Contour Tracking by Stochastic Propagation of Conditional Density

Michael Isard and Andrew Blake

Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK.
{misard, ab}@robots.ox.ac.uk, 01865 273919

Abstract. The problem of tracking curves in dense visual clutter is a challenging one. Trackers based on Kalman filters are of limited use; because they are based on Gaussian densities which are unimodal, they cannot represent simultaneous alternative hypotheses. Extensions to the Kalman filter to handle multiple data associations work satisfactorily in the simple case of point targets, but do not extend naturally to continuous curves. A new, stochastic algorithm is proposed here, the CONDENSATION algorithm — Conditional Density Propagation over time. It uses 'factored sampling', a method previously applied to interpretation of static images, in which the distribution of possible interpretations is represented by a randomly generated set of representatives. The CONDENSATION algorithm combines factored sampling with learned dynamical models to propagate an entire probability distribution for object position and shape, over time. The result is highly robust tracking of agile motion in clutter, markedly superior to what has previously been attainable from Kalman filtering. Notwithstanding the use of stochastic methods, the algorithm runs in near real-time.

1 The problem of tracking curves in clutter

The purpose of this paper is to establish a stochastic framework for tracking curves in visual clutter, and to propose a powerful new technique — the CONDENSATION algorithm. The new approach is rooted in strands from statistics, control theory and computer vision. The problem is to track outlines and features of foreground objects, modelled as curves, as they move in *substantial* clutter, and to do it at, or close to, video frame-rate. This is challenging because elements in the background clutter may mimic parts of foreground features. In the most severe case, the background may consist of objects similar to the foreground object, for instance when a person is moving past a crowd. Our framework aims to dissolve the resulting ambiguity by applying probabilistic models of object shape and motion to analyse the video-stream. The degree of generality of these models must be pitched carefully: sufficiently specific for effective disambiguation but sufficiently general to be broadly applicable over entire classes of foreground objects.

1.1 Modelling shape and motion

Effective methods have arisen in computer vision for modelling shape and motion. When suitable geometric models of a moving object are available, they can

be matched effectively to image data, though usually at considerable computational cost [17, 26, 18]. Once an object has been located approximately, tracking it in subsequent images becomes more efficient computationally [20], especially if motion is modelled as well as shape [12, 16]. One important facility is the modelling of curve segments which interact with images [29] or image sequences [19]. This is more general than modelling entire objects but more clutter-resistant than applying signal-processing to low-level corners or edges. The methods to be discussed here have been applied at this level, to segments of parametric B-spline curves [3] tracking over image sequences [8]. The B-spline curves could, in theory, be parameterised by their control points. In practice this allows too many degrees of freedom for stable tracking and it is necessary to restrict the curve to a low-dimensional parameter x, for example over an affine space [28, 5], or more generally allowing a linear space of non-rigid motion [9].

Finally, probability densities $p(x)$ can be defined over the class of curves [9], and also over their motions [27, 5], and this constitutes a powerful facility for tracking. Reasonable default functions can be chosen for those densities. However, it is obviously more satisfactory to measure the actual densities or estimate them from data-sequences $(x_1, x_2, \ldots)$. Algorithms to do this assuming Gaussian densities are known in the control-theory literature [13] and have been applied in computer vision [6, 7, 4].

1.2 Sampling methods

A standard problem in statistical pattern recognition is to find an object parameterised as x with prior $p(x)$, using data z from a single image. (This is a simplified, static form of the image sequence problem addressed in this paper.) In order to estimate x from z, some information is needed about the conditional distribution $p(z|x)$ which measures the *likelihood* that a hypothetical object configuration x should give rise to the image data z that has just been observed. The data z could either be an entire grey-level array or a set of sparse features such as corners or, as in this paper, curve fragments obtained by edge detection. The posterior density $p(x|z)$ represents all the knowledge about x that is deducible from the data. It can be evaluated in principle by applying Bayes' rule to obtain

$$p(x|z) = kp(z|x)p(x) \tag{1}$$

where k is a normalisation constant that is independent of x. In the general case that $p(z|x)$ is multi-modal $p(x|z)$ cannot be evaluated simply in closed form: instead iterative sampling techniques can be used.

The first use of such an iterative solution was proposed by Geman and Geman [11] for restoration of an image represented by mixed variables, both continuous (pixels) and discrete (the 'line process'). Sampling methods for recovery of a parametric curve x by sampling [24, 14, 25] have generally used spatial Markov processes as the underlying probabilistic model $p(x)$. The basic method is *factored sampling* [14]. It is useful when the conditional observation probability $p(z|x)$ can be evaluated pointwise and sampling it is not feasible and when, conversely, the prior $p(x)$ can be sampled but not evaluated. The algorithm

estimates means of properties $f(x)$ (e.g. moments) of the posterior $p(x|z)$ by first generating randomly a sample $(s_1, s_2, \ldots)$ from the density $p(x)$ and then weighting with $p(z|x)$:

$$E[f(x)|z] \approx \frac{\sum_{n=1}^{N} f(s_n)p(z|s_n)}{\sum_{n=1}^{N} p(z|s_n)} \tag{2}$$

where this is asymptotically ($N \to \infty$) an unbiased estimate. For example, the mean can be estimated using $f(x) = x$ and the variance using $f(x) = xx^T$. If $p(x)$ is a spatial Gauss-Markov process, then Gibbs sampling from $p(x)$ is used to generate the random variates $(s_1, s_2, \ldots)$. Otherwise, for low-dimensional parameterisations as in this paper, standard, direct methods can be used for Gaussians[1] — we use rejection sampling [21]. Note that, in the case that the density $p(z|x)$ is normal, the mean obtained by factored sampling would be consistent with an estimate obtained more conventionally, and efficiently, from linear least squares estimation. For multi-modal distributions which cannot be approximated as normal, so that linear estimators are unusable, estimates of mean x by factored sampling continue to apply.

Sampling methods have proved remarkably effective for recovering static objects, notably hands [14] and galaxies [24], in clutter. The challenge addressed here is to do this over time, estimating $x(t)$ from time-varying images $z(t)$.

1.3 Kalman filters and data-association

Spatio-temporal estimation, the tracking of shape and position over time, has been dealt with thoroughly by Kalman filtering, in the relatively clutter-free case in which $p(z|x)$ can satisfactorily be modelled as Gaussian [16, 12, 23] and can be applied to curves [27, 5]. These solutions work relatively poorly in clutter which easily 'distracts' the spatio-temporal estimate $\hat{x}(t)$. With simple, discrete features such as points or corners combinatorial data-association methods can be effective, including the 'JPDAF' [2, 22] and the 'RANSAC' algorithm [10]. They allow several hypotheses about which data-elements 'belong' to the tracked object to be held simultaneously, and less plausible hypotheses to be progressively pruned. Data association methods do not, however, apply to moving curves where the features are continuous objects, and a more general methodology is demanded.

1.4 Temporal propagation of conditional densities

The Kalman filter as a recursive linear estimator is a very special case, applying only to Gaussian densities, of a more general probability density propagation process. In continuous time this can be described in terms of diffusion [15], governed by a 'Fokker-Planck' equation [1], in which the density for $x(t)$ drifts and spreads under the action of a stochastic model of its dynamics. The random component of the dynamical model leads to spreading — increasing uncertainty

[1] Note: the presence of clutter causes $p(z|x)$ to be non-Gaussian, but the prior $p(x)$ may still happily be Gaussian, and that is what will be assumed in our experiments.

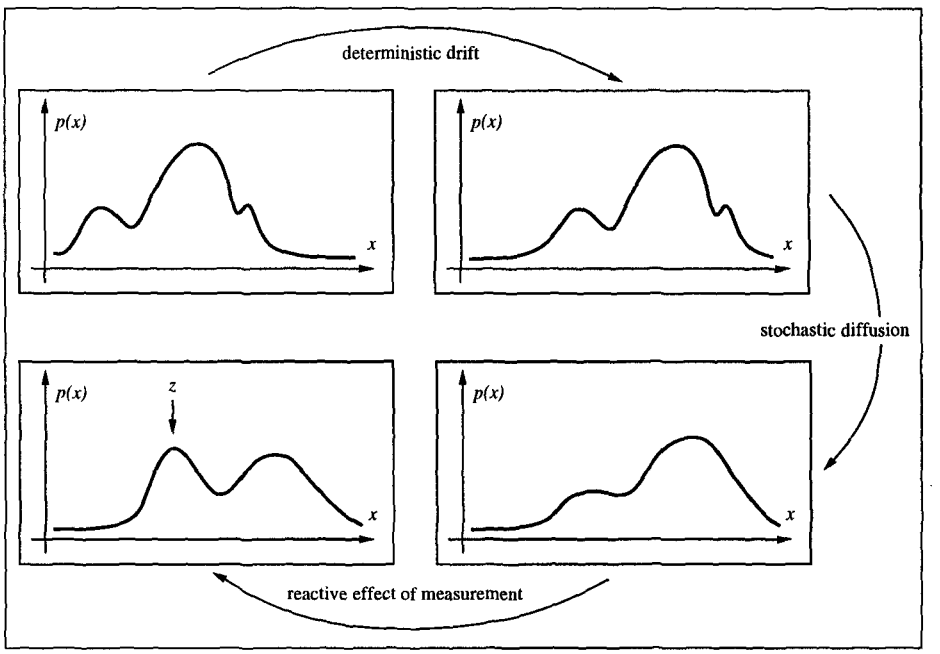

Fig. 1. ***Probability density propagation.*** *Propagation is depicted here as it occurs over a discrete time-step. There are three phases: drift due to the deterministic component of object dynamics; diffusion due to the random component; reactive reinforcement due to measurements.*

— while the deterministic component causes a drift of the mass of the density function, as shown in figure 1. The effect of external measurements $z(t)$ is to superimpose a reactive effect on the diffusion in which the density tends to peak in the vicinity of measurements.

In the simple Gaussian case, the diffusion is purely linear and the density function evolves as a Gaussian pulse that translates, spreads and is reinforced, remaining Gaussian throughout. The Kalman filter describes analytically exactly this process. In clutter, however, when measurements have a non-Gaussian, multi-modal conditional distribution, the evolving density requires a more general representation. This leads to a powerful new approach to tracking, developed below, in which a sparse representation of the density for $x(t)$ is carried forward in time. No mean position or variance is computed explicitly, though they and other properties can be computed at any time if desired.

2 Discrete-time propagation of state density

For computational purposes, the propagation process must be set out in terms of discrete time t. The state of the modelled object at time t is denoted x_t and its history is $\mathbf{x}_t = (x_1, x_2, \ldots, x_t)$. Similarly the set of image features at time t is z_t with history $\mathbf{z}_t = (z_1, \ldots, z_t)$. Note that no functional assumptions (linearity, Gaussianity, unimodality) are made about densities, in the general treatment,

though particular choices will be made in due course in order to demonstrate the approach.

2.1 Stochastic dynamics

A somewhat general assumption is made for the probabilistic framework that the object dynamics form a temporal Markov chain so that:

$$p(x_{t+1}|\mathbf{x}_t) = p(x_{t+1}|x_t) \tag{3}$$

— the new state is conditioned directly only on the immediately preceding state, independent of the earlier history. This still allows quite general dynamics, including stochastic difference equations of arbitrary order; we use second order models and details are given later. The dynamics are entirely determined therefore by the form of the conditional density $p(x_{t+1}|x_t)$. For instance,

$$p(x_{t+1}|x_t) = \exp -(x_{t+1} - x_t - 1)^2/2,$$

represents a one-dimensional random walk (discrete diffusion) whose step length is a standard normal variate, superimposed on a rightward drift at unit speed. Of course, for realistic problems x is multi-dimensional and the density is more complex (and, in the applications presented later, learned from training sequences).

2.2 Measurement

Observations z_t are assumed to be independent, both mutually and with respect to the dynamical process, and this is expressed probabilistically as follows:

$$p(\mathbf{z}_t, x_{t+1}|\mathbf{x}_t) = p(x_{t+1}|\mathbf{x}_t) \prod_{i=1}^{t} p(z_i|x_i). \tag{4}$$

Note that integrating over x_{t+1} implies the mutual conditional independence of observations:

$$p(\mathbf{z}_t|\mathbf{x}_t) = \prod_{i=1}^{t} p(z_i|x_i). \tag{5}$$

The observation process is therefore defined by specifying the conditional density $p(z_t|x_t)$ at each time t, and later, in computational examples, we take this to be a time-independent function $p(z|x)$. Details of the shape of this function, for applications in image-stream analysis, are given in section 4.

2.3 Propagation

Given a continuous-valued Markov chain with independent observations, the rule for propagation of conditional density $p(x_t|\mathbf{z}_t)$ over time is:

$$p(x_{t+1}|\mathbf{z}_{t+1}) = k_{t+1}\ p(z_{t+1}|x_{t+1}) p(x_{t+1}|\mathbf{z}_t) \tag{6}$$

where

$$p(x_{t+1}|\mathbf{z}_t) = \int_{x_t} p(x_{t+1}|x_t) p(x_t|\mathbf{z}_t) \tag{7}$$

and k_{t+1} is a normalisation constant that does not depend on x_{t+1}.

The propagation rule (6) should be interpreted simply as the equivalent of the Bayes rule (1) for inferring posterior state density from data, for the time-varying case. The effective prior $p(x_{t+1}|\mathbf{z}_t)$ is actually a prediction taken from the posterior $p(x_t|\mathbf{z}_t)$ from the previous time-step, onto which is superimposed one time-step from the dynamical model (Fokker-Planck drift plus diffusion as in figure 1), and this is expressed in (7). Multiplication in (6) by the conditional measurement density $p(z_{t+1}|x_{t+1})$ in the Bayesian manner then applies the reactive effect expected from measurements (figure 1).

3 The CONDENSATION algorithm

In contrast to the static case in which the prior $p(x)$ may be Gaussian, the effective prior $p(x_{t+1}|\mathbf{z}_t)$ in the dynamic case is not Gaussian when clutter is present. It has no particular known form and therefore cannot apparently be represented exactly in the algorithm. The CONDENSATION algorithm solves this problem by doing altogether without any explicit representation of the density function itself. Instead, it proceeds by generating sets of N samples from $p(x_t|\mathbf{z}_t)$ at each time-step. Each sample s_t is considered as an (s_t, π_t) pair, in which s_t is a value of x_t and π_t is a corresponding sampling probability. Suppose a particular s_t is drawn randomly from $p(x_t|\mathbf{z}_t)$ by choosing it, with probability π_t, from the set of N samples at time t. Next draw s_{t+1} randomly from $p(x_{t+1}|x_t = s_t)$, one time-step of the dynamical model, starting from $x_t = s_t$, a Gaussian density to which standard sampling methods apply. A value s_{t+1} chosen in this way is a fair sample from $p(x_{t+1}|\mathbf{z}_t)$. It can then be retained as a pair (s_{t+1}, π_{t+1}) for the N-set at time $t+1$, where $\pi_{t+1} = p(z_{t+1}|x_{t+1} = s_{t+1})$. This sampling scheme is the basis of the CONDENSATION algorithm and details are given in figure 2. In practice, random variates can be generated efficiently, using binary search, if, rather than storing probabilities π_t, we store cumulative probabilities c_t as shown in the figure. At any time t, expected values $E[f(x_t)|\mathbf{z}_t]$ of properties of the state density $p(x_t|\mathbf{z}_t)$ can be evaluated by applying the rule (2) from the factored sampling algorithm.

4 Probabilistic parameters for curve tracking

In order to apply the CONDENSATION algorithm, which is general, to the tracking of curves in image-streams, specific probability densities must be established both for the dynamics of the object and for the measurement process. As mentioned earlier, the parameters x denote a linear transformation of a B-spline curve, either an affine deformation, or some non-rigid motion. The dynamical model and learning algorithm follow established methods [6, 7]. The model is a stochastic differential equation which, in discrete time, is

$$x_{t+1} = Ax_t + B\omega_t \tag{8}$$

where A defines the deterministic component of the model and ω_t is a vector of independent standard normal random variables scaled by B so that BB^T is the

Iterate

At time-step $t+1$, construct the n^{th} of N samples as follows:

1. Generate a random number $r \in [0,1]$, uniformly distributed.
2. Find, by binary subdivision on m, the smallest m for which $c_t^{(m)} \geq r$.
3. Draw a random variate $s_{t+1}^{(n)}$ from the density $p(x_{t+1}|x_t = s_t^{(m)})$, assumed Gaussian so direct sampling is possible.

Store samples $n = 1,..,N$ as $(s_{t+1}^{(n)}, \pi_{t+1}^{(n)}, c_{t+1}^{(n)})$ where

$$c_{t+1}^{(0)} = 0$$
$$\pi_{t+1}^{(n)} = p(z_{t+1}|x_{t+1} = s_{t+1}^{(n)})$$
$$c_{t+1}^{(n)} = c_{t+1}^{(n-1)} + \pi_{t+1}^{(n)}$$

and then normalise by dividing all cumulative probabilities $c_{t+1}^{(n)}$ by $c_{t+1}^{(N)}$, i.e. so that $c_{t+1}^{(N)} = 1$.

If required, mean properties can be estimated at any time t as

$$E[f(x)|\mathbf{z}_t] \approx \sum_{n=1}^{N} \pi_t^{(n)} f(s_t^{(n)}).$$

For example, if the mean configuration $\hat{x}$ is required for graphical display, the above rule is used with $f(x) = x$.

Fig. 2. *The* CONDENSATION *algorithm.*

process noise covariance. The model can clearly be re-expressed as a temporal Markov chain as follows:

$$p(x_{t+1}|x_t) = \exp -\frac{1}{2}\|B^{-1}(x_{t+1} - Ax_t))\|^2. \tag{9}$$

In practice, we use second order models, where x_t, A and B are replaced by

$$\begin{pmatrix} x_t \\ x_{t+1} \end{pmatrix}, \quad \begin{pmatrix} 0 & I \\ A_0 & A_1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & 0 \\ 0 & B \end{pmatrix}$$

respectively. Coefficients are learned from sequences of images. An untrained tracker is used to follow training motions against a relatively clutter-free background. The tracked sequence in the form $(x_1, x_2, \ldots)$ is then analysed [6, 7] by Maximum Likelihood Estimation to generate estimates of A_0, A_1 and B, thus defining the model for use by the CONDENSATION algorithm. A set of sample values for time-step $t = 0$ must be supplied to initialise the algorithm. If the prior density $p(x_0)$ is Gaussian, direct sampling may be used for initialisation, otherwise it is possible simply to allow the density to settle to a steady state $p(x_\infty)$ in the absence of object measurements.

4.1 Observations

The measurement process defined by $p(z_t|x_t)$ is assumed here to be stationary in time (though the CONDENSATION algorithm does not require this) so a static function $p(z|x)$ is to be specified. As yet we have no capability to estimate it from data, though that would be ideal, so some reasonable assumptions must be made.

Measurements z arising from a curve x are image-edge fragments obtained by edge-detection along curve normals. We assume that noise and distortions in imaging z are local, so in order to determine $p(z|x)$ it is necessary only to examine image pixels near the image curve which we denote (with mild abuse of notation) $x(s), 0 \leq s \leq 1$. The corresponding measurement sequence is then denoted $z(s)$, where $z(s)$ for each s is the detected edge on the normal at $x(s)$ that lies closest to the curve x. To allow for measurement failures and clutter, the measurement density is modelled as a robust statistic, a truncated Gaussian:

$$p(z|x) = \exp\left\{-\frac{1}{2\sigma^2}\int_0^1 \phi(s)ds\right\} \tag{10}$$

where

$$\phi(s) = \begin{cases} |x(s) - z(s)|^2 \text{ if } |x(s) - z(s)| < \delta \\ \rho \qquad\qquad \text{otherwise} \end{cases} \tag{11}$$

and ρ is a penalty constant, related to the probability of failing to find a feature, either on the curve or the background. Note that ϕ is constant at distances greater than δ from the curve, so δ acts as a maximum scale beyond which it is unnecessary to search for features. In practice, of course, the integral is approximated as a sum over discrete sample intervals of s.

5 Applying the CONDENSATION algorithm to video-streams

5.1 Tracking a multi-modal distribution

In order to test the CONDENSATION algorithm's ability to represent a multi-modal distribution, we collected a 70 frame (2.8 second) sequence showing a cluttered room with three people in it, facing the camera. The person initially on the right of the image moves to the left, in front of the other two. A template was drawn, using an interactive package, to fit around the head and shoulders of a person, and we constructed an affine space of deformations of that template. A motion model was learned by tracking a single person walking around the room; background subtraction was necessary to ensure accurate tracking past the clutter. Results of running the CONDENSATION algorithm are shown in figure 3. Since the feature of interest is primarily x translation, only the distribution of the parameter corresponding to x coordinate has been plotted, however it is clear that the people are of slightly different sizes and heights, and this is modelled in the full distribution. No background subtraction or other preprocessing is used; the input is the raw video stream. Initialisation is performed simply by iterating the stochastic model in the absence of measurements, and it can be seen that

this corresponds to a roughly Gaussian distribution on x coordinate at the first time-step. The distribution rapidly collapses onto the three peaks present in the image, and tracks them correctly, despite temporary difficulties while the people occlude each other. The time-step used for tracking is frame rate (40 ms) since the motion is fairly slow; in the figure, distributions are plotted only every 80 ms for clarity. The stationary person on the left has the highest peak in the distribution; this is to be expected since he is standing against a clutter-free background, and so his outline is consistently detectable. The experiment was run using a distribution of $N = 1000$ samples.

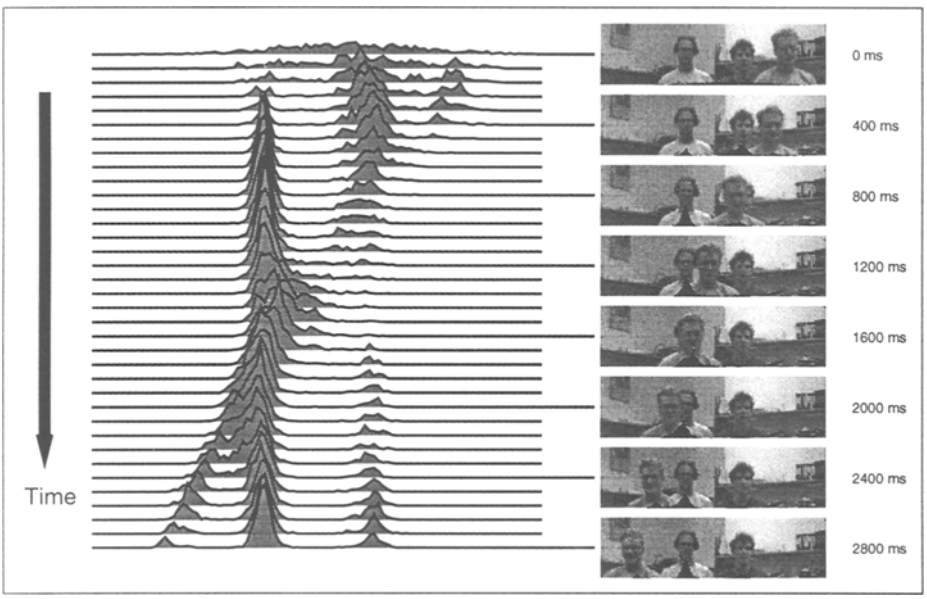

Fig. 3. ***Tracking a multi-modal distribution.*** *A histogram of the horizontal translation component of the distribution is plotted against time. The initial distribution is roughly Gaussian, but the three peaks are rapidly detected and tracked as one person walks in front of the other two.*

5.2 Tracking rapid motions through clutter

Next we collected a 500 field (10 second) sequence showing a girl dancing vigorously to a Scottish reel against a highly cluttered background, in order to test the CONDENSATION algorithm's agility when presented with rapid motions. We drew a head-shaped template and constructed an affine space to represent its allowable deformations. We also collected a training sequence of dancing against a mostly uncluttered background, from which we trained a motion model for use when the CONDENSATION tracker was applied to test data including clutter.

Figure 4 shows some stills from the clutter sequence, with tracked head positions from preceding fields overlaid to indicate motion. The contours are plotted

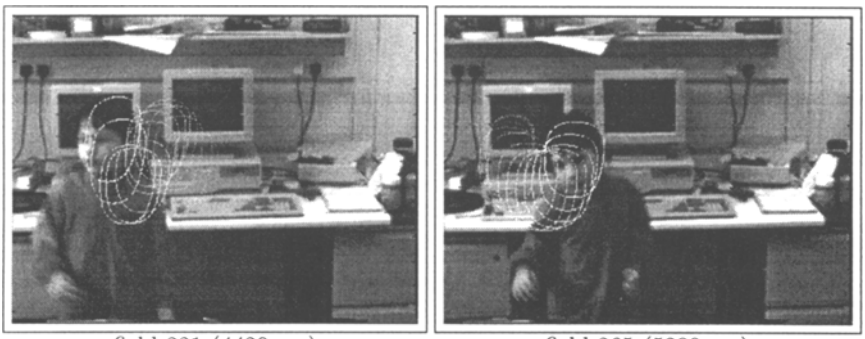

Fig. 4. ***Maintaining tracker agility in clutter.*** *A sequence of 500 fields (10 seconds) was captured showing a dancer executing rapid motions against a cluttered background. The dancer's head was then tracked through the sequence. Representative fields are shown, with preceding tracked head positions to give an indication of the motion. The tracked positions are shown at 40 ms intervals. The distribution consists of* $N = 100$ *samples.*

at 40 ms intervals. The model parameters are estimated by the mean of the distribution at each time-step. The distribution consists of $N = 100$ samples. The distribution was initialised by hand near the dancer's position in the first field, as 100 samples do not sweep out enough of the prior to locate the initial peak reliably. It would be equally feasible to begin with a larger number of samples in the first field, and reduce the size of the distribution when the dancer had been found (this technique was used in section 5.3).

Figure 5 shows the centroid of the head position estimate as tracked by both the CONDENSATION algorithm and a Kalman filter. The CONDENSATION tracker correctly estimated the head position throughout the sequence, but after about 40 fields (0.80 s), the Kalman filter was distracted by clutter, never to recover.

Although it is expected that the posterior distribution will be largely unimodal throughout the sequence, since there is only one dancer, figure 6 illustrates the point that it is still important for robustness that the tracker is able to represent distributions with several peaks. After 920 ms there are two distinct peaks, one caused by clutter, and one corresponding to the dancer's head. At this point the clutter peak has higher posterior probability, and a unimodal tracker like the Kalman filter would discard the information in the second peak, rendering it unable to recover; however the CONDENSATION algorithm does recover, and the dancer's true position is again localised after 960 ms.

5.3 Tracking complex jointed objects

The preceding sequences show motion taking place in a model space of at most 4 dimensions, so in order to investigate tracking performance in higher dimensions, we collected a 500 field (10 second) sequence of a hand translating, rotating,

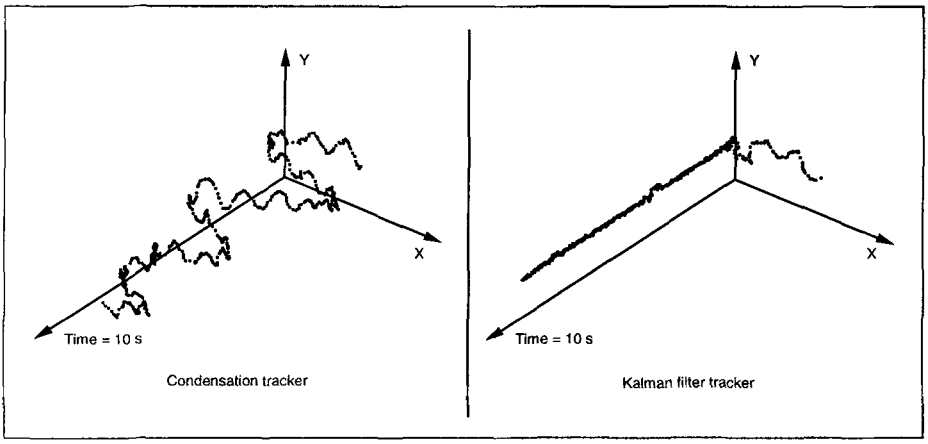

Fig. 5. ***The Condensation tracker succeeds where a Kalman filter fails.*** *The centroid of the state estimate for the sequence shown in figure 4 is plotted against time for the entire 500 field sequence, as tracked by first the* CONDENSATION *tracker, then a Kalman filter tracker. The* CONDENSATION *algorithm correctly estimates the head position throughout the sequence. The Kalman filter initially tracks correctly, but is rapidly distracted by a clutter feature and never recovers.*

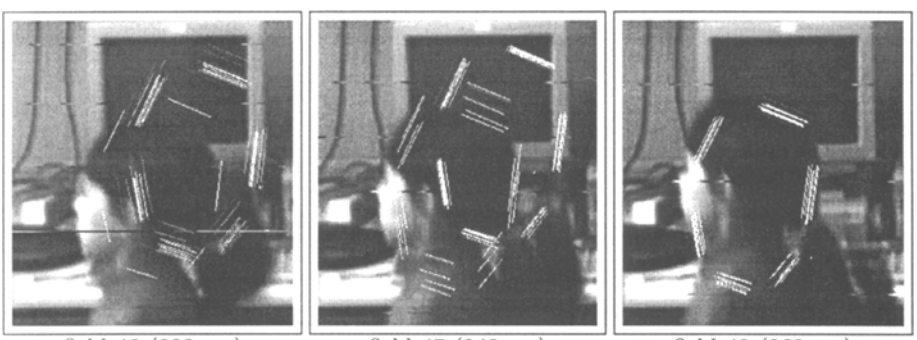

Fig. 6. ***Recovering from tracking failure.*** *Detail from 3 fields of the sequence illustrated in figure 4. Each sample from the distribution is plotted on the image, with intensity scaled to indicate its posterior probability. Most of the samples, from a distribution of $N = 100$, have too low a probability to be visible. In field 46 the distribution has split into two distinct peaks, the larger attracted to background clutter. The distribution converges on the dancer in field 48.*

and flexing its fingers independently, over a highly cluttered desk scene. We constructed a twelve degree of freedom shape variation model and an accompanying motion model with the help of a Kalman filter tracking in real time against a plain white background, using signed edges to help to disambiguate the finger boundaries.

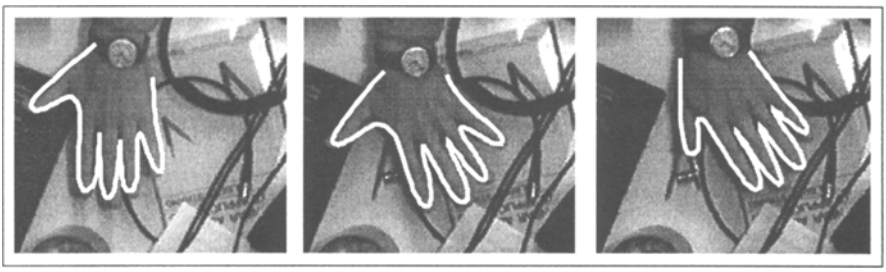

Fig. 7. ***Tracking a flexing hand across a cluttered desk.*** *Representative stills from a 500 field (10 second) sequence of a hand moving over a highly cluttered desk scene. The fingers and thumb flex independently, and the hand translates and rotates. The distribution consists of $N = 500$ samples except for the first 4 fields, when it decreases from 1500 samples to aid initialisation. The distribution is initialised automatically by iterating on the motion model in the absence of measurements.*

Figure 7 shows detail of a series of images from the tracked 500 field sequence. The distribution is initialised automatically by iterating the motion model in the absence of measurements. The initialisation is performed using $N = 1500$ samples, but N is dropped gradually to 500 over the first 4 fields, and the rest of the sequence is tracked using $N = 500$. Occasionally one section of the contour locks onto a shadow or a finger becomes slightly misaligned, but the system always recovers. Figure 8 shows just how severe the clutter problem is — the hand is immersed in a dense field of edges. The CONDENSATION algorithm succeeds in tracking the hand despite the confusion of input data.

6 Conclusions

Tracking in clutter is hard because of the essential multi-modality of the conditional measurement density $p(z|x)$. In the case of curve tracking, multiple-hypothesis tracking is inapplicable and a new approach is needed. The CONDENSATION algorithm is a fusion of the statistical factored sampling algorithm for static, non-Gaussian problems with a stochastic differential equation model for object motion. The result is an algorithm for tracking rigid and non-rigid motion which has been demonstrated to be far more effective in clutter than comparable Kalman filters. Performance of the CONDENSATION algorithm improves as the sample size parameter N increases, but computational complexity is $O(N \log N)$. Impressive results have been demonstrated for models with 4 to 12 degrees of freedom, even when $N = 100$. Performance in several cases was

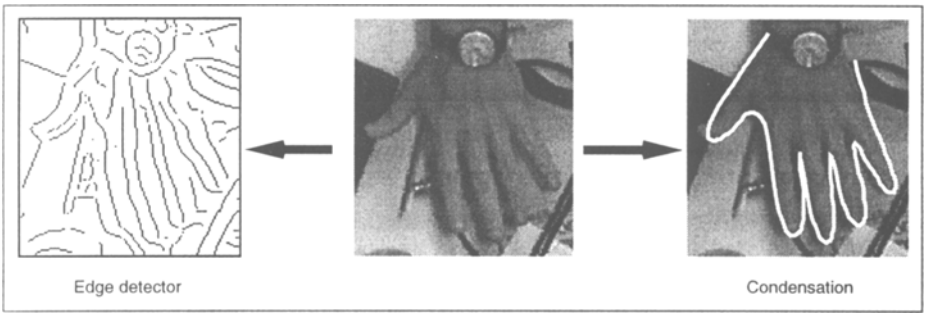

Fig. 8. ***Localising the hand in a dense edge map.*** *Detail of a field from the hand sequence. The result of running a directional Gaussian edge detector shows that there are many clutter edges present to distract the system. The* CONDENSATION *algorithm succeeds in tracking the hand through this clutter.*

improved still further with increased $N = 1000$. The system currently runs with $N = 50$ in real-time (25Hz) on a desk-top graphics workstation (Indy R4400SC, 200 MHz).

The authors would like to acknowledge the support of the EPSRC. They are also grateful for discussions with Roger Brockett, Brian Ripley, David Reynard, Simon Rowe and Andrew Wildenberg, and for experimental assistance from Sarah Blake.

References

1. K. J. Astrom. *Introduction to stochastic control theory.* Academic Press, 1970.
2. Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.
3. R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann, 1987.
4. A. Baumberg and D. Hogg. Generating spatiotemporal models from examples. In *Proc. BMVC*, 413–422, 1995.
5. A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. Journal of Computer Vision*, 11(2):127–145, 1993.
6. A. Blake and M.A. Isard. 3D position, attitude and shape input using video tracking of hands and lips. In *Proc. Siggraph*, 185–192. ACM, 1994.
7. A. Blake, M.A. Isard, and D. Reynard. Learning to track the visual motion of contours. *Artificial Intelligence*, 78:101–134, 1995.
8. R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. 3rd Int. Conf. on Computer Vision*, 616–625, 1990.
9. T.F. Cootes, C.J. Taylor, A. Lanitis, D.H. Cooper, and J. Graham. Building and using flexible models incorporating grey-level information. In *Proc. 4th Int. Conf. on Computer Vision*, 242–246, 1993.

10. M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–95, 1981.
11. Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
12. D.B. Gennery. Visual tracking of known three-dimensional objects. *Int. Journal of Computer Vision*, 7:3:243–270, 1992.
13. C.G. Goodwin and K.S. Sin. *Adaptive filtering prediction and control.* Prentice-Hall, 1984.
14. U. Grenander, Y. Chow, and D. M. Keenan. *HANDS. A Pattern Theoretical Study of Biological Shapes.* Springer-Verlag. New York, 1991.
15. U. Grenander and M.I. Miller. Representations of knowledge in complex systems (with discussion). *J. Roy. Stat. Soc. B.*, 56:549–603, 1993.
16. C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, 59–74. MIT, 1992.
17. D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.
18. D.P. Huttenlocher, J.J. Noh, and W.J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, 93–101, 1993.
19. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, 259–268, 1987.
20. D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Int. Journal of Computer Vision*, 8(2):113–122, 1992.
21. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1988.
22. B. Rao. Data association methods for tracking systems. In A. Blake and A. Yuille, editors, *Active Vision*, 91–105. MIT, 1992.
23. J.M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In J-O. Eklundh, editor, *Proc. 3rd European Conference on Computer Vision*, 35–46. Springer–Verlag, 1994.
24. B.D. Ripley and A.L. Sutherland. Finding spiral structures in images of galaxies. *Phil. Trans. R. Soc. Lond. A.*, 332(1627):477–485, 1990.
25. G. Storvik. A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(10):976–986, 1994.
26. G.D. Sullivan. Visual interpretation of known objects in constrained scenes. *Phil. Trans. R. Soc. Lond. B.*, 337:361–370, 1992.
27. D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: deformable superquadrics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7), 1991.
28. S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991.
29. A. Yuille and P. Hallinan. Deformable templates. In A. Blake and A. Yuille, editors, *Active Vision*, 20–38. MIT, 1992.