A NEW WAY TO USE HIDDEN MARKOV MODELS FOR OBJECT TRACKING IN VIDEO SEQUENCES

Sébastien Lefèvre and Emmanuel Bouton and Thierry Brouard and Nicole Vincent

Laboratoire d'Informatique, Université de Tours
64 avenue Portalis, 37200 Tours - FRANCE
(lefevre,brouard,vincent)@univ-tours.fr

ABSTRACT

In this paper, we are dealing with color object tracking. We propose to use Hidden Markov Models in a different way as classical approaches. Indeed, we use these mathematical tools to model the object in the spatial domain rather than in the temporal domain. Besides in order to manage multidimensional (color) data, Multidimensional Hidden Markov Models are involved. Object learning step is performed using the GHOSP algorithm whereas object tracking step is done by approximate object position prediction and then precise object position localisation. This last step can be seen as an object recognition problem and will be solved using a method based on the Forward algorithm.

1. INTRODUCTION

Object tracking problems have been studied for several years and many different approaches have been proposed. Different cases may occur, the object to be tracked can be any mobile object in a scene or it can be a specific object that is already known and the trajectory of which is to be followed. Here we are interested in the second case. It is possible to learn the considered object in a first time when representations of this object are available. Once the learning has been performed, object tracking consists in object occurrence contextual searching in successive frames of a video sequence. This can be seen as a pattern recognition problem and so can be solved with appropriate methods, as statistical pattern recognition tools.

Among them, Hidden Markov Models are widely used in object tracking. Several authors have even proposed their own Markovian models in order to deal with object tracking, as Rigoll et al [1], Wilson and Bobick [2], and Bui et al [3]. Most of the time, HMM are used in the temporal domain to model the object motion through time. Contrary to these classical approaches, we propose in this paper to perform object tracking using HMM in the spatial domain.

So we are dealing with object learning and tracking using HMM in the spatial domain rather than in the temporal domain. More precisely, we use an extension of HMM, Multidimensional Hidden Markov Models, in order to process color images as they contain multidimensional data.

This HMM model will be recalled in the first section. The object learning phase will be presented next. It is performed offline using the GHOSP algorithm. In the following section we will present the tracking phase, which is composed of two successive steps. We first predict the approximate object position using a simple motion estimator (instead of HMM as in classical methods). We then compute the exact position in a restricted area using HMM built in the learning phase. This last step relies on the use of the Forward algorithm. Finally some results are given to illustrate the efficiency of the proposed approach.

2. MULTIDIMENSIONAL HIDDEN MARKOV MODELS

Hidden Markov Models are mathematical tools widely used in statistical pattern recognition. A HMM can be viewed as a combination of two random variables representing the states of a discrete stochastic process: the first one defines an automata (the hidden part) whereas the second one defines how a given state of the automata produces a given symbol (the visible part). A HMM can then be characterized by a set S (with cardinal N) of hidden states of the HMM, a set V (with cardinal M) of symbols which can be generated by the HMM, a matrix B (of size $M \times N$) giving probabilities of symbols generation in each state, a matrix A (of size $N \times N$) giving probabilities of transitions between states, a vector Π (of size N) giving probabilities for the initial state. A HMM is generally modelled by the triplet $\lambda = (A, B, \Pi)$.

These statistical tools are particularly adapted to model 1-D physical phenomena. However, in our case we are studying images which consist of 2-D sets of data. So we have to choose a given rule in order to scan the image and to obtain a 1-D signal. We have compared experimentally several techniques for image scanning. The results obtained have shown that, contrary to our assumptions, the use of more complex rules like Peano scan [4, 5] does not improve so much the quality of the analysis.

Each pixel is characterized by its color using three components. Many studies have shown the advantages of color image analysis with respect to gray level image analysis. But whereas in classical methods labels (*i.e.* symbols) are associated with regions in the multidimensional space here we are using multidimensional HMM, that is to say labels are defined on each dimension. Two extensions of HMM to multidimensional case have been proposed independently in [6] and [7]. In this paper, we will use the approach proposed in [6] which is called Multidimensional Hidden Markov Models with Independent Process. In this approach, we consider that at a given stage t the HMM generates R different symbols ($R = 3$ in the color representation space used) and not only one anymore. These R processes share the same states with the same transitions (so the phenomenon is characterized by an only one A matrix) but symbols are specific to each process (so the model contains R matrixes B). The characterization of the HMM architecture contains the following additional elements: the number R of processes linked to the Multidimensional HMM, the set V^r (with cardinal M^r) of symbols linked to the process P_r, the matrix B^r giving probabilities of generation of symbols linked to the process P_r, the set V (with cardinal R) of symbols dictionaries V^r linked to each process, and finally the set B (with cardinal R) of matrixes B^r giving probabilities of symbols generation linked to each process.

We have presented the Multidimensional HMM as introduced in [6]. We use this extension of HMM in both learning and tracking steps and we will now describe these two steps.

3. OBJECT LEARNING BASED ON GHOSP ALGORITHM

The object learning phase is crucial in any object tracking system and its quality has to be as high as possible. In this paper, we propose to perform object learning offline using the GHOSP algorithm. In this section we will first describe briefly the GHOSP algorithm and then how we use it in our object tracking system.

The GHOSP (Genetic Hybrid Optimization & Search of Parameters) algorithm [8] results of an hybridisation with a genetic algorithm [9] and the Baum-Welch algorithm [10] dedicated to HMM optimisation. This combination comes from the fact that even if it is very often used, the Baum-Welch algorithm has the disadvantage to be trapped in local optima in the search space, which is not the case for genetic algorithms. The architecture of genetic algorithms involves genes, chromosomes, and several operators: evaluation of fitness, crossover, and mutation. When applied on HMM, genes are taken from the space of probability, chromosomes are the reorganisation in a linear vector of all coefficients stored in A, B, and Π, evaluation is performed by the Forward algorithm, and finally crossover and mutation consist respectively in combination and modification of individuals according to certain constraints. All this is more detailed in [8]. In our approach, we use the multidimensional version of the GHOSP algorithm, which involves a multidimensional version of the Baum-Welch algorithm.

The GHOSP algorithm (in its multidimensional version) gives appropriate multidimensional HMM from a set of multidimensional observations used as learning examples. In our case, these observations are taken from RGB color images which contain object to be learnt. We have to process these images in order to obtain the observation data sent to the GHOSP algorithm. So we consider each color component separately and we sample the intensities of pixels belonging to the three gray level images (one for each color component) in order to define a reasonable set of symbols. So the intensity or value v of every pixel is converted from $[0, 255]$ into a symbol s using the following formula:

$$s = \left\lfloor v \cdot \frac{M^r - 1}{255} \right\rfloor \tag{1}$$

where M^r is the number of possible symbols and $\lfloor x \rfloor$ represents the integer part of x. From this sampling we obtain a three component vector which will be used as input in the GHOSP algorithm.

We then apply the GHOSP algorithm on observations taken from learning images and we obtain a matrix A, a matrix Π, and a set B of three matrixes B^R, B^G, B^B. To solve the problem of unsufficient data in the learning corpus, we use the technique presented in [11]: we add a small constant ($\varepsilon = 0.1$) to every term of the matrixes and we normalise them in order to keep their stochastic properties.

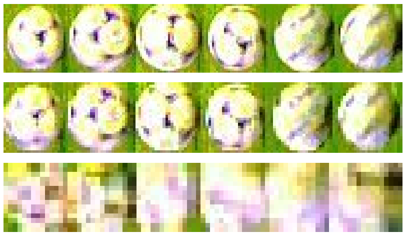

Fig. 1. Images used in learning step. Different object sizes have been tested: from top to bottom (at different scales) images are composed of 34×34, 26×26, and 15×15 pixels

Parameter	Value
Minimum number of states	4
Maximum number of states	6
Number of symbols	11
Number of iterations in the Baum-Welch algorithm	3
Population size in the genetic algorithm	100
Number of parents in the population	90
Number of iterations in the genetic algorithm	80
Number of dimensions	3

Table 1. Parameters of GHOSP algorithm.

Learning images represent the object to be tracked in several positions, with several sizes, and using different viewpoints and lighting conditions. It allows the GHOSP algorithm to generate a more generical HMM which will better represent the object to be tracked. Examples of images and parameters used in learning phase are respectively given in figure 1 and table 1. Among these parameters, population size represents the number of solutions explored at each iteration and number of parents corresponds to the number of best solutions used as starting point for next exploration.

We have presented in this section how we perform object learning offline. Once the object has been learnt, it can be tracked into any video sequence using the multidimensional HMM built with an approach based on Forward algorithm. This method will now be presented.

4. OBJECT TRACKING BASED ON FORWARD ALGORITHM

The learning phase described in previous section allows us to obtain a multidimensional HMM representing the object to be tracked. In this paper, we propose to perform object tracking in two steps: first the approximate object position is predicted using a simple motion estimator (rather than HMM as in HMM-based object tracking approaches) and then the exact object position is searched. This search can be seen as an object detection task in a restricted area and will be done using HMM built in learning phase. In this section we use the Forward algorithm [12] which is a well-known algorithm in the field of HMM. We describe in this section the three processings we perform on every frame of the video sequence in order to track the learned object: determination of the tracking state (object lost or not), approximate position prediction, and finally exact position search.

In every frame, it is first necessary to determine wether the

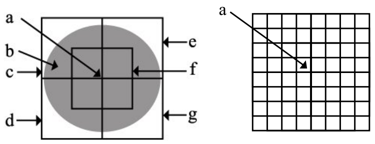

Fig. 2. **Left:** Windows used to compute thresholds on the first image. Position (*a*) of object (*b*) is considered as known. The 5 windows are represented by symbols (*c*) to (*g*). **Right:** Windows built from the predicted position (*a*).

tracked object has been lost or not by using information from previous frames. In a given frame t, we divide the image area to analyse into several windows $W_i(t)$ of size $w \times h$. The forward algorithm [12] is applied on each window $W_i(t)$ to give a global score $P_i(t)$, using the HMM built in the learning phase. We define then a value $p_i(t)$ linked with the fact that the content of window $W_i(t)$ has been generated by the HMM. Considering identical values $p_i(t)$, we would obtain:

$$p_i(t) \approx e^{\frac{\ln(P_i(t))}{n \cdot w \cdot h}} \tag{2}$$

where the HMM dimension is equal to n. So $p_i(t)$ can be seen as a normalisation with respect to observation size and dimension. A high value of $p_i(t)$ means that the associated window $W_i(t)$ represents relatively well the learned object. So we look for the window $W_{\text{max}}(t)$ characterized by the highest value $p_{\text{max}}(t)$. Ideally, the value $p_{\text{max}}(t)$ should be equal to 1 and all windows characterized by values $p_i(t)$ far from $p_{\text{max}}(t)$ should not contain the tracked object. However, we have to consider a tolerance to errors due to noise or motion effects. So we consider that the reference value $p_{\text{ref}}(t)$ in a given frame t will not be equal to 1 but rather to $p_{\text{max}}(t-1)$, that is to say the value associated with the window containing the object in the previous frame $t-1$. From this measure, it is then possible to define an interval $[s_{\text{low}}(t), s_{\text{high}}(t)]$ outside which we assume the object has been lost:

$$\begin{aligned} s_{\text{low}}(t) &= \alpha_{\text{low}} \cdot p_{\text{max}}(t-1) \quad (3) \\ s_{\text{high}}(t) &= \alpha_{\text{high}} \cdot p_{\text{max}}(t-1) \quad (4) \end{aligned}$$

where α_{low} and α_{high} are two coefficients used to qualify the tolerance to errors. Specific processing is necessary for the first frame, where no value $p_{\text{max}}(t-1)$ is available. So in this frame, we create five windows around the object initial position, as shown in left part of figure 2. From these windows, we can compute the maximum value p_{max}, which will be used in the second frame.

Contrary to classical HMM-based approaches, we are using a simple and low computational motion estimator to predict object position. Estimation accuracy is not crucial as result will be used as initialisation for an exact position search performed using HMM. The model is based on results from the two previous frames:

$$\begin{aligned} x(t) &= x(t-1) + c \cdot (x(t-1) - x(t-2)) \quad (5) \\ y(t) &= y(t-1) + c \cdot (y(t-1) - y(t-2)) \quad (6) \end{aligned}$$

The position $(x(t), y(t))$ is then used in the exact position search described further. The constant speed hypothesis is defined by $c = 1$. If the object has not been found (when its speed is not constant), we consider several cases successively: a deceleration ($c = 1/2$), an acceleration ($c = 2$), or a stop of the object ($c = 0$). If the object is not found in any case, we assume it is temporarly occluded. The tracking process is stopped if the occlusion phenomenon persists for a given number of successive frames.

The approximate position predicted using one of the motion models described previously is then considered as initialisation for the exact position search limited in a local window of the current frame around the predicted position. This window is divided in 64 subwindows of dimension l-by-l pixels as shown in right part of figure 2. Every subwindow is then transformed into multidimensional observation using a sampling process relatively similar to the one used in learning phase. We then apply Forward algorithm (in its multidimensional version) on each of the 64 observations using the HMM built in the learning phase. We obtain for each window $W_i(t)$ a score $P_i(t)$ from which we compute the value $p_i(t)$. It allows us to deal with different sizes between windows $W_i(t)$ and images from the learning dataset.

Among the windows characterised by a value $p_i(t)$ belonging to the interval defined by $[s_{\text{low}}(t), s_{\text{high}}(t)]$, we only keep the four highest ones. If no value respects this condition, we consider that the searched object has not been found and the search of the exact position is initialised using another position estimation (with another value for c) as decribed previously. If only one value $p_i(t)$ belongs to interval $[s_{\text{low}}(t), s_{\text{high}}(t)]$, the exact object position is determined by the center of the window $W_i(t)$ for which the value $p_i(t)$ has been obtained. Finally, if several values $p_i(t)$ are contained in the interval, the object position $C(t)$ is computed as a linear combination of window centers for which the values $p_i(t)$ have been obtained:

$$C(t) = \frac{\sum\limits_i p_i(t) \cdot c_i(t)}{\sum\limits_i p_i(t)} \tag{7}$$

where $c_i(t)$ represents the center of the window $W_i(t)$ characterized by the value $p_i(t)$.

5. RESULTS

The method proposed here has been tested on image sequences obtained from TV videos of football games. The goal is to track the ball for the complete length of a shot. Most of the tests have been realised on sequences containing more than one hundred images, as shown in figure 3. The average processing time is about 0.1 second per frame, so the tracking can be performed at a frame rate equal to 10 Hz. Considering a dataset of more than 2500 images, the quality of the proposed method has been evaluated to 87 %.

Even if the method uses a simple motion model with constant speed, objects moving with a non linear motion can be followed precisely, as shown in figure 4. Occlusion is one of the most important problems in the field of object tracking. Our method is able to track objects temporary occluded. An occlusion example is presented in figure 5.

In case of presence in the scene of objects similar to tracked object, the method can lack of robustness. In figure 6, the ball and the sock are relatively similar, and obtained position is incorrect. A verification step using morphological methods will be necessary to solve this problem. The method is also unable to perform tracking correctly when the tracked object size is too small. In this case, the learning phase cannot be performed correctly due to the lack of learning data (object size is less than 25 pixels).

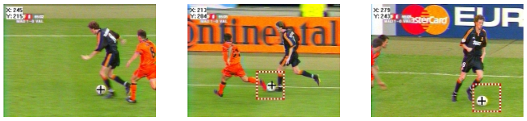

Fig. 3. Object tracking in a sequence containing more than 100 frames (frames 1, 44, and 104).

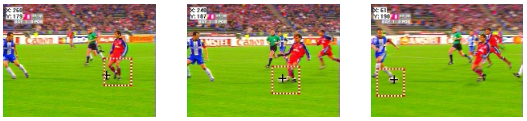

Fig. 4. Tracking of an object with non linear motion (frames 18, 22, and 26).

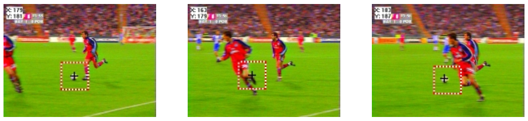

Fig. 5. Object tracking in presence of an occlusion phenomenon (frames 20, 24, and 28).

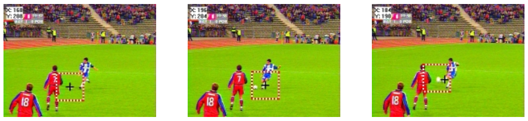

Fig. 6. Incorrect tracking of an object in a scene containing similar objects (frames 9, 12, and 16).

6. CONCLUSION

In this paper, we propose a new way to use HMM to perform object tracking. Contrary to classical approaches, HMM are used to model data in the spatial domain rather than in the temporal domain. More precisely, Multidimensional Hidden Markov Models are used to deal with color images as they contain multidimensional data. Objects are first learned offline using the GHOSP algorithm and then tracked using a two-step approach. First an approximate object position is predicted using a simple motion estimator, then the exact object position is computed. This last step can be seen as an object recognition problem and is solved using statistical pattern recognition tools. Indeed we propose to use HMM built in learning phase in an original approach based on the well-known Forward algorithm. The proposed method is characterized by a low computation cost.

Among perspectives, we consider to implement the proposed approach on a multiprocessors workstation in order to track object in true real time (25 images per second). We also think to use different color spaces or to involve shape information in order to increase robustness in case of presence of similar objects in the analysed images. Furthermore, it is necessary to improve the method so it is able to track very small objets. This will allow the tracking method to be integrated into a football game video sequence analysis system. Indeed the ball will then be tracked in all kind of shots, either close or far. Finally, we will apply the proposed tracking approach to other kind of video sequences, such as auto racing TV broadcasts or digital microscopy.

7. REFERENCES

[1] G. Rigoll, S. Eickeler, and S. Müller, "Person tracking in real-world scenarios using statisticals methods," in *IEEE International Conference on Automatic Face and Gesture Recognition – FG'2000*, Grenoble, France, March 2000, pp. 342–347.

[2] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, September 1999.

[3] H. H. Bui, S. Venkatesh, and G. West, "Tracking and surveillance in wide-area spatial environments using the abstract hidden markov model," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 177–195, February 2001.

[4] N. Giordana and W. Pieczynski, "Estimation of generalized multisensor hidden markov chains and unsupervised image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 465–475, May 1997.

[5] R. J. Stevens, A. F. Lehar, and F. H. Perston, "Manipulation and presentation of multidimensional image data using the peano scan," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 5, pp. 520–526, September 1983.

[6] T. Brouard, *Algorithmes Hybrides d'Apprentissage de Chaînes de Markov Cachées: Conception et Applications à la Reconnaissance de Formes (in french)*, Ph.D. thesis, University of Tours, France, January 1999.

[7] J. Yang, Y. Xu, and C. S. Chen, "Hidden markov model approach to skill learning and its application to telerobotics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 621–631, 1994.

[8] T. Brouard, M. Slimane, J. P. Asselin De Beauville, and G. Venturini, "Hybrid genetic learning of hidden markov models," in *AIDRI Conference on LEARNING – From Natural Principles to Artificial Methods*, Geneva, Switzerland, June 1997, pp. 127–130.

[9] J. H. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, MI, 1975.

[10] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology," *Bull. American Society*, vol. 73, pp. 360–363, 1967.

[11] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of theory of probabilistic functions of a markov process to automatic speech recognition," *Bell Systems Technical Journal*, vol. 62, no. 4, pp. 1035–1074, Apr. 1983.

[12] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.