HIERARCHICAL FEATURE FUSION FOR VISUAL TRACKING

Alexandros Makris[1], Dimitrios Kosmopoulos[1], Stavros Perantonis[1], Sergios Theodoridis[2]

[1] NCSR Demokritos, Intitute for Informatics and Telecommunications, Computational Intelligence Laboratory, 15310, Aghia Paraskevi, Athens, Greece {amakris,dkosmo,sper}@iit.demokritos.gr
[2] University of Athens, Department of Informatics, 15771 Athens, Greece stheodor@di.uoa.gr

ABSTRACT

A new method for object tracking in video sequences is presented. This method exploits the benefits of particle filters to tackle the multimodal distributions emerging from cluttered scenes. The tracked object is described by several models of different complexity, which are probabilistically linked together. The parameter update for each model takes place hierarchically so that the simpler models, which are updated first, can guide the search in the parameter space of the more complex models to relevant regions. This strategy improves the target representation because of the multiple models and reduces the overall complexity. The likelihood for each object model is calculated using one or more visual cues thus increasing the robustness of the proposed algorithm. Our method is evaluated by fusing on salient points and contour models and we demonstrate its effectiveness.

Index Terms— *tracking, sequential Monte Carlo.*

1. INTRODUCTION

The problem of tracking consists of finding in the consecutive frames the location of a given scene object, which might be in a heavily cluttered environment, with varying illumination conditions and possible partial or full occlusions. In this work we are going to address the problem with Bayesian methods and more specifically the Sequential Monte Carlo (SMC) approximation methods [1], [2], [3], [4]. These methods are probabilistic and treat the location of the tracked object as a probability density function, which they attempt to estimate by drawing samples from it. The basic elements that those methods require are: an object model, a dynamic model and an observation model. The object model is the internal representation of the target. Its type varies according to the application, the most common being the contour, the bounding box or high level object specific models (e.g., human body model). Successful tracking is the correct estimation of the object model state. The dynamic model is used to predict the next state given the current one. The observation model links the object model to the data by calculating the likelihood of the object given the state.

The SMC methods mentioned above, also known as particle filters, are a very popular approach to tracking [10], [11]. Their main advantage lies in their ability to cope with multimodal distributions such as those emerging from a cluttered environment. Their simplicity and low computational cost also contributes to their success. Additionally, they can be used to easily fuse different cues, a fact that we have exploited in this work. There are many works in the literature using particle filters with a single cue. The most commonly used cues in these approaches are the edges [10], the color and texture [7], [8], regions (blobs) [9], and motion information [6]. However, these approaches can only be applied under certain conditions, due to their incomplete object model. Contour trackers, for example, loose track when many clutter edges are present. Color histogram based methods perform poorly in the existence of many similar colored objects. To overcome these difficulties, several approaches for feature fusion have been recently appeared [12], [13], [14], [15], [18], [19], [20], [21]. Their goal is to achieve robust tracking by combining several of the above mentioned cues. However, most of these works have high complexity. These approaches differ in the way they fuse the cues. In [14], democratic voting is used to take a decision using the majority of the cues. A different approach uses color information as a proposal distribution to guide the main cue, which is the shape [12]. This way the method tries not to waste resources by searching in low likelihood areas. However, using solely color information is not adequate for the task because the background might contain similar colors. In [15], partitioned sampling is used to reduce the search in the state space while using several features (sound, motion and color). However, these cues are used to sample different state components and if one of them fails to locate the target region the rest will not be able to recover from the failure. In [20] instead of a dynamic model, motion is used to guide the particles in order to account for current measurements.

Our work follows the Bayesian tracking approach. We attempt to overcome the difficulties posed by the complex environment mentioned above by using several object models, which are updated hierarchically within the particle filtering framework. The algorithm is robust in various scene conditions and without any pre-adjustments selects and uses the best models for each situation. Each model uses several visual cues to define its likelihood function. In this paper, three cues are used: salient points within the object, the edges and the color histogram. To avoid the inefficient search in high dimensional spaces, the models are arranged in increasing complexity order and are updated hierarchically so that the simpler models are located first. This strategy enables efficient search in the parameter space and posterior estimation with significantly fewer samples despite the use of complex object models. Our framework enables the use of deterministic algorithms (e.g. KLT [5]) to track the simpler models (e.g., salient points).

The rest of the paper is structured as follows. In Section 2 our proposed method is described after a brief introduction to Bayesian tracking and the SMC approximation. In Section 3 the measurement model and its connection to the proposed algorithm is explained. Section 4 contains the experimental results followed by some conclusions and future work at Section 5.

1-4244-1437-7/07/$20.00 ©2007 IEEE

2. TRACKING ALGORITHM

In this section we provide the background for Bayesian tracking and the SMC methods which will be used to explain the proposed method. Let $\{x_t; t \in N\}$ be an unobserved state sequence representing the true position of the tracked object and $\{z_t; t \in N\}$ the observations for every time step, *t*. The Bayesian tracking consists of calculating the posterior $p(x_t / z_{1:t})$ at every step, given the measurements up to that step and a prior ($p(x_0) \equiv p(x_0 / z_0)$). At each step the solution is expressed as:

$$p(x_t / z_{1:t}) = \frac{p(z_t / x_t) p(x_t / z_{1:t-1})}{p(z_t / z_{1:t-1})} \quad (1)$$

In order to calculate the posterior we need to know the three terms involved in (1): Likelihood $p(z_t / x_t)$: This term can be calculated using the measurement model.

Evidence: $p(z_t / z_{1:t-1}) = \int p(z_t / x_t) p(x_t / z_{1:t-1}) dx_t$

Prior: . $p(x_t / z_{1:t-1}) = \int p(x_t / x_{t-1}) p(x_{t-1} / z_{1:t-1}) dx_{t-1}$

In most practical problems, including that of visual tracking, the analytical forms for the probabilities involved in the above relations are not available except for special cases (e.g., linear dynamics, Gaussian pdf's). Therefore, approximation methods are commonly used. One family of such methods is the SMC, which uses samples (particles) to estimate the involved pdf's [4]. Given *N* particles $\{\mathrm{x}_{t-1}^{(n)}\}_{n=1}^{N}$, at time *t-1*, which approximate the distribution $p(x_{t-1} / z_{1:t-1})$, the SMC methods compute *N* particles $\{\mathrm{x}_{t}^{(n)}\}_{n=1}^{N}$, which approximate the posterior $p(x_t / z_{1:t})$, at time *t* according to:

$$\hat{p}_N(x_t / z_t) = \frac{1}{N} \sum_{n=1}^{N} \delta_{x_t^{(n)}}(x) \quad (2)$$

The steps of the general Sampling Importance Resampling (SIR) [4] algorithm are:

- Sample from the proposal density function q:
  For $n = 1$ to N : Sample $x_t^{(n)}$ from $q(x_t / x_{0:t-1}^{(n)}, z_{0:t})$
- For each sample evaluate the importance weights:

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(z_t / x_t^{n}) p(x_t^{(n)} / x_{t-1}^{(n)})}{q(x_t^{(n)} / x_{0:t-1}^{(n)}, z_{0:t})} \quad (3)$$

- Resample by multiplying or discarding particles according to their weight so that the resulting particle set will be un-weighted and with the same number of particles.

A very common realization of this algorithm uses the transition prior $p(x_t / x_{t-1}^{(n)})$ as proposal distribution which means that the weights are updated by the likelihood $p(z_t / x_t^{(n)})$ [11].

The proposed algorithm is a particle filter based approach to the Bayesian tracking. As in SIR, the posterior is approximated by a set of samples which are propagated through time. This algorithm requires several measurement models of different complexity. These models are probabilistically linked, which means that, if we don't know the state of a model, we can evaluate its conditional probability given the states of the other models. For each model, one or more visual cues are used to define the likelihood. The models are arranged in increasing complexity order. The simpler ones (e.g. a set of salient points within the object, a set of blobs with the same color or texture) are first. More complex models, such as parametric curves or human models may follow. Simpler models are easier to update but they are not robust and do not offer a detailed target representation. In contrast, complex models are more difficult to update but, on the other hand, they offer a very detailed target representation and if they are supported by multiple visual cues they become very robust. Here we use one so-called main model to define the target region, which is the most complex one and is the last to be updated. When a new frame arrives, the rest of the models are updated first and because they are linked to the main model they provide information about its expected position. We should note here that the number of models might change during tracking. As we mentioned above, only the main model is required for the target representation. This fact allows us to flexibly add or remove auxiliary models during tracking without affecting the representation of the target, which is very helpful in varying scene conditions. In the case of salient points, for example, a metric for the quality of each point is used. When this falls below a threshold the point is discarded. When the number of tracked points is low, new points can be added. The higher order models define the target and the search for new points takes place in that area. If it is impossible to find any points, due to illumination conditions, for example, then this model can be completely removed. In the classical particle filtering approach to tracking, the state evolution is used to produce the new samples. However, for models with many parameters many samples are required to sample adequately from the state evolution. A better proposal distribution using some sort of low level information from the current frame can improve the sampling efficiency [12]. Here we propose a model hierarchy, where the simpler models narrow the search space for the more complicated ones. The state can be written as: $x = [x_{[1]}, x_{[2]}, ..., x_{[M]}]$ where *M* is the number of models. Each model, $x_{[i]}$, has *K_i* state parameters and the corresponding measurement parameters are $z_{[i]}, i = 1..M$. We assume that the conditional probabilities of a state of a model at time t, given the state of the others $p(x_{[i]t} / \{x_{[j]t}, j = 1..M, j \neq i\})$ are known. The likelihood $p(z_{[i]} / x_{[i]})$ and the state evolution $p(x_{[i]t} / x_{[i]t-1})$ for each model are also known. The update of the particle set for each model takes place in a predefined sequential fashion. When the particle set for a model is updated it is used to update the subsequent models. The steps of the algorithm are the following:

1. Update, using the SIR algorithm, the state of the first model. If the first model allows it a deterministic algorithm (e.g. KLT) can be used instead of SIR. The proposal density used is the state evolution so the weights are updated by the likelihood:

$$q(x_{[1]t} / x_{[1]0:t-1}^{(n)}, z_{[1]0:t}) = p(x_{[1]t} / x_{[1]t-1}^{(n)}) \quad (4)$$

$$w_{[1]t}^{(n)} \propto w_{[1]t-1}^{(n)} p(z_{[1]t} / x_{[1]t}^{(n)}) \quad (5)$$

In the case that KLT is used, the posterior is not approximated by particles but as a Gaussian centered at the position found by KLT.

2. For every other model, update the particles by sampling from the conditional distribution given the states of the previous models:

$$q(x_{[i]t} / x_{[i]0:t-1}^{(n)}, z_{[i]0:t}) = p(x_{[i]t} / x_{[i]t-1}^{(n)}) p(x_{[i]t} / x_{[1:i-1]t}) \quad (6)$$

$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(z_{[i]t} / x_{[i]t}^{(n)})}{p(x_{[i]t} / x_{[1:i-1]t})} \quad (7)$$

3. Re-Evaluate the likelihood of the models given their updated positions. Evaluate the performance of each model and remove those who have lost track. At this step, models with low likelihood for every particle are removed. From the current target representation, maintained by the main model initialize new models if possible. This function may not take place at every frame but only when few models remain active.

The proposed algorithm breaks the initial problem into M sub-problems. As mentioned above, the models are arranged in increasing complexity order so that the simpler are updated first, which then guide the more complicated ones in relevant regions. This way, fewer particles are required to search efficiently the state space, leading to lower computational complexity and allowing the algorithm to be able to run in real time.

3. APPLICATION: FUSION OF POINTS AND CONTOUR

The proposed algorithm has the following requirements: i) the models' complexity has to vary significantly; ii) one of the models, should define the target area (main model) iii) the conditional probabilities of a model, given the states of the rest, must be defined. Here we apply two models, the first one is simple and is a set of salient points (corners) within the object; the second one is more complex and is the contour of the tracked object (main model). The combined state vector becomes: $x = [x_{[SP]}, x_{[C]}]$, where $x_{[SP]}$ represents the salient points and $x_{[C]}$ represents the contour curve. Each of those models may use several visual cues to define the likelihood. In this work, we use two cues for the contour (which is represented as a spline curve), the edge information and the color histogram. We first describe how we calculate the likelihood for each of these models and afterwards we will link them together by showing how the conditional probabilities are calculated.

Salient Points: The feature is the weighted mean, of the tracked points. For each tracked object, N_p points are used but each one of them is updated independently so the dimension of the search space does not increase with the number of points. The likelihood of this feature is determined by calculating the sum of squared differences of a rectangle around a candidate feature and the original. The parameters passed to the algorithm are the weighted mean point coordinates:

$$x_{[SP]} = \sum_{i=1}^{Np} v(i) L_{SSD}(i) \quad (8)$$

Where $v(i)$ are the image coordinates and $L_{SSD}(i)$ the likelihood of the *i*-th point. The likelihood for this model is:

$$\mathrm{L_p} = p(z_{[SP]} / x_{[SP]}) = \prod_{i=1}^{Np} L_{SSD}(i) \quad (9)$$

where $z_{[sp]}$ the mean of all measured points.

As mentioned in requirement iii), we need the conditional probability of the curve's position given the characteristic points. We assume that these points belong to the object so they must lie inside the curve. However, the object being tracked might not be rigid so these points might move relatively to the curve; therefore the model linking the points with the curve cannot be deterministic. Through the experiments we concluded that a simple Gaussian model is adequate to link them. Since we know the relative positions of the points and the curve, D_{cp}, in the initial frame, we can calculate the estimated position of the curve at each step given the updated point positions and then sample from a Gaussian around this position:

$$p(x_{[C]t} / x_{[SP]t}) = \frac{1}{\sigma_r \sqrt{2\pi}} \exp\left\{ -\frac{[x_{[SP]t} - D_{cp}]^2}{2\sigma_r^{\,2}} \right\} \quad (10)$$

Contour: Here the total likelihood is estimated by a) considering how well the edges fit the current contour (exponential function of edge distance from curve) b) comparing, the histogram of the object that is surrounded by the contour with the template's histogram using the Bhattacharyya distance. The total curve likelihood, L_C, is the product of edge and histogram likelihoods L_{CH}, L_{CE}: $p(z_{[C]} / x_{[C]}) = L_C = L_{CH} L_{CE}$ (11)

A dynamic model is required to update the curve's positions. Unless KLT is used, a dynamic model is also required to update the point's positions. We assume that the frame rate of the video is high so that the positions of the curves at two consecutive frames are close to each other. The model used here is again a Gaussian around the previous curve's position. This model is very simple but behaves very well with the proposed algorithm since it is used along with the information from the current point positions.

4. EXPERIMENTS

The experiments have been made in various grayscale videos and various objects have been tracked to verify the versatility of our tracker using the configuration of the previous section. The KLT algorithm has been used to update the points' positions. The edge and the intensity histogram were used to define the contour's likelihood. The test videos range from simple static camera sequences to very complex ones with moving camera, heavy clutter, illumination changes and partial occlusions. We compared our approach with the classical particle filter [11] and also with a modification which is based upon the classical but uses both edge and intensity histogram to calculate the likelihood and is similar to [21] in the way the cues are fused together though the later also incorporates structure information. The classical algorithm works well only in situations where no significant clutter exists and the target moves rather slowly. The approach that uses the histogram as well is more robust but outperformed by our method.

In Figure 1 the benefits of the constant update of the salient point set are illustrated (the initial points are occluded as the object rotates). Our approach outperforms the other methods in that sequence because much less particles are required (50 versus 500(classical) and 250(histogram)) and consequently the tracking is much faster (120 versus 30-45 fps). The sequence of Figure 2, is more challenging. The fast movement and the abrupt shape change of the target along with the heavily cluttered background lead the other methods to failure. Our methodology succeeded because of the collaboration between the two models, points and contour.

Apart from qualitative we have also conducted quantitative experiments, which is very rare in the related literature probably due to lack of standard ground truth data. We have annotated several sequences by hand and we have calculated the following measures, tracker Detection Rate (TDR) = TP / (TP + FN), and false Alarm Rate (FAR) = FP / (FP + TP), where TP is the number of true positive pixels, FN the false negative and FP the false positive pixels. The target area was defined by a bounding box. In Table 1 we show the results on several sequences compared to the particle filter that uses edges and histogram. Because of the challenging nature of the selected sequences the classical particle filter using only edge information always failed after a few frames and thus is not included in the results. The 'caviar' sequences mentioned on Table 1 are taken from CAVIAR project [17] (corridor view from the Shopping Center), which provides ground truth. The nature of the data didn't allow the tracking of many salient points; therefore our method only slightly outperformed the other. The 'tennis' are challenging sequences similar to that of Figure 2. The 'cars' are sequences taken from traffic cameras.

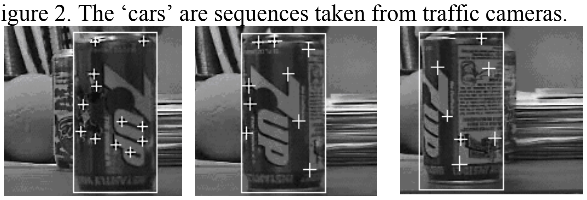

Figure 1. Sequence with self occlusion. (50 particles, 120fps). Frames 1, 200, and 350.

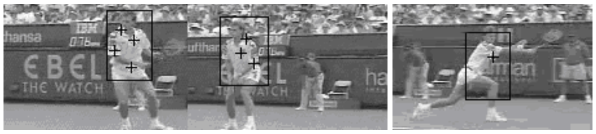

Figure 2. Tennis sequence (150 particles, 50fps). Frames 1, 100, and 200.

Sequences			Proposed Method			Particle Filter		
Theme	**Clips**	**Frames**	**TDR**	**FAR**	**N_p**	**TDR**	**FAR**	**N_p**
Tennis	5	800	87	16	150	20	90	500
Cars	4	1200	91	11	100	80	29	500
Caviar	20	5000	92	12	90	88	21	500
Total	**29**	**7000**	**91**	**12**		**78**	**30**	

Table 1. Tracking results of our method compared to the classical particle filter. N_p is the number of particles used.

5. CONCLUSIONS

In this paper we proposed a particle filtering based tracking algorithm. The algorithm enhances significantly previous approaches in terms of robustness and speed by fusing several cues hierarchically in order to achieve robust tracking in various situations. To guide the search in the state space, several object models are used. The simpler ones are updated first and the subsequent use the information from the previous. Our experiments demonstrate that our framework is robust and fast. It handles occlusion, cluttered background and fast changes in position and appearance of the target very well. The proposed methodology can be extended by using more object models and by fusing additional visual cues.

Acknowledgement: This work has been supported by the 03ED698 and SP-IAL-91 research projects,(PENED and Image-Audio-Language) co-financed by National and Community Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology and 75% from E.U.).

6. REFERENCES

[1] Neal, R. M. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.

[2] Liu, J. and Chen, R. Sequential Monte Carlo methods for dynamic systems. J. Amer. Statist. Assoc., 93:1032–1044, 1998.

[3] Doucet, A., Godsill, S.,&Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing, 10:3, 197–208, 2000.

[4] Andrieu, C., de Freitas, N., Doucet, A., Jordan, M. I. An introduction to MCMC for machine learning. Machine Learning, vol. 50, pp. 5--43, Jan. - Feb. 2003.

[5] Shi, J., Tomasi, C.,. Good Features to Track, IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), 1994.

[6] Sidenbladh, H., Black, M. J., and Fleet, D. J. 2000. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In Proceedings of the 6th European Conference on Computer Vision-Part II (June 26 - July 01, 2000).

[7] Perez, P., Hue, C., Vermaak, J., Gangnet, M.. Color-based Probabilistic Tracking. Proc. Eur. Conf. on Comp. Vis. ,DK, 2002.

[8] Chen, H.T., Liu, T.L., and Fuh, C.S.. Probabilistic Tracking with Adaptive Feature Selection. Proc. of the 17th Intl. Conf. on Pattern Recognition Cambridge, UK, vol. 2 736-739, Aug,2004.

[9] Isard, M., MacCormick, J.. BraMBLe: A Bayesian Multiple-Blob Tracker Proc.Int.Conf. Computer Vision, vol. 2 34-41, 2001.

[10] Gordon, N. J., Salmond, D. J., and Smith, A. F. M.,. Novel approach to nonlinear/non-gaussian bayesian state estimation. IEE Proc. on Radar and Signal Processing, vol.140, 107-113, 1993.

[11] Isard,M.,Blake,A. Condensation-Conditional density propagation for visual tracking. Int.Journ. of Comp.Vision,29:5–28, 1998

[12] Isard, M. and Blake, A. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In Proc. Of Europ. Conf. on Computer Vision, vol. 1, pp. 767–781, 1998.

[13] MacCormick, J. and Isard, M. Partitioned sampling, articulated objects, and interface-quality hand tracking. In Proc. of European Conf. on Computer Vision, vol. 2, pp. 3–19, 2000.

[14] Spengler, M., Schiele, B. Towards robust multi-cue integration for visual tracking. In: International Workshop on Computer Vision Systems. (2001) 94–107

[15] Perez, P., Vermaak, J., Blake, A. Data Fusion for Visual Tracking with Particles. Proceedings of IEEE (issue on State Estimation), vol 92, issue 3, 495- 513, 2004.

[16]Wu, Y., Huang, T. S. Robust Visual Tracking by Integrating Multiple Cues Based on Co-Inference Learning, International Journal of Computer Vision 58(1), 55–71, 2004.

[17] EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[18] Moreno-Noguer F., Sanfeliu A., Samaras D. Fusion of a Multiple Hypotheses Color Model and Deformable Contours for Figure Ground Segmentation in Dynamic Environments. In Proc. 3rd IEEE Workshop on Articulated and Nonrigid Motion, ANM'04, (in conjunction with CVPR'04), 2004.

[19] Bohyung H. Davis L., Robust observations for object tracking, Image Processing, 2005. ICIP 2005. IEEE International Conference on , vol.2, no.pp. II- 442-5, 11-14 Sept. 2005

[20] Odobez, J.-M.; Gatica-Perez, D.; Ba, S.O., "Embedding Motion in Model-Based Stochastic Tracking," Image Processing, IEEE Transactions on , vol.15, no.11pp. 3514- 3530, Nov. 2006.

[21] Li, P., Chaumette, F., Image Cues Fusion for Object Tracking Based on Particle Filter, AMDO,99-107, 2004.