A multimedia content modeling and classification methodology using visual information for the protection of sensitive user groups.

Alexandros Makris [*]

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
amakris@di.uoa.gr

Abstract. The thesis concerns the problems of visual tracking and violence detection in video sequences. For the visual tracking problem, two feature fusion frameworks are presented. For violence detection, a system that classifies movie segments as violent or non-violent is proposed. The first tracking framework called *'Model Fusion via Proposal' (MFP)* framework, provides a way to efficiently fuse visual cues using independent trackers to construct an improved proposal distribution for the main tracker. The fusion method results in reduced computational requirements due to the better proposal and the gradual exploitation of the state space. The *'Hierarchical Model Fusion' (HMF)* framework, extends the MFP framework by integrating the multiple models into a single tracker which exploits all the visual cues. This way the robustness of the approach is further increased. To this end, we extended the Bayesian framework to allow the integration of multiple models and we derived a particle filtering based approximation algorithm which allows the efficient integration of complementary models of different complexity with redundancy of information. Both tracking frameworks use multiple object models to describe the target. This feature enables the development of an adaptation strategy which adds or deletes models to cope with target appearance changes. For violence detection, a system that classifies movie segments as violent or non-violent is proposed. The system fuses audio and visual information. The audio module uses state-of-the-art methods. The visual features concern the general motion in the scene, the detection of gunshots, and the motion of the detected people.

1 Introduction

1.1 Automated Video Understanding

A very active research area, automated video understanding, concerns the analysis of video to extract semantic information. Most vision systems adopt a bottom-up approach. First low level tasks such as motion segmentation, object recognition, and tracking are performed followed by scene analysis or event recognition

[*] Dissertation Advisor: Professor Sergios Theodoridis

to extract high level concepts about the scene. Its applications lie in the fields of surveillance, control, and analysis. *Motion segmentation* is usually the first step in surveillance systems and consists of detecting the objects of interest and segmenting them from the static background. *Object recognition* is an important step required for video understanding. The recognition may concern specific objects (e.g. a face of a specific person [15], [14], [3]) or object classes (e.g. vehicles, people [2], [4]). The objects can be described by different cues. The next crucial step in vision systems consists of *tracking* the objects of interest extracted by motion segmentation or by object recognition. This is essential to establish the correspondences between the detected objects from frame to frame and possibly reduce the computational cost by avoiding detecting the objects in every frame. Usually the detection process is much more complex than tracking. *Event recognition* is the step that naturally follows the tracking of the objects of interest [7], [12].

1.2 Contribution

The contribution of this thesis concerns the development of novel tracking algorithms based on the particle filtering framework. Furthermore, a method for violence detection in movies is presented, using a classification approach, with features stemming from the tracking of objects using the proposed algorithm and from other computer vision techniques. The work resulted in the following publications: [9], [11], [1].

Proposed Tracking Frameworks The most important issues of the PFs are the efficient and information-rich *target representation* and the selection of the *proposal distribution*. We tackle these issues by proposing two generic frameworks (Model Fusion via Proposal(MFP), Hierarchical Model Fusion(HMF)) for fusing visual cues within the Bayesian framework. The MFP framework, provides a way to efficiently fuse visual cues using independent trackers to construct an improved proposal distribution for the main tracker. The fusion method results in reduced computational requirements due to the better proposal and the gradual exploitation of the state space. The HMF framework, extends the MFP framework by integrating the multiple models into a single tracker which exploits all the visual cues. This way the robustness of the approach is further increased. To this end, we extended the Bayesian framework to allow the integration of multiple models and we derived a particle filtering based approximation algorithm which allows the efficient integration of complementary models of different complexity with redundancy of information. Additionally, we developed an adaptation technique, to automatically delete and re-initialize the auxiliary models.

Proposed Violence Detection Method A system that classifies movie segments as violent or non-violent is proposed. The system fuses audio and visual information to increase the robustness. Two independent modules for audio and video based classification were developed. The audio module has been developed in [5] where more details can be found. The video module uses features which describe the amount and the direction of motion in the scene, the motion of the detected people, and the illumination variations caused by gunshots. The features are used to classify the segments in three activity classes according to the amount of human activity of the segment (no, normal, and high activity) and to two classes according to the existence or not of gunshots in the scene. Similarly, The audio module uses several features to classify the movie segments in one of several audio classes (e.g. music, speech, gunshots, fights). The output of the video and audio classifiers is fed to a meta-classifier which decides for the presence of violence in the segment. The system as well as the independent modules were tested in real movie dataset. Both the modules when used independently reach a satisfactory level of performance. The fusion methods boosts that performance resulting in a system that detects about 4 out of 5 violent events (80% recall) and about 1 out of 2 events classified as violent are indeed violent (50% precision).

2 Hierarchical Model Fusion Framework

2.1 Algorithm Description

In the HMF framework the target is represented by several models of increasing dimension, which are probabilistically linked. The parameter update for each model takes place hierarchically so that the simpler models, which are updated first, guide the search in the state space of the more complex models to relevant regions. The most complicated model (in terms of state dimension) and the last in hierarchy, is called main model and its parameters fully describe the target. The rest of the models are referred as auxiliary as the estimation of their state is not required by the application.

A simple example (see Figure 1) will clarify the proposed concept. Let us consider a case of a target of which we want to estimate the bounding box. We will use two models, an auxiliary that tracks a feature point in the target and the main model, the bounding rectangle. The state of the first model has two parameters, the point's coordinates $x_s = [i_{s_x}; i_{s_y}]$, while the rectangle model has three, the coordinates of its center and a scale parameter, $x_b = [i_{b_x}; i_{b_y}; s_b]$. When the tracking is initialized the relative position of the rectangle's center and the point's is measured. If the tracked object is rigid this relative position should be almost constant between two consecutive frames. Thus if the location of the feature point is found on the next frame we can infer the coordinates of the center of the rectangle. The advantage of this strategy is that we first search in a two-dimensional space for the feature point and then we search in an one-dimensional space for the scale instead of searching in a three-dimensional state space to locate the rectangle directly. One should argue here that the coordinates obtained from the feature point model might not be very accurate or that this strategy will fail for non rigid objects. These issues are addressed by relaxing the link between the two models which will be discussed in detail in the following.

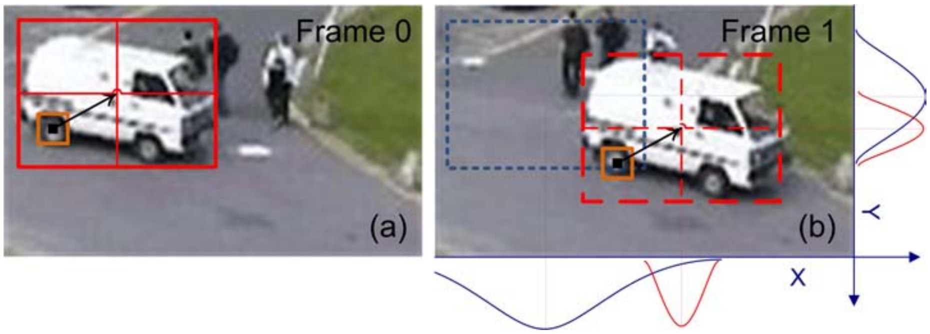

Fig. 1. In (a) the tracking is initialized with two models describing the target, a bounding rectangle and a salient point. The arrow shows the relative position of the salient point and the center of the rectangle. In (b) the position of the point is updated and using the stored relative distance the proposal for the rectangle given this position is shown in the x and y axis (red). This proposal is much closer to the target than the proposal derived by the state evolution model of the rectangle (blue).

In the general case M object models are used for target representation. The state can be written as [1]:

$$\mathbf{x} = [\mathbf{x}_{[1]}; \mathbf{x}_{[2]}; \ldots; \mathbf{x}_{[M]}] \tag{1}$$

Where $\mathbf{x}_{[i]}$ are the state vectors of each object model. To each object model corresponds a measurement model with parameters $\mathbf{z}_{[i]}$. The graphical model of Figure 2c encodes the architecture of our framework. It depicts the following assumptions which we make to derive an algorithm for the recursive calculation of the posterior:

- The total likelihood is given by multiplying the likelihoods of individual models: $p(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^{M} p(\mathbf{z}_{[i]}|\mathbf{x}_{[i]})$.
- The state evolution is decomposed as:
  $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \prod_{i=1}^{M} p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t}))$.

where $Pa(\mathbf{x}_{[i]t})$ denotes the parent nodes of $\mathbf{x}_{[i]t}$.

To construct algorithms that will be able to update the posterior of each model sequentially we derive the following equation which is an extension to the classical Bayesian tracking equation, but takes place in M steps. Each step updates the state of the corresponding model. Using simple probability rules and the assumptions mentioned above the filtering equation for the $i-th$ step is given by:

$$\begin{aligned} &p(\mathbf{x}_{[1:i]t}, \mathbf{x}_{0:t-1}|\mathbf{z}_{[1:i]t}, \mathbf{z}_{1:t-1}) = \\ &= p(\mathbf{x}_{[1:i-1]t}, \mathbf{x}_{0:t-1}|\mathbf{z}_{[1:i-1]t}, \mathbf{z}_{1:t-1}) \frac{p(\mathbf{z}_{[i]t}|\mathbf{x}_{[i]t}) p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t}))}{p(\mathbf{z}_{[i]t}|\mathbf{z}_{[1:i-1]t}, \mathbf{z}_{1:t-1})} \end{aligned} \tag{2}$$

[1] For notational clarity we avoid using the ^{T} superscript, instead we use the Matlab's $';'$ notation to concatenate vectors.

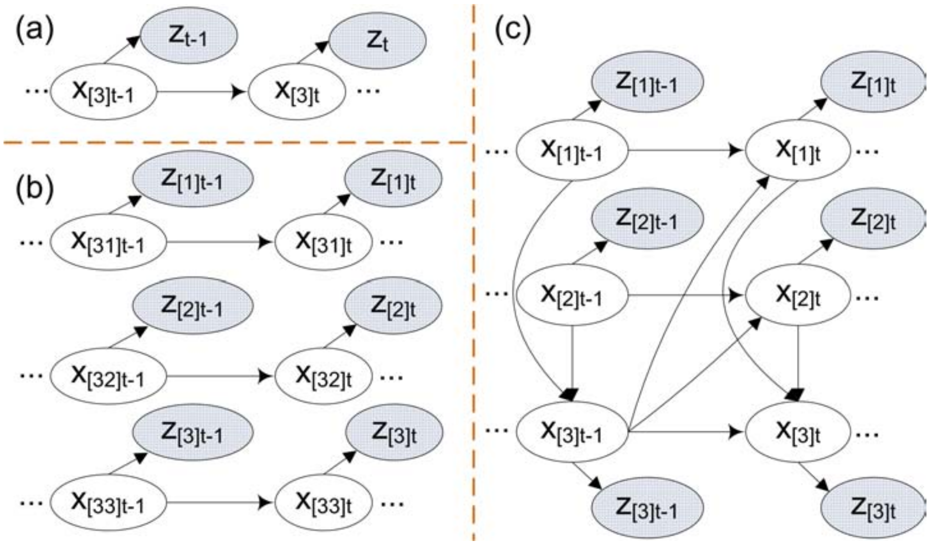

Fig. 2. Graphical Models depicting only slices $t-1$ and t of the temporal dimension. The state to be approximated is denoted as $x_{[3]}$. (a) Standard Particle Filter [8]. (b) Partitioned Sampling [10],[13], the state is partitioned in 3 parts ($x_{[31]}$, $x_{[32]}$, $x_{[33]}$), which are updated independently, each one depending on different measurements. (c) Proposed Graphical Model, using 2 auxiliary models ($x_{[1]}$, $x_{[2]}$) which are linked to the main model ($x_{[3]}$).

As mentioned above, equation (2) can be used to construct Bayesian tracking algorithms that use multiple object models. Here, we use it to construct a PF based algorithm to iteratively update the posterior of each model.

We assume that at time $t-1$ the posterior $p(\mathbf{x}_{[1:M]0:t-1}|\mathbf{z}_{[1:M]1:t-1})$ is approximated by a weighted particle set comprised of N weighted sample trajectories: $\{\mathbf{x}^{(n)}_{[1:M]0:t-1}, w^{(n)}_{[M]t-1}\}^{N}_{n=1}$. To update the particle set that approximate the posterior at time t we proceed in a sequential fashion. Each model is updated using the information from the already updated models at time t. The proposal distribution is selected to factorize as:

$$q(\mathbf{x}_{[1:i]t}, \mathbf{x}_{0:t-1}|\mathbf{z}_{[1:i]t}, \mathbf{z}_{1:t-1}) = \\ q(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t}), \mathbf{z}_{[i]t}) q(\mathbf{x}_{[1:i-1]t}, \mathbf{x}_{0:t-1}|\mathbf{z}_{[1:i-1]t}, \mathbf{z}_{1:t-1}) \quad (3)$$

As in standard PF the samples are drawn from the first factor of Eq. (3).

The weights are given by:

$$w^{(n)}_{[i]t} = \frac{p(\mathbf{x}^{(n)}_{[1:i]t}, \mathbf{x}^{(n)}_{0:t-1}|\mathbf{z}_{[1:i]t}, \mathbf{z}_{1:t-1})}{q(\mathbf{x}^{(n)}_{[1:i]t}, \mathbf{x}^{(n)}_{0:t-1}|\mathbf{z}_{[1:i]t}, \mathbf{z}_{1:t-1})} \quad (4)$$

By substituting equations (2) and (3) into (4) we get the following weight update equation:

$$w^{(n)}_{[i]t} \propto w^{(n)}_{[i]t-1} \frac{p(\mathbf{z}_{[i]t}|\mathbf{x}^{(n)}_{[i]t})p(\mathbf{x}^{(n)}_{[i]t}|Pa^{(n)}(\mathbf{x}_{[i]t}))}{q(\mathbf{x}^{(n)}_{[i]t}|Pa^{(n)}(\mathbf{x}_{[i]t}), \mathbf{z}_{[i]t})} \tag{5}$$

The steps of the iteration for the update of model i at time t of the proposed algorithm are the following (see Figure 3):
Given the particle set:
$\{\mathbf{x}^{(n)}_{[1:i-1]t}, \mathbf{x}^{(n)}_{0:t-1}, w^{(n)}_{[i]t}\}^{N}_{n=1}$:

1. **Sample:** For $n = 1$ to N draw $\mathbf{x}^{(n)}_{[i]t}$ from $q(\mathbf{x}^{(n)}_{[i]t}|Pa(\mathbf{x}_{[i]t})^{(n)}, \mathbf{z}_{[i]t})$.
2. **Update** the weights of each particle using Eq. (5).
3. **Normalize** the weights.
4. **Resample** the particle set according to its weights, so that the resulting particle set will be un-weighted and with the same number of particles.

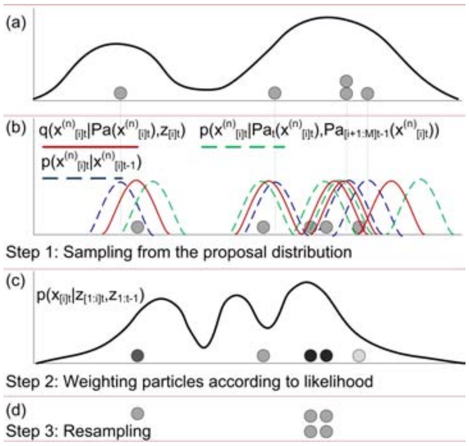
(a)
(b)
$q(x^{(n)}_{[i]t}|Pa(x^{(n)}_{[i]t}),z_{[i]t})$
$p(x^{(n)}_{[i]t}|Pa_t(x^{(n)}_{[i]t}),Pa_{[i+1:M]t-1}(x^{(n)}_{[i]t}))$
$p(x^{(n)}_{[i]t}|x^{(n)}_{[i]t-1})$
Step 1: Sampling from the proposal distribution
(c)
$p(x_{[i]t}|z_{[1:i]t},z_{1:t-1})$
Step 2: Weighting particles according to likelihood
(d)
Step 3: Resampling

Fig. 3. Update for model i at time t. (a) The pdf and particles at time $t-1$. (b) The proposal is formed by fusing information from the current model's previous state and from the rest of the models. $Pa_t(\mathbf{x}_{[i]t})$ denotes the parent nodes of $\mathbf{x}_{[i]t}$ from time t, $Pa_{[i+1:M]t-1}(\mathbf{x}_{[i]t})$ denotes the parent nodes of $\mathbf{x}_{[i]t}$ from time $t-1$ excluding $\mathbf{x}_{[i]t-1}$. (c) The new particles are weighted. Darker particles have higher weight. (d) Resampling.

2.2 Tracker Implementation

In this section we use the framework to build a tracker, which we applied in tracking various targets in challenging situations. We combined three different object models to represent the target which are in the order which are updated:

(i) A salient point tracking model. This model has only 2 position parameters.

(ii) A blob tracking model. The blob represents a rectangular region of the target with homogeneous color with 3 parameters which describe its position and scale.

(iii) The target's contour. This is the main object model. It is represented as a b-spline curve and contains 5 parameters which allow several geometric transformations.

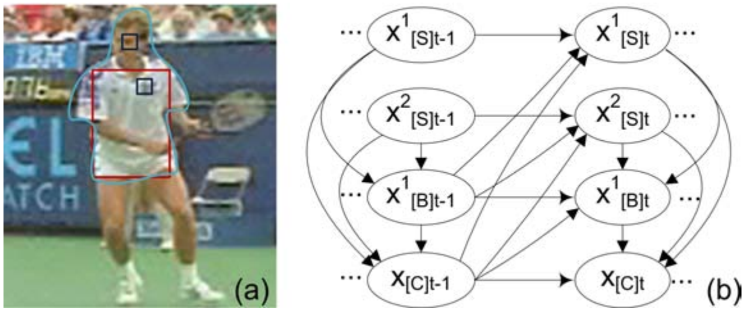

Fig. 4. (a)Sample image showing two salient points(dark), one blob(red) and one curve model(blue). (b)Graphical model of the implemented tracker, depicting slices $t-1$ and t of the temporal dimension for the aforementioned models. The evidence nodes are omitted for clarity.

The combined state vector is: $\mathbf{x} = [\mathbf{x}_{[S]}; \mathbf{x}_{[B]}; \mathbf{x}_{[C]}]$, where $\mathbf{x}_{[S]}$ represents the salient points $\mathbf{x}_{[B]}$ represents the blobs and $\mathbf{x}_{[C]}$ represents the contour curve. For a single target more than one salient points or blobs models can be used.

Adaptation Procedure So far we proposed a framework to fuse information from different cues using several object models which are initialized at the first frame. Here, we will expand the proposed framework to adapt the auxiliary models during tracking using information from the main model. The adaptation of the auxiliary models is integrated in the update equation. Each auxiliary model has a parameter that encodes the confidence of the object to belong to the target (target confidence). The adaptation consists of deleting an object if the target confidence is below a threshold and detecting and initializing new objects. For the k_s salient point model the target confidence parameter $\mathbf{x}^{k_s}_{[s_t]t}$ is initialized when the point is detected and updated during tracking by the following filtering equation:

$$\mathbf{x}^{k_s}_{[s_t]t} = a_{s_{tc}} \mathbf{x}^{k_s}_{[s_t]t-1} + (1 - a_{s_{tc}}) f_{stc}(\mathbf{x}^{k_s}_{[S]t-1}, \mathbf{x}_{[C]t-1}) \qquad (6)$$

Where $a_{s_{tc}}$ is the filtering parameter and t_{stc} is the threshold for deleting an auxiliary model. $f_{stc}(\cdot)$ is a metric measuring the compatibility between the points and the contour model on the previous frame:

$$f_{stc}(\mathbf{x}^{k_s}_{[S]t-1}, \mathbf{x}_{[C]t-1}) = \exp\left\{-\frac{d^2_{bht}(L^{k_s}_{[S]}, L_{[C]})}{2\sigma^2_{stc}}\right\} \tag{7}$$

where σ_{stc} is the deviation, $L^{k_s}_{[S]}$, $L_{[C]}$ are the likelihood vectors for the k_s-th salient point and the curve model respectively defined as
$L^{k_s}_{[S]} = [p(\mathbf{z}^{k_s}_{[S]}|\mathbf{x}^{k_s(1)}_{[S]}); ...; p(\mathbf{z}^{k_s}_{[S]}|\mathbf{x}^{k_s(N)}_{[S]})]$ and similarly for the curve model. This equation models the similarity between the likelihood vectors which is high when the two models describe the same target. In that case the link from one model to the other is meaningful. In contrast when one of the models is distracted by clutter then the similarity between the two vectors is expected to be lower. The same equations hold for the initialization and update of the target confidence parameter of the blob model $\mathbf{x}^{k_b}_{[b_t]t}$.

When $x^{k_s(n)}_{[s_t]t-1} < t_{stc}$ or $x^{k_b(n)}_{[b_t]t-1} < t_{stc}$ for the k_s-th salient point and k_b-th blob models respectively then the auxiliary model is deleted and a detection procedure searches for new salient points or blobs to re-initialize in the target region as defined by the main model.

3 Tracking Experiments

The experiments have been executed using several challenging video sequences and various objects have been tracked to verify our methods. More specifically, we experimented with sequences containing deformable objects, abrupt motion, heavy clutter, partial occlusions, and short full occlusions. For the experiments we implemented the following trackers which we compare: **SIR** - the original SIR algorithm, **MFP** and **aMFP** - the MFP tracker with and without adaptation, **HMT aHMT** - the HMF tracker with and without adaptation.

To compare the trackers we annotated several sequences by hand and we calculated the 'Tracker Detection Rate' (TDR) and 'False Alarm Rate' (FAR) measures [6]:

$$TDR = \frac{TP}{TP + FN}, FAR = \frac{FP}{FP + TP} \tag{8}$$

where TP, FN and FP denote the true positive, false negative and false positive area respectively.

In the experiment displayed in Figure 5 we compare our HMT tracker to the SIR in a PETS 2006 surveillance sequence using the blob and contour models. The contour hypotheses of our method are much more concentrated near the actual target than those of the SIR because of the strong prior provided by the blob model.

The experiment displayed in Figure 6, illustrates the concept of the model adaptation using the aHMT tracker with two model types, salient points and contour. The salient point models are deleted and re-initialized as the initial

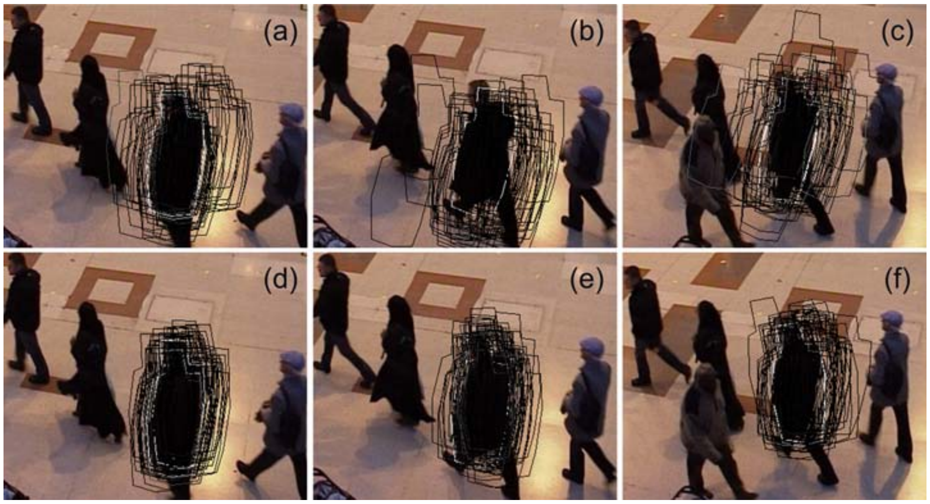

Fig. 5. Tracking Results - Surveillance Sequence:(a),(b),(c)-SIR, 50 particles (d),(e),(f)-HMT, 50 particles. Frames 1,35,50. Both trackers use the blob and the contour models, for image clarity we only show the contour particles. The particles of the HMT tracker are more concentrated near the target due to the better proposal distribution.

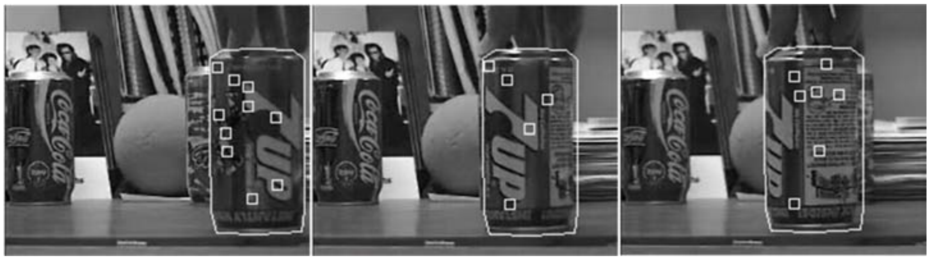

Fig. 6. Tracking Results - 7up Sequence:aHMT, 60 particles, frames 1,180,300

points are occluded due to the object rotation. The main model (contour), defines the target area and the search for new points is performed there.

In several cases some type of auxiliary models do not provide valid information for the target. In such cases the adaptation mechanism discards these models without replacing them. One such situation is observed in Figure 7 where the blob models are misled by similar background colors and are quickly discarded. The spots are not helpful throughout the whole sequence and therefore are discarded for several frames as well.

In the HMT framework the object models are connected and form a single graphical tracking model whereas in the MFP framework several independent trackers are used with each one having a single object model and they exchange information only through the proposal distribution. This type of connection

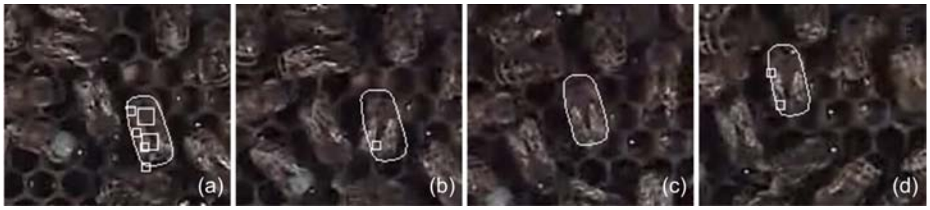

Fig. 7. Tracking Results - Sequence without blobs:aHMT, 50 particles, frames 1,15,30,50. The blob models, although initialized in (a), are quickly discarded because the background contains similar colors that distract it. In (c), the spot models are also discarded and only the contour is used. In (d), several new spots are detected.

between the models does not guarantee that the trackers will remain locked on the same target, especially when the adaptation method is not used. In Figure 8, this point is highlighted. The contour model is distracted by the player of the other team and destabilizes the tracker.

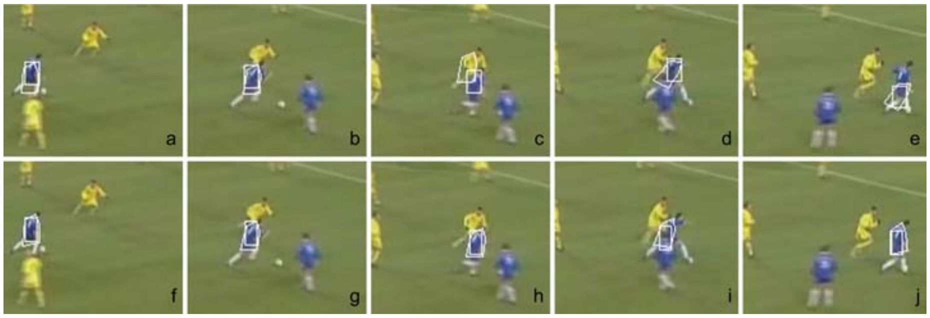

Fig. 8. Tracking Results - Soccer Sequence:(a)-(e) MFP, 50 particles, (f)-(j) HMF, 50 particles, frames 1,21,24,29,40. In (c), the contour model is distracted by the player of the other team as opposed to the HMF tracker which is not distracted as seen in (h).

4 Conclusions

Two feature fusion frameworks for visual tracking were presented. The implemented trackers were used in the following to create features for the developed violence detection system, which classifies movie segments as violent or non-violent. The trackers are based on the particle filtering methods. Their main goal was to create better hypotheses thus reducing the computational cost and enabling the use of high dimensional object models in real time applications.A violence detection system was also developed. The proposed system fuses audio and visual information using features that do not restrict the scope of its application and was tested on a real film dataset.

References

1. Anagnostopoulos, V., Kosmopoulos, D., Doulamis, A., Makris, A., Lalos, C., Varvarigou, T.: Automated production of personalized video content for visitors of thematic parks. In: IE06. pp. 173–181 (2006)
2. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. 24(4), 509–522 (2002)
3. Bowyer, K.W., Chang, K., Flynn, P.: A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. Comput. Vis. Image Underst. 101(1), 1–15 (2006)
4. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2). pp. 264–271 (2003)
5. Giannakopoulos, T.: Study and application of acoustic information for the detection of harmful content, and fusion with visual information. PhD Dissertation, NKUA (2009)
6. Hall, D., Nascimento, J., Ribeiro, P., Andrade, E., Moreno, P., Pesnel, S., List, T., Emonet, R., Fisher, R.B., Victor, J.S., Crowley, J.L.: Comparison of target detection algorithms using adaptive background models. In: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on. pp. 113–120 (15-16 Oct 2005)
7. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. Systems, Man and Cybernetics, Part C, IEEE Transactions on 34(3), 334–352 (2004), http://dx.doi.org/10.1109/TSMCC.2004.829274
8. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal of Computer Vision 29(1), 5–28 (1998), citeseer.ist.psu.edu/isard98condensation.html
9. Kosmopoulos, D.I., Doulamis, A., Makris, A., Doulamis, N., Chatzis, S., Middleton, S.E.: Vision-based production of personalized video. Image Commun. 24(3), 158–176 (2009)
10. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. In: ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II. pp. 3–19. Springer-Verlag, London, UK (2000)
11. Makris, A., Kosmopoulos, D.I., Perantonis, S.J., Theodoridis, S.: Hierarchical feature fusion for visual tracking. In: ICIP (6). pp. 289–292 (2007)
12. Pantic, M., Pentland, A., Nijholt, A., Huang, T.: Human computing and machine understanding of human behavior: a survey. In: ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces. pp. 239–248. ACM, New York, NY, USA (2006)
13. Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proceedings of the IEEE 92(3), 495–513 (2004)
14. Tan, X., Chen, S., Zhou, Z.H., Zhang, F.: Face recognition from a single image per person: A survey. Pattern Recogn. 39(9), 1725–1745 (2006)
15. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Comput. Surv. 35(4), 399–458 (2003)