A Nairac, B. Ripley, and C. Rothwell were most helpful.

Notes

1. The Kalman filter is in fact acting, in this initialization step, as a Wiener filter (Gelb 1974).
2. It is also possible to write an exact solution to the integral using exponentials.

References

Ayache, N., Cohen, I., and Herlin, I. 1992. Medical image tracking. In A. Blake and A. Yuille, eds., *Active Vision*, pp. 285–302, MIT Press: Cambridge, MA.

Bar-Shalom, Y., and Fortmann, T. 1988. *Tracking and Data Association*. Academic Press: San Diego, CA.

Bartels, R., Beatty, J., and Barsky, B. 1987. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann: San Mateo, CA.

Bennett, A., and Craw, I. 1991. Finding image features for deformable templates and detailed prior statistical knowledge. In P. Mowforth, ed., *Proc. British Machine Vision Conference*, pp. 233–239, Glasgow. Springer-Verlag: London.

Blake, A. 1992. Computational modelling of hand-eye coordination, *Proc. Roy. Soc. London B.*, 337:351–360.

Blake, A., Brady, J., Cipolla, R., Xie, Z., and Zisserman, A. 1991. Visual navigation around curved obstacles, *Proc. IEEE Intern. Conf. Robot. Automat.* 3:2490–2499.

Blake, A., and Yuille, A., eds. 1992. *Active Vision*, MIT Press: Cambridge, MA.

Blake, A., Zisserman, A. and Cipolla, R. 1992. Visual exploration of freespace. In A. Blake and A. Yuille, eds., *Active Vision*, pp. 175–188, MIT Press: Cambridge, MA.

Bookstein, F.L. 1988. Thin-plane splines and the decomposition of deformations, *IEEE Trans. Patt. Anal. Mach. Intell.* 10:

Cipolla, R., and Blake, A. 1992. Motion planning using image divergence and deformation. In A. Blake and A. Yuille, eds., *Active Vision*, pp. 39–58. MIT Press: Cambridge, MA.

Cipolla. R., and Yamamoto, M. 1990. Stereoscopic tracking of bodies in motion, *Image and Vis. Comput.* 8(1):85–90.

Curwen, R. 1993. *Dynamic and Adaptive Contours.* Ph.D. thesis, University of Oxford.

Curwen, R., and Blake, A. 1992. Dynamic contours: real-time active splines. In A. Blake and A. Yuille, eds., *Active Vision*, pp. 39–58. MIT Press: Cambridge, MA.

Dickmanns, E., and Graefe, V. 1988. Applications of dynamic monocular machine vision, *Mach. Vis. Applic.* 1:241–261.

Faux, I., and Pratt, M. 1979. *Computational Geometry for Design and Manufacture.* Ellis-Horwood (VCH Pubs: New York).

Fischler, M.A., and Elschlager, R.A. 1973. The representation and matching of pictorial structures, *IEEE Trans. Computers* C-22(1).

Gelb, A., ed. 1974. *Applied Optimal Estimation.* MIT Press: Cambridge, MA.

Grenander, U., Chow, Y., and Keenan, D.M. 1991. *HANDS. A Pattern Theoretical Study of Biological Shapes.* Springer-Verlag: New York.

Harris, C. 1992. Tracking with rigid models. In A. Blake, and A. Yuille, eds., *Active Vision*, pp. 59–74. MIT Press: Cambridge, MA.

Harris, C., and Stennett, C. 1990. Rapid—a video-rate object tracker, *Proc. 1st British Mach. Vis. Conf.*, pp. 73–78, Oxford.

Horn, B. 1986. *Robot Vision.* McGraw-Hill: New York.

Inoue, H., and Mizoguchi, H. 1985. A flexible multi window vision system for robots, *Proc. 2nd Intern. Symp. Robot. Res.*, pp. 95–102.

Kass, M., Witkin, A., and Terzopoulos, D. 1987. Snakes: Active contour models, *Proc. 1st Intern. Conf. Comput. Vis.*, pp. 259–268, London.

Koenderink, J., and Van Doorn, A. 1991. Affine structure from motion, *J. Opt. Soc. Amer. A.* 8(2):337–385.

Lipson, P., Yuille, A., Keeffe, D., Cavanaugh, J., Taafe, J., and Rosenthal, D. 1990. Deformable templates for feature extraction from medical images. In O. Faugeras, ed., *Proc. 1st Europ. Conf. Comput. Vis.*, France, pp. 413–417. Springer-Verlag: New York.

Lowe, D. 1992. Robust model-based motion tracking through the integration of search and estimation, *Intern. J. Comput. Vis.* 8(2):113–122.

Maybeck, P. 1979. *Stochastic Models, Estimation and Control*, vol I. Academic Press: San Deigo, CA.

Menet, S., Saint-Marc, P., and Medioni, G. 1990. B-snakes: implementation and application to stereo, *Proceedings DARPA*, pp. 720–726.

Mundy, J., and Zisserman, A. 1992. *Geometric Invariance in Computer Vision.* MIT Press: Cambridge, MA.

Rao, C. 1973. *Linear Statistical Inference and Its Applications*, Wiley: New York.

Scott, G. 1987. The alternative snake—and other animals, *Proc. 3rd Alvey Vis. Conf.* pp. 341–347.

Sullivan, G. 1992. Visual interpretation of known objects in constrained scenes, *Phil. Trans. Roy. Soc. London B* 337: 109–118.

Szeliski, R., and Terzopoulos, D. 1991. Physically based and probabilistic modeling for computer vision. In B.C. Vemuri, ed., *Proc. SPIE 1570, Geometric Methods in Computer Vision*, pp. 140–152, San Diego, CA. Society of Photo-Optical Instrumentation Engineers.

Terzopoulos, D., and Metaxas, D. 1991. Dynamic 3D models with local and global deformations: deformable superquadrics, *IEEE Trans. Patt. Anal. Mach. Intell.* 13(7).

Thompson, D.W., and Mundy, J.L. 1987. Three-dimensional model matching from an unconstrained viewpoint, *Proc. Intern. Conf. Robot. Automat.*, Raleigh, NC.

Ullman, S., and Basri, R. 1991. Recognition by linear combinations of models, *IEEE Trans. Patt. Analy. Mach. Intell.* 13(10):992–1006.

Wang, H., and Brady, M. 1992. Vision for mobile robots, *Proc. Roy. Soc. London B.* 337:341–350.

Yuille, A.L., Hallinan, P.W., and Cohen, D.S. 1992. Detecting facial features using deformable templates, *Intern. J. Comput. Vis.* 8, 2, 99–112.