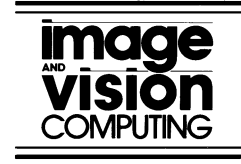




ELSEVIER

Image and Vision Computing 20 (2002) 581–594



www.elsevier.com/locate/imavis

Detecting lameness using ‘Re-sampling Condensation’ and ‘multi-stream cyclic hidden Markov models’

Derek R. Magee*, Roger D. Boyle

School of Computer Studies, Leeds, West Yorkshire LS2 9JT, UK

Received 10 June 2001; accepted 25 March 2002

Abstract

A system for the tracking and classification of livestock movements is presented. The combined ‘tracker-classifier’ scheme is based on a variant of Isard and Blakes ‘Condensation’ algorithm [Int. J. Comput. Vision (1998) 5] known as ‘Re-sampling Condensation’ in which a second set of samples is taken from each image in the input sequence based on the results of the initial Condensation sampling. This is analogous to a single iteration of a genetic algorithm and serves to incorporate image information in sample location.

Re-sampling condensation relies on the variation within the spatial (shape) model being separated into pseudo-independent components (analogous to genes). In the system, a hierarchical spatial model based on a variant of the point distribution model [Proc. Br. Mach. Vision Conf. (1992) 9] is used to model shape variation accurately. Results are presented that show this algorithm gives improved tracking performance, with no computational overhead, over Condensation alone.

Separate cyclic hidden Markov models are used to model ‘healthy’ and ‘lame’ movements within the Condensation framework in a competitive manner such that the model best representing the data will be propagated through the image sequence. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Re-sampling; Hidden Markov models; Multi-stream

1. Introduction

In recent years, there has been much interest in object tracking [1–5] and temporal modeling [6–9]. The combination of object tracking and temporal modelling gives rise to many exciting application possibilities, for example; Isard and Blake [1] use a temporal model to improve the speed and robustness of their object tracker. Wren and Pentland [8] and Davis and Bobick [9] use temporal models to classify observed human movements and in the case of the latter use this information to trigger interactive responses in a virtual environment. Johnson et al. [6] build a joint behaviour model in which a virtual human reacts in a realistic manner to observed behaviour in a limited domain. Sumpter and Bulpitt [7] use object tracking and temporal modelling to predict the behaviour of a flock of ducks or sheep for use in the control system of a robotic sheepdog.

We combine a stochastic ‘Re-sampling Condensation’ tracker with multiple ‘cyclic hidden Markov models’ to

model and classify livestock ‘behaviour’. In this paper, we take lameness as an example behaviour although it is hoped that this scheme may be extended to more complex animal behaviours such as the abnormal behaviour exhibited by animals when on heat (oestrus).

Over the last few decades, the nature of farming has changed dramatically with small, labour intensive farms being replaced by large, highly automated farms. With this change in the way, things are done comes a new concern for animal welfare. Traditionally, a skilled stock-man would look after a relatively small number of animals and have much direct contact with them on a daily basis. On the modern automated farm, a few people will look after several hundred animals and as such, there is much less direct contact. There is much interest in the farming community in automatic systems that can improve animal welfare. In particular, monitoring the health of a large group of animals is a task well suited to some sort of automatic system. Machine vision systems are a good candidate for many of these tasks as they are non-invasive (therefore safe), relatively cheap (in comparison with mechanical, electrical or chemical systems) and can augment the capacity of humans in observation tasks.

* Corresponding author. Tel.: +44-113-2336818; fax: +44-113-2335468.

Repeat (for each sample at each time step):

Select A sample is selected stochastically from the previous time step based on a ‘fitness’ measure of how well the hypothesis fitted the data at that timestep. (This forms an approximation to $f(\alpha_t|Y_t)$ with evenly weighted samples.)

Predict A single state of this sample at the next time step is predicted using a stochastic temporal model ($Pr(\alpha_{t+1}|\alpha_t)$). (This implements equation 4)

Evaluate The ‘fitness’ of the predicted sample location is evaluated at the current time step. (This is an approximation to $f(\alpha_{t+1}|Y_{t+1})$ with a weighted sample set.)

Note: Samples may be initialised randomly or according to some known probability distribution for the first frame.

Fig. 1. Practical implementation of particle filtering.

2. Background

2.1. Particle filters/the Condensation algorithm

The particle filter is a simple, yet powerful, algorithm for estimating the state of a dynamic system over time where this cannot be measured directly, but may be inferred from a set of observations at each time step. The application of this algorithm to object tracking in computer vision is known as the Condensation (CONDitional DENSity propAGATION) algorithm [1]. The algorithm is based on propagating a probability density function for the state of the system (α_t) over time using a model of the dynamics of the system ($f(\alpha_{t+1}|\alpha_t)$) and a set of observations at each timestep (Y_t).¹ This is a two stage algorithm. First, the current density is propagated into the future using the dynamic model:

$$f(\alpha_{t+1}|Y_t) = \int f(\alpha_{t+1}|\alpha_t) dF(\alpha_t|Y_t) \quad (1)$$

Second (using Bayes theorem), the probability density of the system state at time $t + 1$ (given observation at this time Y_{t+1}) is:

$$f(\alpha_{t+1}|Y_{t+1}) = \frac{f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|Y_t)}{f(y_{t+1}|Y_t)} \quad (2)$$

where

$$f(y_{t+1}|Y_t) = \int f(y_{t+1}|\alpha_{t+1}) dF(\alpha_{t+1}|Y_t) \quad (3)$$

For many problems in the real world, these integrals cannot be solved analytically so numerical methods must be used. The particle filter approach is to approximate these densities with a set of discrete samples (‘particles’). It is as such only necessary to be able to evaluate $f(y_t|\alpha_t)$ (the probability of the observation y_t given the system state α_t) and $f(\alpha_{t+1}|\alpha_t)$ (a first order model of the system dynamics). As such, the prediction density (Eq. (1)) becomes:

$$f(\alpha_{t+1}|Y_t) = \sum_{\alpha_t} Pr(\alpha_{t+1}|\alpha_t) f(\alpha_t|Y_t) \quad (4)$$

¹ Notation taken from Ref. [10].

and the filtering density (Eq. (2)) becomes:

$$f(\alpha_{t+1}|Y_{t+1}) = \frac{f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|Y_t)}{\sum_{\alpha_{t+1}} f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|Y_t)} \quad (5)$$

In practice, the conditional state probability distribution functions ($f(\alpha_{XX}|Y_{XX})$) are approximated by a fixed size set of samples. This allows for highly efficient sampling from these distributions. The algorithm proceeds as shown in Fig. 1.

2.1.1. Temporal analysis using particle filters

The original application of the particle filter in machine vision was for object tracking [1] in an image sequence, in which models of an object (spatial and temporal) were fitted to an image sequence to obtain object properties over time. Propagating multiple hypotheses gave improved robustness over single hypothesis techniques such as Kalman filters [11]. Condensation has also been applied to the field of temporal modelling (prediction and classification) of objects. Johnson and Hogg [12] use condensation to propagate multiple prediction hypothesis for pedestrian trajectory classification. In this method, object tracking is performed by a separate (deterministic) module. Walter et al. [13,14] show that incorporating current observation information into this scheme gives improved classification results for human trajectory and gesture classification applications. Again, object tracking is performed by a separate module. Black and Jepson [15] use a similar scheme with multiple temporal models as a combined tracker and classifier to analyse an augmented whiteboard.

2.1.2. Overcoming the curse of dimensionality using current time priors

Particle filters work well when the conditional densities ($f(y_t|\alpha_t)$) are reasonably flat; however, when there are severe outliers, the method is highly inefficient as it has no ability for adaption of the prior during the filtering density stage. In such cases, many samples are allocated to parts of the distribution with high predicted (prior) probability, but low observational probability with (relatively) few samples allocated to parts of the distribution with low predicted probability but high observational probability. This necessitates the use of a large number of samples to retain

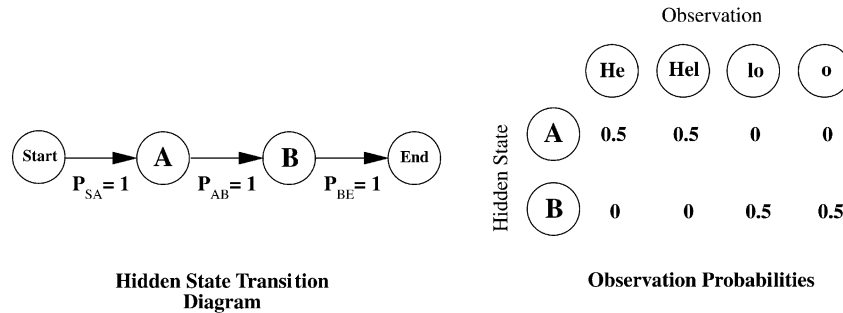


Fig. 2. Modelling the word Hello using a hidden Markov model.

approximation accuracy. This problem increases with the dimensionality of the problem and is often called ‘the curse of dimensionality’. This is in contrast to other stochastic techniques such as Markov Chain Monte Carlo [16] (or Genetic Algorithms) which update their prior approximation adaptively due to their iterative formulation and, as such, do not allocate a (relatively) large number of samples to parts of the distribution with low observational probability. This is discussed by Pitt and Shephard [10] in a non-machine vision context. They propose the addition of an auxiliary variable (k) to the system state representation, which represents an index into the prior probability density. A weight (π^k) is associated with each index value, thus the evaluation of the filtering density may be written as:

$$f(\alpha_{t+1}, k | Y_{t+1}) \propto f(y_{t+1} | \alpha_{t+1}) f(\alpha_{t+1} | \alpha_t^k) \pi^k \quad (6)$$

This addition of the auxiliary variable may be thought of as introducing an additional prior (one based on current timestep observations). Initially, the weight values are set equal (the method is then equivalent to the standard particle filter). After the filtering density has been evaluated for a number of samples, the weights are updated with respect to the samples evaluated up to this point and the new weights used in subsequent evaluations. This method is applicable to any application to which the standard particle filter is applicable and has been demonstrated to outperform the standard particle filter on the analysis of stock market data.

MacCormick and Isard [17] also discuss the inefficiencies of the conventional particle filtering approach. They propose the use of ‘partitioned sampling’ in problems where: (a) the variation of the system may be separated into several independent components, (b) the conditional observation probability ($f(y_t | \alpha_t)$) may be evaluated separately for each of these components, and (c) the system dynamics may be expressed separately for each of these components. Their method is posed as a hierarchical search in which the state space is divided into two or more partitions. This is applied to a hand tracking problem in which the state space is divided into fist, thumb and index finger components. The partitions are evaluated sequentially (fist followed by thumb followed by index finger) taking a weighted re-sampling (an ‘importance sample’) from the

output of the previous stage as the prior for the subsequent stage.

The algorithm presented in this paper is along similar lines to the work of Pitt and Shephard [10] and MacCormick and Isard [17] in that it uses partial observation at the current timestep to adjust the prior used to control the sampling scheme. In a generic re-writing of the particle filter equations for these (and our) method the filtering density becomes:

$$f(\alpha_{t+1} | Y_{t+1}) \propto f(y_{t+1} | \alpha_{t+1}) f(\alpha_{t+1} | Y_t, \hat{Y}_{t+1}) \quad (7)$$

where \hat{Y}_{t+1} is a partial observation at time $t+1$ and $f(\alpha_{t+1} | Y_t, \hat{Y}_{t+1})$ is an estimate of the system state at time $t+1$ based on this partial observation (and past observations). The differences between the methods of Pitt and Shephard [10], MacCormick and Isard [17] and our own lie in the way the prediction density ($f(\alpha_{t+1} | Y_t, \hat{Y}_{t+1})$) is formed and incorporated into the particle filter approximation.

2.2. Hidden Markov models

Hidden Markov models (HMMs) [18] are used to model observations from a stochastic process where there is some underlying structure, but observations are not deterministic. In many cases, the exact nature of this process is not observable (i.e. ‘Hidden’), for example speech, however the resulting observations (sound in the case of speech) are. HMMs model the underlying process using a first order Markov chain and the relationship between the process and the observations by a probability distribution (either discrete or continuous). Fig. 2 shows this by modelling the structure of the word ‘hello’ using one state for each syllable.

This model of the word hello illustrates two important features of the HMM. Firstly, although the Markov chain is only first order, a temporal history is encoded by limiting the state transitions possible. In this example, state A will always occur before state B thus encoding the temporal ordering of the two syllables in the word. Secondly, if the four possible permutations of syllables in the word are examined (‘Hello’, ‘He-o’, ‘Hel-lo’ and ‘Hel-o’), it can be seen that an invalid combination is possible (He-o). If the HMM were to be used for recognition this would not be a problem as there is no other word in the English language that consists of

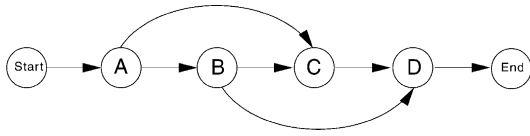


Fig. 3. A Left–right HMM architecture as used in many speech applications.

these syllables, however, if the HMM were to be used for speech generation this would present a problem. HMMs intended for sequence generation or prediction in general require more complex state transition architectures than HMMs intended for recognition only. Fig. 3 shows a typical ‘left–right’ hidden state architecture used in real speech recognition applications.

Rabiner in his excellent introduction to HMMs [18] lists three problems that need to be solved in order to use HMMs. Problem 1 is the evaluation problem, or how the probability that an observed sequence was produced by a given HMM may be calculated. Problem 2 is the recovery of a ‘correct’ sequence of hidden states for an observation sequence. Problem 3 is the training problem, or how to optimise a set of HMM parameters so as to best describe how an observation sequence (or sequences) come about. In addition to these there is a fourth problem, prediction. The problem is, given some knowledge of the system in the past/present, how can we predict the future behaviour of the system. This is an important, but not widely covered, problem. This is possibly due to speech recognition (the principal original use of HMMs) requiring no prediction from the HMM. Prediction is essential to our application and is discussed in Section 4.

Conventionally, problem 1 (evaluation) is tackled by the Forward–Backward procedure which is simply a shortcut method to enumerating every possible hidden state sequence and summing the probabilities that they give rise to the observed sequence (the shortcut is described in full in Ref. [18]). An alternative ‘online’ approximation to this is possible using a particle filter, this is described by Black and Jepson [15]. Section 6 describes our application of this method within the context of this paper. Problem 2 (recovery of hidden states) is usually tackled by the Viterbi algorithm [19] (also described in Ref. [18]). This is again a shortcut to evaluating all possible hidden state sequences. Within the particle filtering approximation proposed by Black and Jepson [15] (and used by us), this is not an issue as hidden state information is propagated with the particle filter samples and does not need to be estimated.

Problem 3 (training) is not covered by the particle filter approximation of Black and Jepson and is usually tackled using the Baum–Welch method [18]. This is essentially an application of the Expectation Maximisation (EM) algorithm [20] which calculates a locally optimum set of parameters (state transition probabilities and observational probability densities) by iteratively refining an initial estimate to these parameters. The parameter update is performed by defining two parameters: $\xi_t(i, j)$ (the prob-

ability of being in state i at time t and state j at time $t + 1$ given an observation sequence and model) and $\gamma_t(i)$ (the probability of being in state i at time t given an observation sequence and model) such that:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (8)$$

$\xi_t(i, j)$ is calculated using a variation on the Forward–Backward method [18]. The update iteration for a single sequence is then to update the state transition probabilities (a_{ij}):

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (9)$$

and to update the observational probability densities ($b_j(k)$):

$$\bar{b}_j(k) = \frac{\sum_{t=1, O=k}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (10)$$

where O is the observational state (the numerator of Eq (10) implies a sum over timesteps in which state k is observed). It is possible to optimise HMM parameters with respect to more than one observational sequence by expanding the numerator and denominator sums in Eqs. (9) and (10) to sum over all sequences as:

$$\bar{a}_{ij} = \frac{\sum_{\text{Seq}=1}^{N_s} \sum_{t=1}^{T_{\text{Seq}}-1} \xi_t(i, j)}{\sum_{\text{Seq}=1}^{N_s} \sum_{t=1}^{T_{\text{Seq}}-1} \gamma_t(i)} \quad (11)$$

$$\bar{b}_j(k) = \frac{\sum_{\text{Seq}=1}^{N_s} \sum_{t=1, O=k}^{T_{\text{Seq}}-1} \gamma_t(i)}{\sum_{\text{Seq}=1}^{N_s} \sum_{t=1}^{T_{\text{Seq}}-1} \gamma_t(i)} \quad (12)$$

3. Building spatial models for tracking applications

Previous work has described how multiple contour models of an object class may be built [21] and how the variation within these models can be separated into independent components [22]. These models are a variation on the point distribution model [2] in which a contour is modelled by a mean shape (described by a set of points) and a set of linear ‘modes of variation’. We have applied this scheme to modelling livestock, building multiple models of

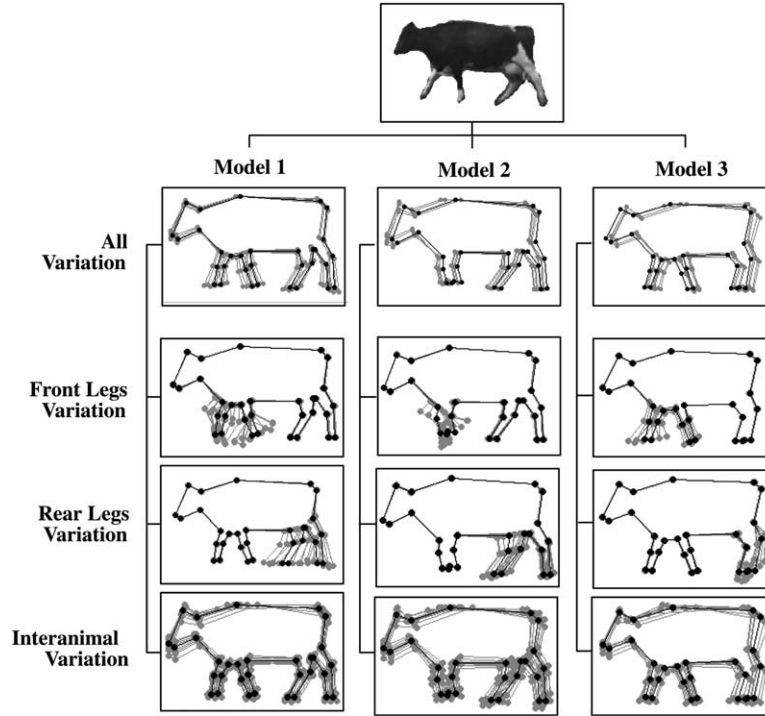


Fig. 4. Separating object variation using a hierarchical scheme.

cow outlines and separating the variation of each model into inter-animal, front legs and rear legs components as shown in Fig. 4.

Our approach to the separation of variation has much in common with the work of Edwards et al. [23] in that we use a form of discriminant analysis on a training data set partitioned into sub-classes. In our case, one set of sub-classes is contour examples relating to particular individuals and the others are groupings based on similar front or rear leg pair positions. The point distribution model can be used to express an object contour in terms of an N -dimensional vector. Linear discriminant analysis may be used on a set of such examples divided into meaningful classes to generate a description space optimised for inter-class discrimination by solving the eigen equation:

$$S_b E = \lambda S_w E \quad (13)$$

where S_b is the inter-class covariance matrix, S_w , the intra-class covariance matrix, E , the matrix of eigen-vectors (unknown) and λ is the vector of eigen-values (unknown).

The inter-class component of variation in the training data set may be removed by projecting into a truncated version of this inter-class optimised eigen-space (describing only the inter-class variation²) and then back into the point description space. The result of this double projection contains only the inter-class component of variation. This variation may be subtracted from the training data variation to produce a new training set with little or no inter-class

variation. The method may then be repeated on this training set with a different set of sub-classes to separate out a different sort of variation.

In Ref. [22], we described a similar technique known as ‘Delta Analysis’ which was demonstrated to give superior component separation for data sets in which different components of variation have some degree of spatial separation. This method produces a pair of eigen-spaces optimised for inter- and intra-class variation, respectively. The eigen-spaces are formed by performing conventional principal components analysis (once for each space) on a single covariance matrix formed for the inter-class eigen-space as:

$$S_b = \frac{1}{n_t} \sum_{i=1}^{n_c} \sum_{j=1, j \neq i}^{n_c} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} \delta_{i,j,k,l} \delta_{i,j,k,l}^T \quad (14)$$

where $\delta_{i,j,k,l} = (\mathbf{y}_{i,j} - \mathbf{y}_{k,l}) \mathbf{v}_w$, $\mathbf{y}_{x,y}$ is the data vector \mathbf{y} of class x , n_t , the total number of data items, n_c , the number of classes, n_i , the number of members in class i , n_j , the number of members in class j and the inverse intra-class variance vector

$$\mathbf{v}_w = \left(\frac{1}{v_1}, \frac{1}{v_2}, \dots, \frac{1}{v_n} \right)$$

and formed for the intra-class eigen-space as:

$$S_w = \frac{1}{n_t} \sum_{i=1}^{n_c} \sum_{j=1}^{n_i} \sum_{k=1, k \neq j}^{n_i} (\mathbf{y}_{i,j} - \mathbf{y}_{i,k})(\mathbf{y}_{i,j} - \mathbf{y}_{i,k})^T \quad (15)$$

where $\mathbf{y}_{x,y}$ is the member \mathbf{y} of class x , n_c , the total number of

² The complete eigen-space describes all variation with the inter-class variation principally described by the first few eigenvectors).

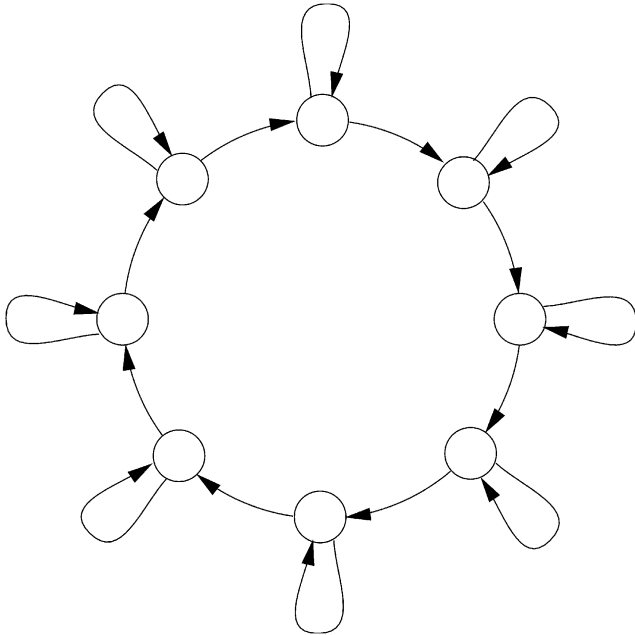


Fig. 5. CHMM hidden state architecture.

classes, n_t , the total number of data items and n_i is the number of members in class i .

As described previously, the variation relating to a truncated version of either of these spaces may be removed from the training set by projecting the data into these spaces and back (see before). The method can then be repeated with a different grouping of the data to separate out a different sort of variation. A given example (either novel or from the training set) may then be represented as a weighted sum of the eigenvectors from the various (truncated) eigen-spaces:

$$\mathbf{x} \approx \bar{\mathbf{x}} + \sum_{n=1}^{n_{wc}} w_n \mathbf{e}_{w_{c_n}} + \sum_{n=1}^{n_{bc}} b_n \mathbf{e}_{b_{c_n}} + \dots \quad (16)$$

The component weights (w_n , b_n , etc.) may be calculated for a novel object example using least squares approximation methods (exact calculation is not usually possible).

Each model is discretised by performing vector quantisation on the projections of the training data in the eigen-spaces of the model. This results in a set of prototypical ‘states’ for inter-animal, front legs and rear legs components. This goes some way to alleviating the ‘curse of dimensionality’ discussed in Section 2.1.2 as each eigen-space is then represented by a single (discrete) variable. In recent work, Toyama and Blake [24] present a Condensation based tracker in which the spatial model is based on ‘exemplars’ from a training set. The advantage of such a scheme (in terms of dealing with the curse of dimensionality) is discussed. Our approach is perhaps as close to this ideal as it is possible to get while still retaining the advantages of separating the variation into separate components.

4. Building multi-stream cyclic hidden Markov models

Cyclic hidden Markov models (CHMMs) use a hidden state architecture in which the first and last states are joined. This is shown in Fig. 5.

The multi-stream cyclic hidden Markov model is a CHMM that has multiple observation probability distributions associated with each hidden state (one for each observational ‘stream’). This is shown in Fig. 6. In our example, we model the front and rear leg pairs as two separate observational streams based on a single underlying CHMM hidden state architecture.

The training of these models is performed by extending Eq. (11) to sum over all observational streams as:

$$\bar{a}_{ij} = \frac{\sum_{Str=1}^{N_{st}} \sum_{Seq=1}^{N_s} \sum_{t=1}^{T_{Seq}-1} \xi_t(i, j)}{\sum_{Str=1}^{N_{st}} \sum_{Seq=1}^{N_s} \sum_{t=1}^{T_{Seq}-1} \gamma_t(i)} \quad (17)$$

and updating the observational densities using the update described by Eq. (12) once for each of the observational streams. It should be noted that continuous (e.g. Gaussian mixture model based) observation probability distributions could be used, however our spatial model is quantised for tracking efficiency reasons (this is discussed in Section 3).

Hidden state transition probabilities are initialised by defining parameters P_s , P_c and P_x as the probability that the next hidden state remains the same, changes or the current state is the last state in the sequence, respectively. If we assume initially that these parameters are the same for each hidden state we get the probability distribution given in Eq. (18) for the cycle length in states.

$$P(n) = P_s^r P_x P_c^N \times \binom{N+r-1}{N-1} C_r = P_s^r P_x P_c^N \frac{(N+r-1)!}{(N-1)!r!} \quad (18)$$

where n is the sequence length, $P(n)$, the probability of a sequence of length n resulting from the CHMM, N , the number of hidden states in the CHMM and r is the number of hidden state repetitions ($n - N$).

The (hand labelled) training sequences consist of a series of symbolic descriptions representing each frame of a particular observational sequence. Each of these descriptions contains a token representing the spatial model used (Fig. 4), a token representing the (quantised) front legs state and a token representing the (quantised) rear legs state (a separate temporal model is used for inter-animal state). The use of discrete observational probability distributions allows easy concatenation of the observational probability densities as a single histogram (for each stream and hidden state) as shown in Fig. 7.

Training sequences are parsed into cycles (of various lengths) based on spatial model transitions. An arbitrary model transition (e.g. model 2 to model 1) is selected as the cycle start/end point as shown in Fig. 8.

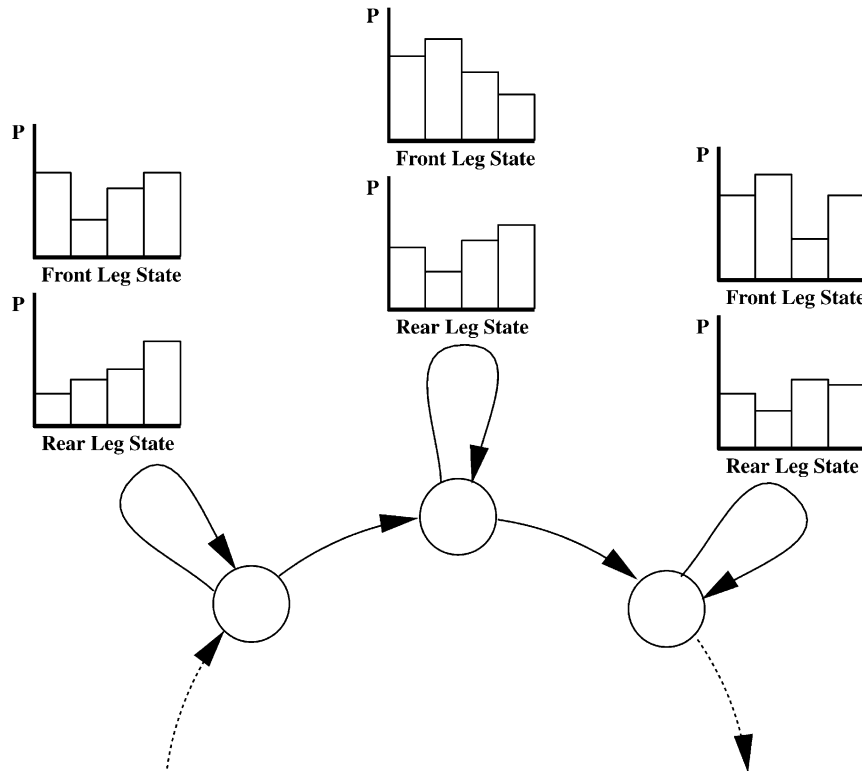


Fig. 6. A ‘Multi-stream’ hidden Markov model.

The number of hidden states (N) is selected to be the minimum cycle length in the training set. Values of N below this value may be used giving a CHMM with more generality and less specificity. Values of N above this value are not used as this would not allow generation of all training sequence lengths. An initial estimate of P_x as the reciprocal of the average sequence length is used and (given $P_c + P_s + P_x = 1$) an exhaustive one-dimensional search is performed in P_c (with fine quantisation) to minimise the square difference between the theoretical probability distribution of cycle length ($P(n)$) and a distribution calculated from the training sequences.

Initial visible state probability distributions are estimated by aligning each training sequence cycle with the set of hidden states, ‘time stretching’ such that the training sequence length is equal to the number of hidden states in the CHMM. Probabilities are estimated from the relative

number of observations lying completely or partially over each hidden state as shown in Fig. 9.

The transition and observation probability estimates are improved using the Baum–Welch re-estimation method [18].

5. Object tracking using Re-sampling Condensation

The main drawback with the Condensation algorithm is that sample location is determined purely from a prior formed by prediction from past observations. Isard and Blake explain [25] that this results in sample locations clustering round regions of predicted high probability with few samples representing areas of lower predicted probability. This is a problem in applications where there are multiple possible outcomes of differing probabilities. Other sampling algorithms such as Markov Chain Monte Carlo [16] and Genetic Algorithms do not exhibit this problem as they iteratively re-sample the solution based on previous results. These algorithms are however unsuitable for ‘real time’ applications due to their high computational cost.

In Section 2.1.2, we discussed ways of including partial observations in the prior density used by dividing the sampling effort into stages. The prior used in later stages is updated with respect to observations taken in the earlier stages. Pitt and Shephard [10] model this density as an auxiliary variable tied to the prior (particle) approximation. This method is generally applicable and has a slight

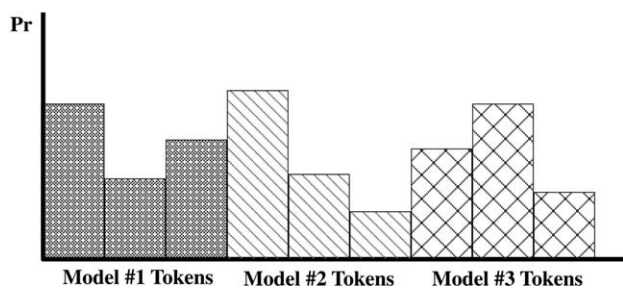


Fig. 7. Representing observational tokens from multiple models with a single histogram.

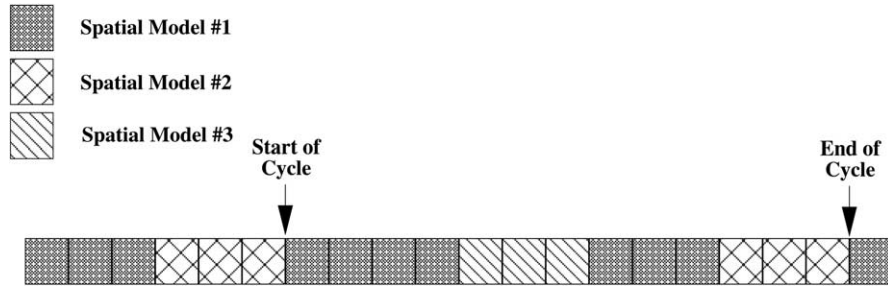


Fig. 8. Dividing the training data into cycles based on model transitions.

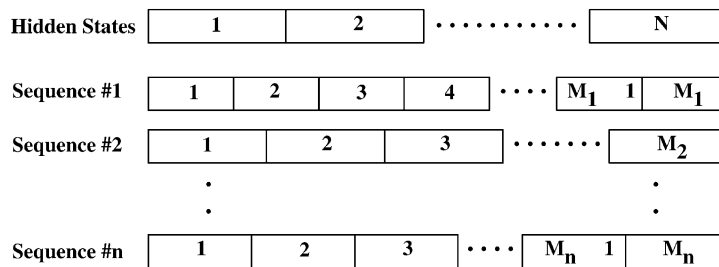


Fig. 9. Initial estimation of observation probabilities.

computational overhead over the conventional particle filtering/Condensation scheme. MacCormick and Isard [17] present a method applicable only in the case where the system state may be partitioned and observation densities and dynamics modelled separately for each of these partitions. This work illustrates the benefit of partitioning the system state but the criteria of independent observation evaluation for the different partitions is not always possible to meet (active appearance models³ are a good example of this [23]). In the rest of this section, we present an alternative method for incorporating partial observations in the prior densities known as Re-sampling Condensation which also utilises the advantage of separating out variation components without the constraints of independent observation evaluation and independent dynamics modelling for these components.

Re-sampling condensation is a novel two stage algorithm with no computational overhead over the standard Condensation algorithm (the algorithm has in fact lower computational cost) that, under certain circumstances, gives more robust tracking results as image information is included in sample location. Re-sampling Condensation splits the samples into two groups. Sampling of the first group is performed using standard Condensation (as in Section 2.1). Multiple samples are then selected stochastically from this initial sample set (based on a fitness function) and combined to give a new sample location. Mathematically, the prior used for this second sample set is a ‘partial’ posterior calculated based on a partial observation. This ‘second stage

prediction density’ may be written as:

$$f(\alpha_{t+1}|\hat{Y}_{t+1}) = \frac{f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|Y_t)}{\sum_{\hat{\alpha}_{t+1}} f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|Y_t)} \tag{19}$$

The reader should note the prediction density in Eq. (19) is simply the filtering density in Eq. (5) evaluated over a partial sample set. This prediction density ($f(\alpha_{t+1}|\hat{Y}_{t+1})$) is used in place of the prior prediction density ($f(\alpha_{t+1}|Y_t)$) to form the ‘second stage filtering density’:

$$f(\alpha_{t+1}|Y_{t+1}) = \frac{f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|\hat{Y}_{t+1})}{\sum_{\hat{\alpha}_{t+1}} f(y_{t+1}|\alpha_{t+1})f(\alpha_{t+1}|Y_t)} \tag{20}$$

As with the conventional particle filtering scheme, densities are approximated by particles. The particle approximation to the second stage prediction density described in Eq. (20) is formed by sampling from the prediction density (Eq. (19)) three times per particle⁴ and combining the results as shown in Fig. 10.

The process of sampling from the second stage prediction density and evaluating the second stage filtering density approximation is analogous to a single iteration of a genetic algorithm in which there are three parents and two crossover points. The location of the crossover points is fixed, based on the separate component description (Fig. 10). This method extends to other models in which variation is separated into N components. In such cases, an $N - 1$ point crossover would be used. The complete Re-sampling Condensation scheme is shown in Fig. 11.

³ The active appearance model algorithm is not currently suitable for stochastic tracking algorithms due to the high computational cost of a single observation evaluation. However, this may not be the case in the future.

⁴ The three samples are constrained to come from the same spatial model (Fig. 4) as they cannot otherwise be combined.

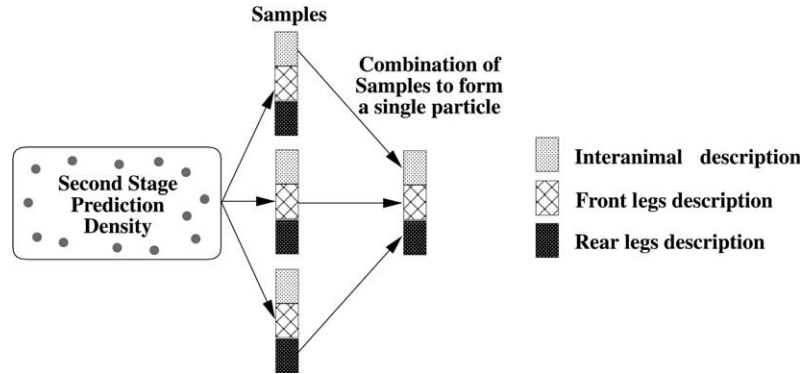


Fig. 10. Combination of multiple samples to form a single particle.

It should be noted that Re-sampling Condensation relies on the variation within the spatial model being separated into pseudo-independent components (position, scale, inter-animal, front legs and rear-legs characteristics in the livestock example). These are analogous to the ‘genes’ in a genetic algorithm. Experimental results (Section 7) suggest that the more components that a spatial model may be separated into, the better the final solution. This is intuitive as, considering a system with two independent parameters with N and M possible states for each parameter, searching using a model with a single parameter results in a search space of size $N * M$ where as searching using a model with two parameters results in two search spaces of size N and M , respectively. This is shown in Fig. 12.

In most tracking examples, parameters are not truly independent, however, component separation can still yield improved search efficiency. It should be noted that the method of MacCormick and Isard [17] incorporates this computational saving also, however, the (more generally applicable) method of Pitt and Shepard [10] does not.

5.1. Prediction using temporal models

We have, as yet, not described how the temporal prediction (dynamics) function ($\Pr(\alpha_{t+1}|\alpha_t)$) is to be evaluated. As in the conventional particle filtering scheme this is done on a per-particle basis. Eq. (4) describes the prediction process. A particle is sampled from the filtering (posterior) distribution at the previous timestep (evaluating the $f(\alpha_t|Y_t)$ part of Eq. (4) and a (usually stochastic) temporal model (or models) is/are used to predict the location of this particle at the next timestep. Section 4 describes a hidden Markov model based temporal model for predicting forward the front and rear leg states. Prediction is performed by stochastically selecting a hidden state (based on the current state and hidden state transition probabilities) and visible states from the associated observational probability densities.

In our application the state (α) of an object (a cow) consists of inter-animal parameters, position and size in addition to the leg position parameters. The inter-animal characteristics are predicted forward stochastically using a

first order Markov chain trained on the same training sequences used to train the model described in Section 4. Position and scale are determined stochastically from the mean and standard deviation of these parameters over the previous particle set, by assuming a normal distribution and selecting from this distribution (see Ref. [26] for method).

5.2. Evaluation of observational density

Another important part of the particle filtering algorithm is evaluating the observational likelihood function ($f(y_{t+1}|\alpha_{t+1})$ in Eq. (5). In our example, this is based on edge information. Edge strength (based on a pair of horizontal/vertical Sobel edge detectors) is modelled over time as a Gaussian distribution with the mean and the square of deviation from the mean being updated on a rolling average basis as in Eqs. (21) and (22), respectively.

$$\bar{I}_{x,y,t+1} = K\bar{I}_{x,y,t} + (1 - K)I_{x,y,t} \quad (21)$$

$$\bar{\sigma}_{x,y,t+1}^2 = K\bar{\sigma}_{x,y,t}^2 + (1 - K)(\bar{I}_{x,y,t} - I_{x,y,t})^2 \quad (22)$$

If \bar{I} is considered an estimate of the mean and $\bar{\sigma}^2$ an estimate of the variance of edge strength over time, any novel edges with a strength greater than a fixed number of standard deviations above the mean edge strength may be considered ‘moving’ or ‘transient’ edges (these values are not used to update the distribution in Eqs. (21) and (22)). It is these edges we base our observational likelihood function on. The advantage of these over static edges (as used by Isard and Blake [1]) is that static clutter is eliminated and many false maxima in the observational density are eliminated.

To perform the evaluation of the observational likelihood function the system state is used to generate a set of hypothesised edges in the image domain based on the spatial model (see Section 3). These edges are compared on a pixel by pixel basis with the image pixels and the proportion of matches recorded as the observation likelihood for that sample. This differs slightly from the approach of Isard and Blake [1] who only evaluate a subset of the edge locations (evenly spaced around the model perimeter). Their method evaluates the maximum edge strength along a normal to the perimeter (rather than simply at the evaluation point as in

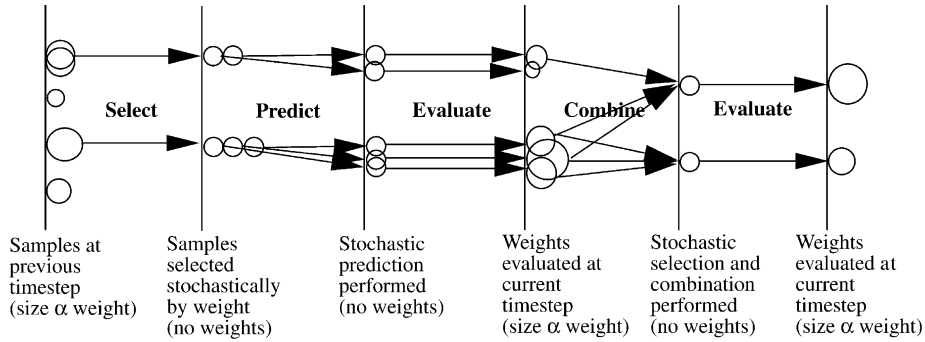


Fig. 11. Re-sampling condensation flow diagram.

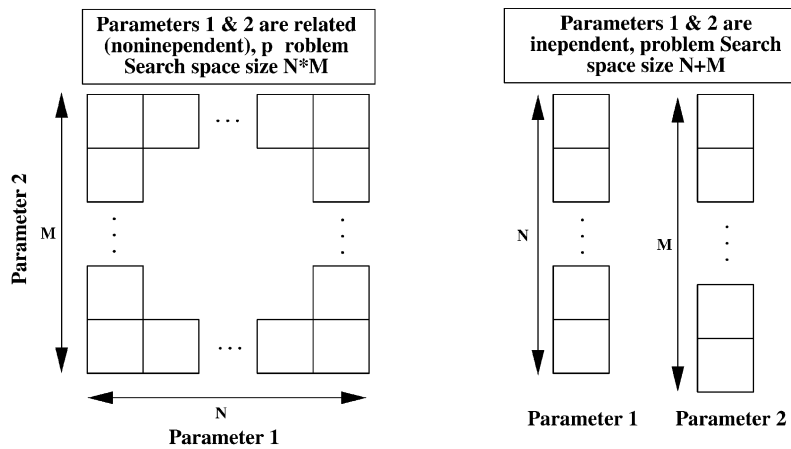


Fig. 12. Reducing problem search space size for independent parameters.

our approach). The advantage of this method is hypothesised edges near, but not corresponding to, image edges contribute positively to the observational likelihood. In our application there is significant noise and gaps in edges occur in some frames. In order to make the Isard and Blake approach robust to this a large number of evaluation points are required. This is because if a small number of points are used and one falls on a gap the observational likelihood is greatly biased. Increasing the number of evaluation points increases the computational cost of a single evaluation and

thus reduces the number of particles possible on a given computational resource. Our method (being simpler to evaluate at each contour point) can evaluate more points around the contour for the same computational cost. Our method of calculating edges produces reasonably wide



Fig. 13. Typical livestock tracking scenario.

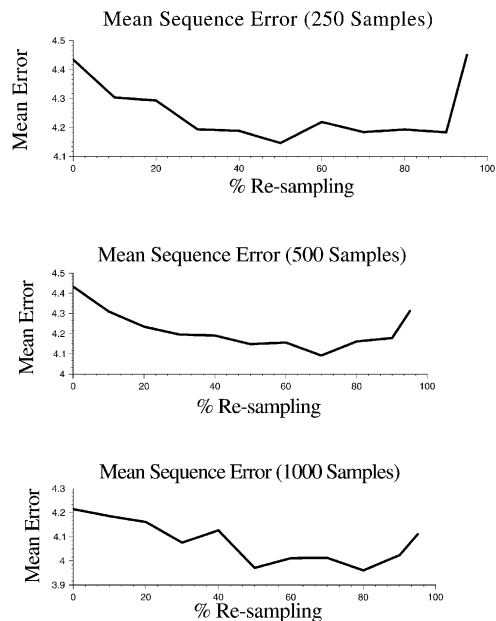


Fig. 14. Mean error (pixels) of tracker at different levels of re-sampling.

edges and, as such, low observational likelihood for hypothesised edges close to, but not corresponding to, image edges is not a great problem.

6. Combined tracking and behaviour analysis using multiple CHMMs

In Section 2.1 it was described how the Condensation algorithm has wider application than simple object tracking. In particular Black and Jepson [15] use Condensation to track and classify the trajectories of a coloured whiteboard marker. In their scheme multiple trajectory models are used and the ‘gesture’ classified as one of six actions to be performed.

In our scheme we use the combined tracker and classifier paradigm of Black and Jepson to model object shape change over time in a Re-sampling Condensation framework (see Section 5). In our example ‘normal’ and ‘lame’ behaviours are modelled by separate CHMMs (see Section 4) within the Re-sampling Condensation framework. Samples are allocated to each model initially in even proportion and the CHMM is propagated through time with the sample. Over time the CHMM that best fits the observed object behaviour will dominate (i.e. more samples will be associated with that model). A simple classification can then be performed by comparing the number of samples associated with each model. A more complex classification method involves summing the posterior probabilities (relative fitnesses) for samples associated with each CHMM. We take this latter approach in our evaluation (see Section 7.2).

7. Evaluation and discussion

7.1. Evaluation of tracking results

Re-sampling condensation was evaluated using a test set of 10 sequences of healthy cows walking from right to left in a farmyard setting as shown in Fig. 13. These sequences are of approximately 5 s in length at 25 fps and contain at least three complete cycles of the CHMM. Ground truth about these sequences was obtained by hand fitting landmark points. Separate sequences were used for the construction of the spatial (shape) model. Approximately 500 hand labelled outline examples were used in the training of the spatial models. These were taken from similar sequences to the test data.

A set of leave-one-out tests was performed by building a CHMM from nine of these sequences and using the remaining sequence as a test sequence. The training data was obtained by projecting the hand fitted ‘ground truth’ points into the model parameter spaces and selecting the nearest vector quantisation prototype (a crude but reasonably effective method). Tracking was performed at various levels of re-sampling (i.e. number of re-samples vs. number

of condensation samples) from 0% (Normal Condensation) to 95%. The tracking results for the sample of maximum fitness were compared to the hand fitted ground truth, and statistics such as mean error gathered. Some results at different numbers (250, 500 and 1000) of total samples are given in Fig. 14 over.

The results in Fig. 14 show lower average error rates for Re-sampling Condensation at intermediate levels of re-sampling, although there is a drop off in performance at high levels of re-sampling. It should be noted that the tracking error is superimposed on the quantisation error inherent in the quantised model. The average error for the quantised ground truth data used for training is 4.08 pixels using the crude quantisation method described previously. It is encouraging that the tracker can, at optimal re-sampling, improve on this value. Error standard deviation shows similar trends to mean error although this is always higher than for the quantised training data. Examining individual sequences around the optimum operating points indicated by the graphs in Fig. 14, the Re-sampling Condensation tracker performs better than Condensation alone (0% re-sampling) in 80–100% of cases. Even at very low (< 20%) and very high (> 90%) levels of re-sampling the Re-sampling Condensation tracker can perform better in more than 50% of cases.

These results are as would be expected as the inclusion of current image information in sample location improves tracking performance. The fall off in performance at high levels of re-sampling is due a ‘gene deficiency’ in the initial Condensation set of samples when few samples are allocated to the Condensation stage. It may be the case that this initial ‘population’ of samples contains no members with the ‘most correct’ individual characteristics and as such the most correct (highest fitness) solution cannot be found by the re-sampling stage. It may also be the case that the initial population of samples contains a high proportion of members with the most correct individual characteristics, in which case tracking performance will be particularly good. It is as such not desirable to operate at high levels of re-sampling as tracking robustness is lower than at intermediate levels of re-sampling. It should also be noted that the optimum level of re-sampling increases with the total number of samples due to this phenomenon.

7.2. Evaluation of lameness classification

The evaluation of the scheme in the detection of lameness was performed on sequences of humans walking as insufficient amounts of lame cow data were available. The 11 (healthy) subjects were asked to perform a choreographed lame walking motion in addition to their regular walking motion as shown in Figs. 15 and 16 respectively.

The choreography of the lameness was performed by imitation of the first subject by subsequent persons in order to obtain a degree of similarity between lameness styles.



Fig. 15. A 'typical' lame walk sequence.



Fig. 16. A 'typical' healthy walk sequence.

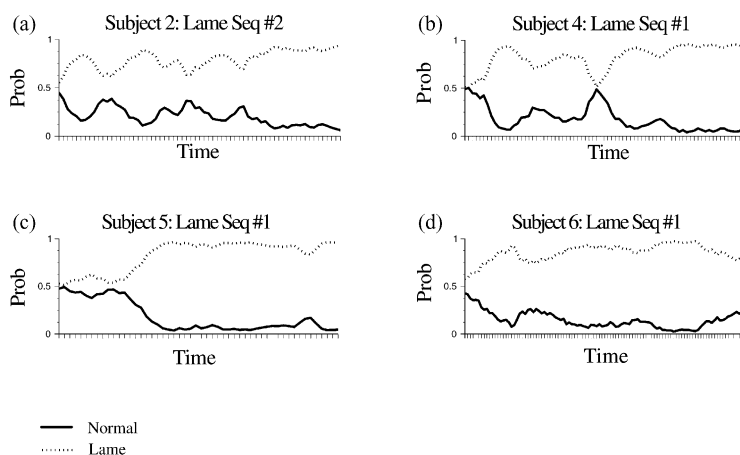


Fig. 17. Model probability vs. time for 'typical' lame walk sequences.

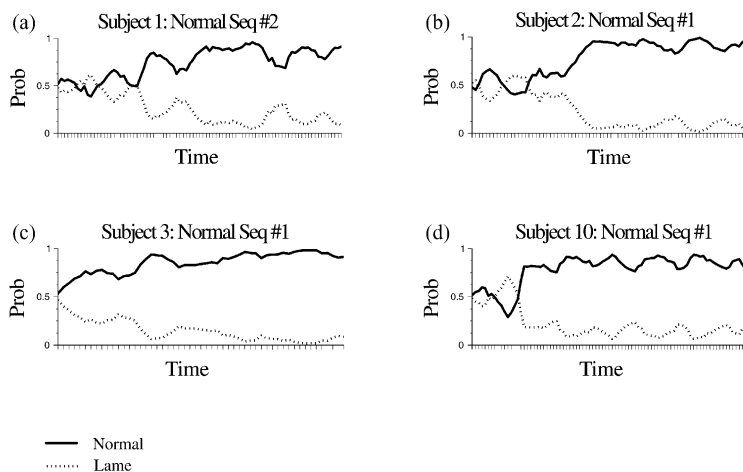


Fig. 18. Model probability vs. time for 'typical' normal walk sequences.

There was, however, a reasonably large degree of variation between subjects (and between examples of the same subject), as could be expected. The magnitude of difference between the human lame and 'healthy' sequences was similar to that between the livestock lame and healthy sequences we could obtain, however the human subjects were not actually lame. The degree to which their choreographed lameness matches true human lameness is unclear, however detecting human lameness was not our goal.

Two sets of each were taken for each person and a set of

leave-one-out tests performed. The spatial (shape) model used was a B-spline based model (a la Baumberg and Hogg [3]) rather than the straight line approximation as used for the cows as people are not well approximated by straight lines. The sum of the posterior probability (normalised fitness) for each CHMM was recorded over time for each sequence. Averaging the probabilities over time (excluding the first cycle to allow for initialisation of the tracker) gives an indication of the relative probability that each sequence is either healthy or lame.

Using this method all 44 sequences were correctly

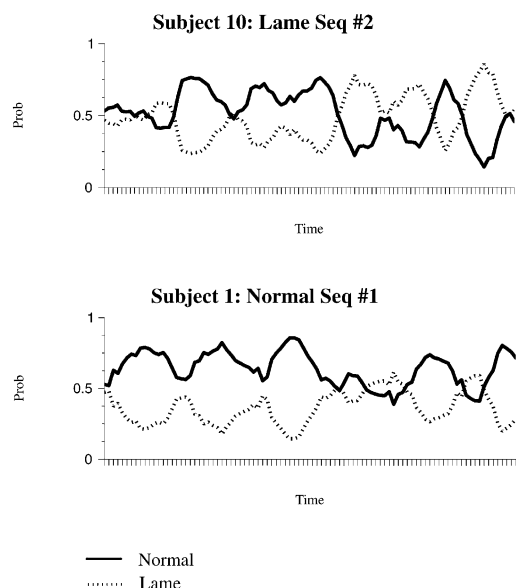


Fig. 19. Model probability vs. time for the two ‘non-typical’ sequences

classified and in all but two cases the results were very clear as can be seen from Figs. 17 and 18.

Two sequences produced results which, although they were classified correctly, were not as clear (see Fig. 19). On investigation this was found to be related to the lame portion⁵ of the lame CHMM being similar to the healthy walking motion in the rotationally normalised shape representation. The spatial model used to model the cows is not normalised by rotation and thus would not be subject to this problem. In the non-typical lame sequence a significant number of samples were being propagated half a cycle out of phase with the actual cycle resulting in poor tracking performance. Similarly for the healthy sequence samples relating to the lame CHMM were being propagated in both phases resulting in reasonable tracking by the lame model compared to the healthy model. These problems would be eliminated by using a non-rotationally normalised model or including relative rotation as an additional stream in the CHMM as out of phase samples would not be propagated.

8. Conclusions and further work

We have presented an extension to Isard and Blake’s Condensation algorithm [1] that uses partial observation at the current timestep to improve the prior density used in particle sampling for object tracking. The method works by partitioning the particle set between two stages; an initial ‘sampling’ stage (identical to the standard Condensation framework) and a ‘re-sampling’ stage which uses the output of the first stage as a prior and a novel particle combination

method (which is in many ways similar to techniques used in genetic algorithms) to produce the final posterior density estimate.

Our method differs from the partitioned sampling algorithm of MacCormick and Isard [17] in the fact our partial observation is a complete evaluation of the observational likelihood ($f(y_{t+1}|\alpha_{t+1})$) in Eq. 5 for the initial partition. Their algorithm only evaluates a part of the observational likelihood (e.g. one physical section of hand being tracked at a time) which is used to form an importance sampling function for the next partition. Their work may be thought of as an application of the ICondensation importance sampling algorithm of Isard and Blake [25] in which an independent observational system (e.g. a colour or blob tracker) is used to form an importance sampling function to a Condensation tracker leading to more efficient particle sampling. The key drawback of such a system is that such an independent observational system must be available and, in the case of MacCormick and Isard’s work [17], this is achieved by partitioning the problem. This requires the observational likelihood to be independently evaluatable for each partition (and the dynamics to be independently evaluatable at for each partition). As discussed previously this is not always possible even when variation may be separated, (e.g. for active appearance models [23]) which leads to this technique being less than generally applicable.

Pitt and Shephard’s ‘Auxiliary Particle Filter’ method [10] is conceptually closer to our work than the work of MacCormick and Isard [17] as it does not make any assumptions about being able to partition the observational likelihood function. The prior used at the second stage of this algorithm is augmented with an auxiliary variable for each particle in the posterior from the previous timestep. This variable is calculated from a partial observation at the first stage. The principal conceptual difference between this work and our own is that this auxiliary variable is used to bias the existing prior (thus necessitating evaluation of system dynamics at the second stage), whereas we define a new prior based on the partial posterior produced by the first stage (thus no dynamics need be evaluated at the second stage). Our new prior is formed using combination of particles which utilises the benefit of separating variance without imposing constraints on the observational likelihood or dynamics functions (as in Ref. [17]). Our method is more computationally efficient, under many circumstances, than the conventional particle filtering/Condensation algorithm as dynamics need not be evaluated at the second (re-sampling) stage.⁶This allows more particles to be used for a given computational resource and thus a better approximation to the posterior density ($f(\alpha_{t+1}|Y_{t+1})$).

Experimental results show improved tracking performance over the conventional Condensation algorithm with the

⁵ N.B.: The lame walking cycle consists of approximately half a cycle that is identical to a healthy walk and half a cycle that differs. In the healthy walking cycle the two half cycles are identical from the view used.

⁶ There is a small computational overhead in the combination sampling but this is usually small in comparison with the evaluation of the dynamics or the observational likelihood function.

existence of an optimal proportion of re-sampling particles. We have also shown how multiple cyclic hidden Markov models (CHMMs) may be used to model the temporal changes in object shape that represent a walking behaviour. These multiple CHMMs may be included in the Re-sampling Condensation framework to form a combined tracker and behaviour classifier. Experimental results using this scheme to classify the difference between a normal human walking motion and an artificial lame motion for previously unseen persons are very encouraging although insufficient lame data was available to evaluate this scheme on Livestock. This combined tracker and classifier paradigm has very wide applicational scope beyond livestock monitoring and could be applied to areas such as gesture and gait recognition in future work.

Acknowledgments

This work was funded by the BBSRC. The authors would like to thank Prof. M. Forbes from the Leeds Institute for Plant Biotechnology and Agriculture (University of Leeds) and the staff at ADAS Bridgets Research Institute for their help with data collection.

References

- [1] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, *International Journal of Computer Vision* 29 (1998) 5–28.
- [2] T.F. Cootes, C. Taylor, D. Cooper, J. Graham, Training models of shape from sets of examples, *Proceedings of the British Machine Vision Conference*, 1992, pp. 9–18.
- [3] A. Baumberg, D. Hogg, An efficient method for tracking using active shape models, *Proceedings of the IEEE Workshop on Motion of Non-rigid Objects*, 1994, pp. 194–199.
- [4] D. Terzopoulos, R. Szeliski, Tracking with Kalman snakes, *Active Vision* (1992) 3–20.
- [5] A. Pentland, B. Horowitz, Recovery of non-rigid motion and structure, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1991) 730–742.
- [6] N. Johnson, A. Galata, D. Hogg, The acquisition and use of interaction behaviour models, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 866–871.
- [7] N. Sumpter, A. Bulpitt, Learning spatio-temporal patterns for predicting object behaviour, *Proceedings of British Machine Vision Computing*, 1998, pp. 649–658.
- [8] C. Wren, A. Pentland, Understanding purposeful human motion, *Proceedings of the IEEE International Workshop on Modelling People*, 1999, pp. 19–25.
- [9] J. Davis, A. Bobick, The representation and recognition of action using temporal templates, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 928–934.
- [10] M. Pitt, N. Shephard, Filtering via simulation: auxiliary particle filters, *Journal of the American Statistical Society* 94 (1999) 590–599.
- [11] A. Blake, R. Curwen, A. Zisserman, A framework for spatiotemporal control in the tracking of visual contours, *International Journal of Computer Vision* 11 (1993) 127–145.
- [12] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, *Image and Vision Computing* 14 (1996) 609–615.
- [13] S. Gong, M. Walter, A. Psarrou, Recognition of temporal structures: learning prior and propagating observation augmented densities via hidden markov states, *Proceedings of the IEEE International Conference on Computer Vision*, 1999, pp. 157–162.
- [14] M. Walter, A. Psarrou, S. Gong, Learning prior and observation augmented density models for behaviour recognition, *Proceedings of BMVC*, 1999, pp. 23–32.
- [15] M. Black, A. Jepson, Recognizing temporal trajectories using the condensation algorithm, *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 16–21.
- [16] W. Gilks, S. Richardson, D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, 1996.
- [17] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, *Proceedings of the European Conference on Computer Vision* 2 (2000) 3–19.
- [18] L. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE* 77 (1989) 257–286.
- [19] A. Viterbi, Error bounds for convolution codes and an asymptotically optimal decoding algorithm, *IEEE Transactions on Information Theory* 13 (1967) 260–269.
- [20] A. Dempster, D.R.N. Laird, Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society. Series B* 39 (1977) 1–38.
- [21] D. Magee, R. Boyle, Building shape models from image sequences using piecewise linear approximation, *Proceedings of British Machine Vision Computing*, 1998, pp. 398–408.
- [22] D. Magee, R. Boyle, Building class sensitive models for tracking application, *Proceedings of the British Machine Vision Conference*, 1999, pp. 398–408.
- [23] G. Edwards, A. Lanitis, C. Taylor, T. Cootes, Statistical models of face images—improving specificity, *Image and Vision Computing* Vol 16, no. 3, pp 203–211, 1998.
- [24] K. Toyama, A. Blake, Probabilistic tracking in a metric space, *Proc. International Conference on Computer Vision*, pp 50–57, 2001.
- [25] M. Isard, A. Blake, Icondensation: unifying low-level tracking in a stochastic framework, *Proceedings of the Fifth European Conference on Computer Vision*, 1, 1998, pp. 893–908.
- [26] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1992.