Tracking Many Objects Using Subordinated CONDENSATION

David Tweed and Andrew Calway
Computer Science Department, Bristol University, UK
{tweed,andrew}@cs.bris.ac.uk

Abstract

We describe a novel extension to the CONDENSATION algorithm for tracking multiple objects of the same type. Previous extensions for multiple object tracking do not scale effectively to large numbers of objects. The new approach – subordinated CONDENSATION – deals effectively with arbitrary numbers of objects in an efficient manner, providing a robust means of tracking individual objects across heavily populated and cluttered scenes. The key innovation is the introduction of bindings (*subordination*) amongst particles which enables multiple occlusions to be handled in a natural way within the standard CONDENSATION framework. The effectiveness of the approach is demonstrated by tracking multiple animals of the same species in cluttered wildlife footage.

1 Introduction

Tracking objects over the course of an image sequence is one of the basic tasks in Computer Vision. The resulting trajectory can be either of interest in its own right or used as the foundation for a higher level analysis. Applications include surveillance and recognition systems [11, 6, 9] and advanced human-computer interaction[8]. However, designing robust tracking algorithms is difficult, requiring mechanisms to deal with issues such as weak distinguishing image features, background clutter, erratic and discontinuous motion, multiple and occluding objects, and many other problems.

Tracking algorithms use two sources of information to tackle these issues: a model of the dynamical behaviour of the object being tracked; and a model of its appearance within an image. The former can be obtained either from simplifying assumptions or by using *a priori* information obtained, e.g., that obtained from exemplars, whilst the latter will involve a combination of assumptions about the imaging geometry and the 3-D structure of the objects being tracked. The tracking process then involves finding a model configuration which balances consistency of the dynamical model against image support within the individual frames, leading to robust estimates of the object position.

The classic formulation of this approach is via the Kalman filter, although its assumption of a unimodal probability distribution and standard Gaussian interpretation means that its performance is limited in difficult tracking scenarios. A more robust approach and one that has found considerable success recently is particle filtering [3] and in particular the CONDENSATION algorithm [2]. These methods are based on creating an approximation to the full probability distribution of the object's configuration over all

BMVC 2002 doi:10.5244/C.16.26

possible locations using sampling. This enables them to retain multimodal probability distributions, making them robust to temporarily ambiguous image support and hence able to maintain tracking even in the presence of clutter and occlusions. They also have the added advantage of being simple to implement, with low computational cost and clear potential for real-time use (cf. applications in [8, 6]). Equally important is that they also provide a coherent Bayesian formulation within which to incorporate *a priori* knowledge about the dynamic behaviour of the object being tracked, facilitating the use of densities learnt from exemplars and hence improving the robustness of the tracking.

This paper is concerned with using particle filtering for a particularly challenging task, namely that of tracking many objects of the same type in a cluttered sequence. Our application area is robust tracking of animals through wildlife footage as a precursor to automated indexing and archiving [12]. As the sequence in Fig. 3 illustrates, the footage typically contains poor distinguishing image features and significant clutter caused by herding and flocking animals. Multiple object tracking within such sequences is difficult since the appearance model uses the same types of image features to support each object, and thus there is plenty of scope for ambiguity in the measurements as objects start to become occluded or pass by each other. Its sampling framework and ability to retain multiple modes would suggest that particle filtering should provide a means of addressing such problems, although to date this has not been fully exploited. Previous extensions to multiple objects, whilst having theoretical rigour, have been designed to deal with at most two or three objects and do not readily scale to large numbers of objects [7, 6, 5].

To address this, we have developed a novel extension of CONDENSATION which deals naturally and efficiently with tracking an arbitrary number of similar objects. The key innovation is the use of an explicit representation of occlusion amongst particles within the sampling framework which allows particles to occlude each other. This is implemented by introducing asymmetric bindings between pairs of particles as and when they occlude, with the occluded particle becoming *subordinate* to the occluder. In contrast to augmenting the particle state by background and foreground objects as in [7], this maintains a constant particle size irrespective of the number of objects being tracked. Although this means that we lose some of the rigorous probabilistic interpretation, we gain the advantage that the method scales naturally to many objects, with the only overhead being the need to use larger sample sets (as would be expected).

In the next section we describe the new approach and then in Section 3 give details of using it within the CONDENSATION framework for tracking animals in wildlife footage described in [12]. Results on the sequence in Fig. 3 illustrate that the approach is robust to multiple occlusions and provides an effective method for tracking individual birds across highly populated and cluttered scenes.

2 Particle Filtering and CONDENSATION

As our extensions to CONDENSATION are motivated partly by the desire for efficient implementation we review the basics of particle filtering and CONDENSATION first; readers are referred to [2, 8, 6] for further information.

The aim of *particle filtering* is to construct a probability density $p(X_{t+1})$ over the state of a temporal stochastic process X_t over a multidimensional state space X. Density $p(X_{t+1})$ is conditioned upon the density in the previous frame $p(X_t)$, a stochastic state evolution model $\mathcal{P}$ and observations of data y_t affected by the value of X_t. (Thus for tracking X_t is the location of the object and y_t is the image at time t). Particle filtering uses a weighted sample set $\{(\mathbf{s}_i, \pi_i)\}_{i=1}^{N}$ to represent the distribution $\sum_{i=1}^{N} \pi_i \delta(\mathbf{s}_i)$, where $\delta(\cdot)$ is a Dirac delta function and $\mathbf{s}_i$ is a point in the state space X (cf. [2].) This distribution can be sampled from effectively and can be shown [2] to become a more accurate representation of the true underlying distribution as the number of particles N increases. Since this includes multi-modal distributions particle filters can cope effectively with temporarily ambiguous image support.

Particle filtering deals extensively with particles with associated (positive) weights. To simplify the exposition throughout the rest of the paper, we define '*choose proportionately k* particles from $\mathcal{S}$' to mean 'choose (with replacement) k particles from set $\mathcal{S}$, with the probability of selecting each item being proportional to its weight'. Thus, the basic particle filtering algorithm is as follows:

1.Sample $p(X_t|y_t \ldots y_0)$ *Choose proportionately N* samples from $\{(\mathbf{s}_i, \pi_i)\}$ to get $\{\mathbf{s}'_i\}$.

2.Evolve Apply the *stochastic* state-evolution model $\mathcal{P}$ to particles in $\{\mathbf{s}'_i\}$ to get $\{\mathbf{s}''_i\}$.

3.Measure support to get $p(X_{t+1}|y_{t+1} \ldots y_0)$ For each particle in $\{\mathbf{s}''_i\}$ measure the support π''_i from frame $t{+}1$ to get the weights for new density $\{(\mathbf{s}''_i, \pi''_i)\}$.

Whilst step (1) can be performed in $O(N \log N)$ time using uniformly distributed random numbers and binary search, in practice an $O(N)$ deterministic approximation is used. Conceptually this involves assigning the particles consecutive subintervals of $[0,1)$ with length proportional to their weight and the particle owning the interval containing the point i/N (for $0 \leq i < N$) is taken as the basis for a new sample $\mathbf{s}_i$ in p_t. In principle this could give poorer results than true sampling; in practice, as almost all the mass of the distribution should be concentrated in a few particles, there is little difference.

Finally, when dealing with complex models amenable to decomposition, the algorithm can be extended to *partitioned sampling* [8]. In the simplest form, when the configuration space X is structred as $X_{top} \times X_{bottom}$ then the algorithm can be applied twice for each frame, first to the distribution of X_{top} and then sampling from this distribution $p(X_{top})$ to find the distribution of X_{bottom}. For example, in hand tracking the configuration of a hand can be represented as the position of the body of the hand – X_{top} – and the positions of the fingers relative to the it – X_{bottom}. Informally this is advantageous because to 'find' the supported configuration $x = (x_{top}, x_{bottom})$ standard CONDENSATION's stochastic evolution step must produce x_{top} and x_{bottom} *in the same particle*, whereas partitioned sampling first 'finds' x_{top} (in the sense of a distribution which is peaked over likely values), then 'finds' the x_{bottom} value given the correct x_{top} value. Thus the number of particles needed for good tracking changes from $O(|X_{top}| \times |X_{bottom}|)$ to $O(|X_{top}| + |X_{bottom}|)$. (We gloss over some details here – [7] has a full discussion.)

2.1 Subordinated CONDENSATION

In [7] a *probabilsitic exclusion principle* for tracking multiple occluding objects was introduced by having each particle containing two objects: a (potentially occluding) foreground object and a (potentially occluded) background object, where 'occlusion' is taken to mean 'partial or total occlusion'. The principle states that a 'feature' in the image can

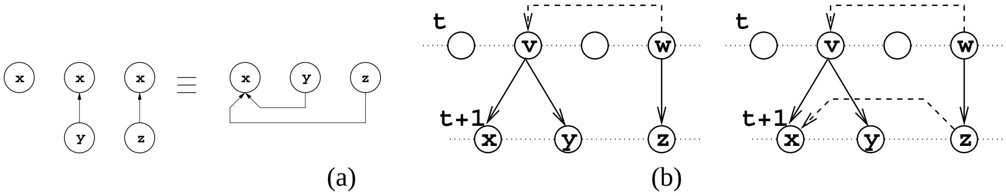


Figure 1: Representing multi-object particles using subordinated particles: (a) basic representation scheme; (b) temporal propagation of subordination links

be used to support at most one of the two objects. By factorising the state density, partitioned sampling can then be used with the first level being the foreground object and the second level the background object. This sequential sampling makes it easy to prevent any observations which correspond to the foreground configuration from being used for the background object. However augmenting the particle states in this way requires a separate layer from each new particle so that it does not scale to the large number of objects we want to consider. Equally it is potentially less compact when the objects are not overlapping as two configurations differing only in their relative depths describe the same underlying situation. Our aim is to produce an algorithm which incorporates the notion of ‘observation exclusivity’ but which flexibly deals with the spectrum from no overlapping objects to many overlapping objects.

The fundamental idea is to keep a single set of particles to represent the state density but to allow the possibility of one particle occluding another. If a particle A *is subordinated to* a particle B then A is behind B. The obvious way to do this is by ‘binding together’ some particles, in effect running part of the density using standard CONDENSATION and part using the exclusion scheme in [7]. However, consideration shows that the occlusion relation is asymmetric: the behaviour of the occluded particle depends upon the precise details of the occluder whereas the occluders’ behaviour is indistinguishable from a normal particles’. Thus we can economise on storage and more importantly evaluation of image support by giving each particle a (possibly null) *subordination link* to an occluding particle (as shown in Fig. 1). It turns out that this scheme has very few differences from the standard CONDENSATION algorithm, the two key ones being that subordination links must be propagated over time and the observation support of occluded particles must be calculated accounting for the occluder.

Propagating subordination links over time is done (using an assumption that the actual behaviour of the occluder and occluded are independent) using a probabilistic scheme illustrated by Fig. 1b: With particle w subordinated to v, the descendants of both are produced in the precisely the standard way by sampling. Each descendant of the occluded particle is then linked (independently) to a descendant of the occluder by choosing proportionately from the set of descendants of the occluder. In terms of the lefthand illustration in Fig. 1b, z (descendant of w) has a link formed with v’s descendant x with probability $\pi_x/(\pi_x+\pi_y)$ and to y otherwise. Thus, if x is chosen the link from z to x is made as in the righthand part of Fig. 1.

These considerations lead to the algorithm in Fig. 2. There are three subtle points here: (i) Using the deterministic sampling approximation and keeping the samples in order, all of the occluders of a particle have all been evaluated before the particle they are occluding. (ii) As written it is possible for a particle which produces descendants in step 1a to have an

1. Generate the set of derived particles $\mathcal{S}_{t+1}=\{\mathbf{s}_1,\ldots,\mathbf{s}_N\}$ for time $t+1$ from the density at t using the deterministic approximation. As each $\mathbf{s}_i$ is generated:
    - (a) **If** $\mathbf{s}_i$'s parent particle is occluded by $\mathbf{s}^t_{occl}$ **then**
        - *choose proportionately* a descendant $\mathbf{s}^{t+1}_{occl}$ of $\mathbf{s}^t_{occl}$
        - mark the area $\mathbf{s}^{t+1}_{occl}$ occupies in *occluded_bitmap* and jump to step 1c.
    - (b) **Otherwise**, **if** random fraction $<$ probability of entering occlusion **then** add $\mathbf{s}_i$ to the *newly_occluded_particles_list* and start at step 1a for next particle.
    - (c) **Otherwise**, measure the image support π_i for $\mathbf{s}_i$ (using *occluded_bitmap*).
    - (d) **If** $\mathbf{s}^{t+1}_{occl}$ no longer occludes $\mathbf{s}_i$ **then** remove the subordination link.
2. For each sample $\mathbf{s}_i$ on *newly_occluded_particles_list*:
    - (a) Determine (from *all* current particles in $\mathcal{S}_{t+1}$) the set of potential occluders.
    - (b) *Choose proportionately* an occluding particle $\mathbf{s}^{t+1}_{occl}$ and mark the area $\mathbf{s}^{t+1}_{occl}$ occupies in *occluded_bitmap*.
    - (c) Measure image support π_i for $\mathbf{s}_i$ (again taking account of *occluded_bitmap*).

Figure 2: Overview of subordinated CONDENSATION algorithm

occluder which does *not* produce descendants (because it has a very low weight). Again, for deterministic sampling we can avoid this by boosting such weights by the minimum amount required to ensure they will generate one descendant in the iteration. (iii) In step 2a to ensure that a particle representing an object does not become subordinated to another particle representing the same object, particles with a common ancestor within the last N frames are excluded from the set of potential occluders. This is an effective compromise limiting the amount of stored history needed.

Note that in the algorithm steps 1a, 1d and 2a require a *task specific* technique for ascertaining if the one of the two objects $\mathbf{s}_i$ and $\mathbf{s}_j$ must be occluding the other, whilst steps 1a and 2b require finding all the image pixels belonging to an object $\mathbf{s}_i$. We describe the algorithms we used in our animal tracking experiments in Section 3.1.

2.2 Sample impoverishment and probability decay

For multiple object tracking however there are some final issue, stemming from our use of the state density to represent the configuration of multiple objects rather than a single object (with further details in [12]).

Sample impoverishment. As the state evolution model is stochastic rather than deterministic, even given the 'correct' previous state it may take many samples to produce the 'correct' current state. Thus the number of new particles derived from a previous state is proportional to how much the tracker explores the possiblities. The basic CONDENSATION algorithm generates derived particles proportional to the current support, so a mode tends to be lost as its weight falls below the noise inherent in the evolution model. This can be avoided by *importance sampling* [8] which allows additional samples to be derived in certain areas (motivated by some other source of information). Here the number of samples derived from $\mathbf{s}_i$ is proportional to an *importance function* f but then using $\pi_i/f(\mathbf{s}_i)$ as the weight of the resulting samples. However the greater the difference between π_i and $f(\mathbf{s}_i)$ the lower the weight of the final particle even when the image observations give strong support, so this technique can only be used to adjust the sampling behaviour by a limited amount.

Probability decay. We measure the support for a particle **s** using a fixed observation model $\mathcal{O}$. This is inevitably a simplified model which, whilst giving high support for all objects, will have unpredictable variations between objects. Consider the idealised case of objects A and B which *consistently* give reponses of p and q respectively. Then after t time-steps the ratio of the state density weight for B compared to that of A will be approximately$(q/p)^t$, i.e., it has decayed geometrically. This issue is difficult to deal with in a standard numerical way since by definition the degree to which an object matches the learned model is *a priori* unknown. Rather, we want to deal with it by periodically ‘normalizing’ the confidence of any hypotheses which are sufficiently supported.

We deal with these two issues in an integrated way based upon finding clusters within the state density and using those to construct a Voronoi tesselation [10] based upon these cluster centres. For these purposes only the 2-D positions of the reference point are used. Within each of these cells the distribution is describing primarily one object. Thus, we use the following to avoid sample impoverishment and probability decay using the following steps respectively:

- Every step, build an importance function which results in equal numbers of samples being taken in each Voronoi cell.
- Every N steps rescale the weights in each cell so that the peak weight is 1.

Empirically we have found that $N = 5$ works well. This intuitive scheme needs to be modified for the case of subordinated particles, and we do this in a simplistic way of treating each ‘depth-level’ independently, i.e., by computing a separate Voronoi diagram for each occupied depth level.

3 Models for animal tracking

In this section we briefly describe our representation of animals used within particles and then describe our implementation of observation exclusivity for this representation. Experiments using subordinated CONDENSATION for tracking birds in the sequence shown in Fig. 3 are then presented.

State representation. Our basis is a set of 2-D points linked together to form a‘skeleton’ as shown for a bird in Fig. 4a–b. (Note the skeleton models image-plane appearance and *not necessarily* the anatomical skeleton.) It was manually chosen so that the configuration of the animal away from the nodes is well approximated by the links between them. For example, between the joint in the middle of the wing and the end of the wing the line between them follows the bird; this would not be the case if the middle of wing joint were removed and the end of the wing linked directly to the node on the side of the body.

The nodes on the skeleton were then divided by hand into suitable levels for partitioned sampling. Thus, Fig. 4b shows that the midpoint of the body is sampled over first,

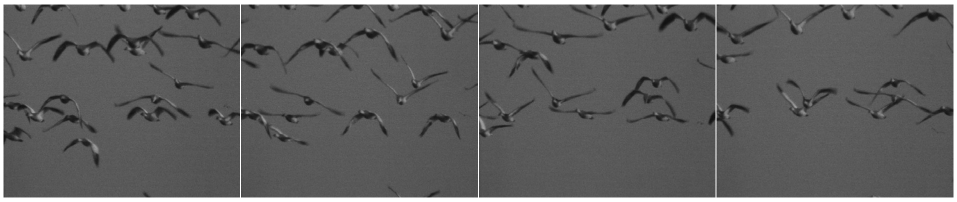
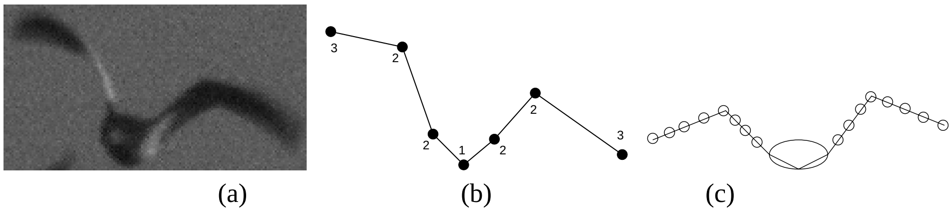


Figure 3: Frames 6, 18, 30 & 42 from a sequence of birds flying against an orange sky



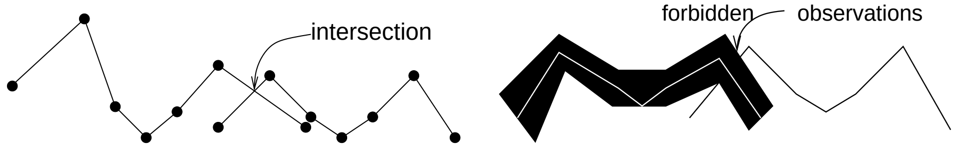
(a) (b) (c)

Figure 4: Representing animals: (a) original bird; (b) points representing (c) observation model consisting of ellipse for body and evenly spaced points along wings

then the positions of the upper halves of the wings, and finally the tips of the wings. This strategy of sampling over dependent points after sampling over those they depend upon is effective in reducing the total number of samples required, but does require manual analysis to determine it. This is not a drawback as our application is dealing with *potentially* previously unseen animals and so we require a minimal level of manual analysis to collect data for learning the motion model (described next).

Stochastic motion model. We present only an overview of the motion model for space reasons; see [12] for more details. We assume that (i) all the animals are moving in the same way and (ii) the motion can be split into a global position change and a periodic relative motion of the body parts. As we are tracking many animals over potentially hundreds of frames it is reasonable to use manual markup of points to be tracked on *one* animal over approximately two cycles of its motion to learn a global motion model which is then used for all the animals. We assume 'constant velocity plus noise' for the position of a reference point on the animal – the midpoint of the underside in the case of the bird in fig 4b – since the camera is often jerkily panning to keep the animals framed in the shot and a separate model to predict the positions of the points on the skeleton relative to the reference point. The limb model essentially treats the position in motion cycle as a *latent* variable which increments cyclically each frame and constructs a 'body-configuration to body-configuration' predictor at each point in the cycle.

Instance and model coordinates. There are two kinds of inaccuracy in the 2-D limb model described above, namely the inherent noise in the motion and the much larger variations due to differences in 3-D orientation and physical size between the animal the model was learned from and the animal being tracked. Rather than use a 3-D model (which would be difficult in practice given the relative scarcity of data and the fact the birds are small objects far from the camera) we attempt to take each animals coordinates as defined in a separate *instance coordinate system* and the single global motion model



(a) (b)

Figure 5: Schematic of methods used for (a) deciding if occlusion must be occurring and (b) deciding which image features cannot be used by the occluded object

as defined in a *model coordinate system*. Then if we attach to each animal a warping f from instance coordinates to model coordinates, we can perform prediction by mapping the state into model coordinates, applying the model and mapping the prediction back into instance coordinates. We use the simple warping

$$(x_i,y_i) \overset{f}{\longleftrightarrow} (\bar{x}_i,\bar{y}_i)=((x_i\cos\theta-y_i\sin\theta)/x_{scale},(x_i\sin\theta+y_i\cos\theta)/y_{scale}) \tag{1}$$

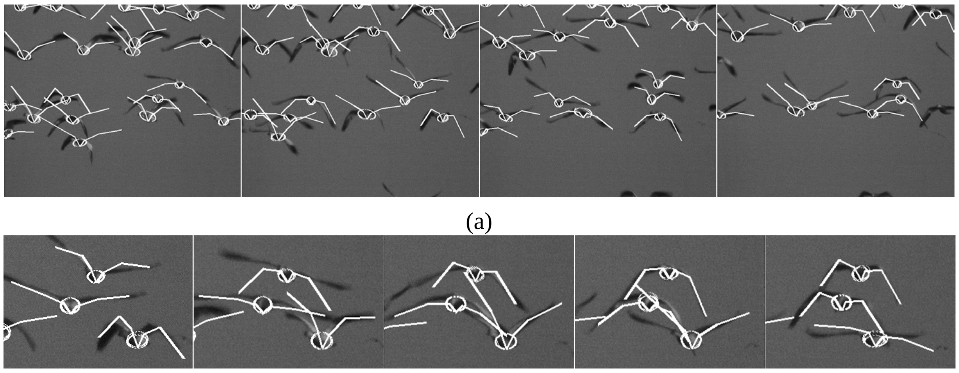
with two scale factors as different sized animals tend to scale differently in vertical and non-vertical directions, and θ permitting physical rotation. These easily interpretable parameters can thus be restricted to lie in ‘plausible’ regions (e.g., birds can be allowed to fly at an angle of up to 45° whilst prohibiting unrealistic configurations such as flying upside down), and a least squares solution from $\{(x_i,y_i)\}$ and $\{(\bar{x}_i,\bar{y}_i)\}$ can be found analytically.

Measuring image support. For the initial experiments described in Section 4 we used one crudely hand-segmented frame to learn a bird/background likelihood model based upon pixel RGB values, modelling each class with a 6-component Gaussian mixture model [1]. The first level of the sampling should strongly localise the overall location of the bird using the large body. We do this by finding (via hill-climbing) the ellipse through the main point of the bird which maximises the sum of the log likelihoods of pixels inside the ellipse belonging to the bird and outside the ellipse belonging to the background – where the initial values are given by static ‘per-bird’ parameters – and using this as the observation weight. The two lower levels progressively localise the wings, so their support is measured by summing the log likelihood of a small number of points equally spaced along the wings being in the bird class. This is summarised in Fig. 4c.

3.1 Implementing observation exclusivity

The algorithm in Section 2.1 requires a method for determining if one particle could be occluding another and a method for preventing measurements of the support for an occluded object using features which belong to its occluder. Although these are related tasks, deciding possible occlusion is *much* more common and should therefore use a relatively inexpensive technique.

As we are assuming that the animals can be represented by a skeleton of points, we use whether the two skeletons intersect as a test for whether one of the animals must be occluding the other (as shown in Fig. 5a) since determining intersection can be performed quickly. This is a reasonable approximation for ‘thin-limbed’ animals but would be less appropriate for ‘bulky’ animals such as bears.


(a)

(b)

Figure 6: Tracking results: (a) frames 6, 18, 30 & 42 from the birds sequence; (b) subregion of frames 19, 21, 23, 25 & 27 showing an example of occlusion

To exclude foreground object features being used in background objects we simply prevent image pixels corresponding to marked ones in the bitmap being used when getting support for the background object (as shown in Fig. 5b). To approximate the set of pixels belonging to the foreground object we use a very simple technique: each point on the bird is linked to another, so that there is a well-defined normal vector to this edge at each point. The offset (in positive and negative directions) along the normal which maximises where the probability of belonging to the bird (measured using the same likelihood model as the basic observations) first drops below $1/2$ can be found. These points define a (non-convex) polygon which can be filled using standard graphics techniques [4].

4 Experiments

We demonstrate the performance of the algorithm on 50 frames from the sequence in figure 3 consisting of over 15 birds flying against an orange sky. As our elementary image feature is a bird/background likelihood model on pixel RGB values, the background is 'cluttered' in as much as many background pixels support belonging to the bird class. In addition several of the birds pass behind others during the sequence, testing the occlusion-resilience of the algorithm. The dynamical model was learned from two cycles of *one* manually marked up bird. The initial state density was generated by marking the approximate reference point of each bird in the initial frame (along with the body ellipse static parameters as in Section 3) and placing an equal number of samples in the vicinity random choices for both the position in the motion cycle and the parameters for the instance-to-model coordinate-mapping. The tracker was then run using 1200, 6000 and 6000 particles respectively on each of the three levels of the model (i.e., very roughly 400 particles per individual bird on the lower levels). Fig. 6a shows the mean configurations within the cluster at equally spaced frames throughout the sequence. Although not shown, the tracker quickly converges onto the correct position in the cycle and mapping parameters for each bird over the first five frames leading the result for frame 6. The results show the robust tracking of the birds in the sequence despite heavy occlusion between birds, an detailed example of which is shown in Fig. 6b. (Note that the unmarked bird on the right of the final image in Fig. 6a moves into shot midway through the sequence, a situation the algorithm currently does not deal with.)

Note that whilst the wings sometimes drift from the correct position, this is essentially due to the somewhat simplistic measurement model being used, and in all cases the model locks onto the wings again later in the sequence.

5 Conclusions and future work

We have presented an extension to the CONDENSATION algorithm which can be used for efficiently tracking both individual and multiple objects, based upon forming *subordination* links between particle and enforcing an observation exclusion principle on linked particles. Tracking results on the bird sequence suggest its potential as a robust and efficient method for tracking multiple objects.

One unsatisfactory aspect is the use of a relatively strong observation model in the form of an object/background likelihood model learned from the sequence. In future work we intend to find ‘image features’ better suited to the issues in wildlife footage and are amenable to being learned automatically. We will also be investigating bootstrapping of the models required for tracking, possibly by using generic models (e.g., bird, biped, quadruped, etc) from a library, deforming them to improve tracking performance. Another improvement would be to use some feedback about how well the tracker is performing to determine the number of particles to be used in the next frame.

References

[1] C M Bishop. *Neural Networks for Pattern Recognition*. OUP, 1995.

[2] A Blake and M Isard. CONDENSATION – conditional density propagation for visual tracking. *Int J Computer Vision*, 29(1):5–28, 1998.

[3] A Doucet, N de Freitas, and N Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[4] Foley, van Dam, Feiner, and Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1990.

[5] C Hue, J Le Cadre, and P Perez. Tracking multiple objects with particle filtering. Technical Report 1361, IRISA, October 2000.

[6] M Isard and J MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *ICCV*, pages 34–41, 2001.

[7] J MacCormick and A Blake. A probabilistic exclusion principle for tracking multiple objects. In *ICCV*, pages 572–578, 1999.

[8] J MacCormick and M Isard. Partitioned sampling, articulated objects and interface quality hand tracking. In *ECCV*, pages 3–19, 2000.

[9] D Ormoneit, H Sidenbladh, M J Black, and T Hastie. Learning and tracking cyclic human motion. In *NIPS*, pages 894–900, 2000.

[10] R Sedgewick. *Algorithms*. Addison-Wesley, 1992.

[11] H Tao, H S Sawhney, and R Kumar. A sampling algorithm for detecting and tracking multiple objects. In *Proc. Vision Algorithms (associated ICCV)*, 1999.

[12] D Tweed and A Calway. Tracking multiple animals in wildlife footage. In *ICPR*, 2002.