Vision-Based Production of Personalized Video

D. I. Kosmopoulos[a] A. Doulamis[b] A. Makris[a] N. Doulamis[c]
S. Chatzis[c] S. E. Middleton[d]

[a] *NCSR Demokritos, Inst. of Informatics and Telecommunications, GR- 15310, Greece*

[b] *Technical University of Crete, Chania, Crete, Greece*

[c] *Department of Electrical and Computer Engineering, National Technical University of Athens, GR-15773, Greece*

[d] *IT Innovation Centre, University of Southampton SO16 7NP, UK*

Abstract

In this paper we present a novel vision-based system for the automated production of personalised video souvenirs for visitors in leisure and cultural heritage venues. Visitors are visually identified and tracked through a camera network. The system produces a personalized DVD souvenir at the end of a visitor's stay allowing visitors to relive their experiences. We analyze how we identify visitors by fusing facial and body features, how we track visitors, how the tracker recovers from failures due to occlusions, as well as how we annotate and compile the final product. Our experiments demonstrate the feasibility of the proposed approach.

Key words: human identification, tracking, automated content production

1 Introduction

A major part of the success of a museum, a gallery or a theme park is the total experience it offers and how it engages the visitor. One of the visitors' main

Email addresses: dkosmo@iit.demokritos.gr (D. I. Kosmopoulos), adoulam@cs.ntua.gr (A. Doulamis), amakris@iit.demokritos.gr (A. Makris), ndoulam@cs.ntua.gr (N. Doulamis), stchat@telecom.ntua.gr (S. Chatzis), sem@it-innovation.soton.ac.uk (S. E. Middleton).

concerns is to capture the visit, either by photographing or by videotaping the venue and their experiences in it.

However the results are usually not satisfactory, due to the limited visitors' filming experience, the generally low quality of the equipment that most visitors use for capturing, the inappropriate configuration of the media handling devices to the specific venue conditions and the limited efforts that often the visitors make. If the shooting is done by the visitor that person is left out of the content. Shooting from locations that are inaccessible for the visitors is not possible, e.g., at a roller coaster taken from above. Moreover, in certain venues such as museums video cameras are not permitted. As an alternative some parks sell DVDs with a "typical" day in the park, however these are not interesting since the people appearing in them are unrelated to the customer. PhotoPass [1] is a service offered in some entertainment parks, however it requires the presence of professional photographers, and the production is done manually, with obvious consequences to the cost. Recent scientific and technical advancements allow the automated production of personalized video content from the visit, which can be acquired by several cameras that are located at key points of the park or venue and are triggered by the presence of the persons who have asked for the service.

Nowadays, the few systems that are available for this purpose use Radio Frequency Identification tools, called RFID's for triggering media acquisition due to the relative simplicity of the involved technology, e.g., [2]. In those systems, Radio Frequency (RF) receivers are placed in appropriate locations. The user that has asked for the service has to carry a transmitter and as she passes near the RF receivers her location is registered and the cameras that monitor the specific area are activated. However, the use of RFIDs is problematic in several cases because the receiver is only activated when the emitter is within the receiver range (practically a few meters). The exact positioning can be defined only by using complex installations with multiple receivers and is not always possible, especially when the target moves within larger areas following unpredictable trajectories. Furthermore, the use of RFIDs is very unreliable and very often not possible at all in rides with large metal parts, e.g., the bumper car ride, due to signal reflections.

Recent advances in computer vision and surveillance systems have made feasible installations that will be able to detect, recognize, localize and continuously track the visitors of thematic parks using visual input, which does not suffer from the previous drawbacks. A detailed survey of methods and systems used in visual surveillance can be found in [3]. The proposed system capitalizes on these recent advancements and goes one step further by automatically acquiring, organizing, streaming and creating personalized videos. According to our knowledge it is the first such system for the purposes of the leisure industry.

According to the use scenario the visitor has to be registered in a dedicated booth, where a camera captures face and body images. When the visitor enters the venue she is viewed by identification cameras (i-cams). A unique ID is assigned to the visitor and matching is performed between the i-cam and one of the cameras that are used for tracking (t-cam). As long as she is in the Field of View (FOV) of the deployed cameras she is tracked across several t-cams maintaining the same ID. The cameras within the theme park will work simultaneously for multiple visitors. Moreover, remotely located people (e-visitors) may enter the Internet, through their terminal devices (PCs, mobile phones, PDAs), to watch their tour in real time. This way, the visitors can share their experiences with remote friends or the family. The final digital product is automatically produced when the visitor decides to leave. It may comprise edited video, still images, promotional material or raw content based on user preferences.

To achieve the aforementioned goals we have created three sub-systems: real-time, offline and on-demand processing (see figure 1). The real-time sub-system uses video input from cameras positioned around the venue to identify visitors using face and body models and to track them at almost frame rate; then it records raw footage and creates MPEG-7 metadata describing the visitor-specific footage. Part of the same sub-system is the adaptive streaming, which provides video streams for viewing by visitors' nominated e-visitors. The offline sub-system performs background processing to collect and enrich the visitor-specific metadata by semantic information to support efficient content-based footage retrieval when the final edit is created. The on-demand sub-system initially allows visitors to be registered with the system; as soon as the visit is completed it uses the raw footage, the visitor-specific metadata and the film template to create the final video edit and souvenir DVD; the same subsystem provides interfaces to visitors, e-visitors and administrators.

The main innovative features of the proposed system, which will be further described in the next sections, are:

- a person identification framework combining facial and body matching using a linearly optimized or a neural network based fusion scheme;
- a hierarchical particle filtering framework for real-time tracking, enhanced with recovery mechanism for occlusion handling;
- an automated production system using feature-augmented grammar.

In the next section we survey the recent developments in human recognition, tracking and automated production, on which our system capitalizes. We provide details on the real time subsystem in section 3 emphasizing on the innovative features of the system. Similarly in the section 4 we describe the on demand subsystem. In section 5 we provide the system setup and the experimental results that verify the system capabilities in subsystem and in overall system level. In section 6 we discuss the results of our research and finally in section 7 we conclude this paper.

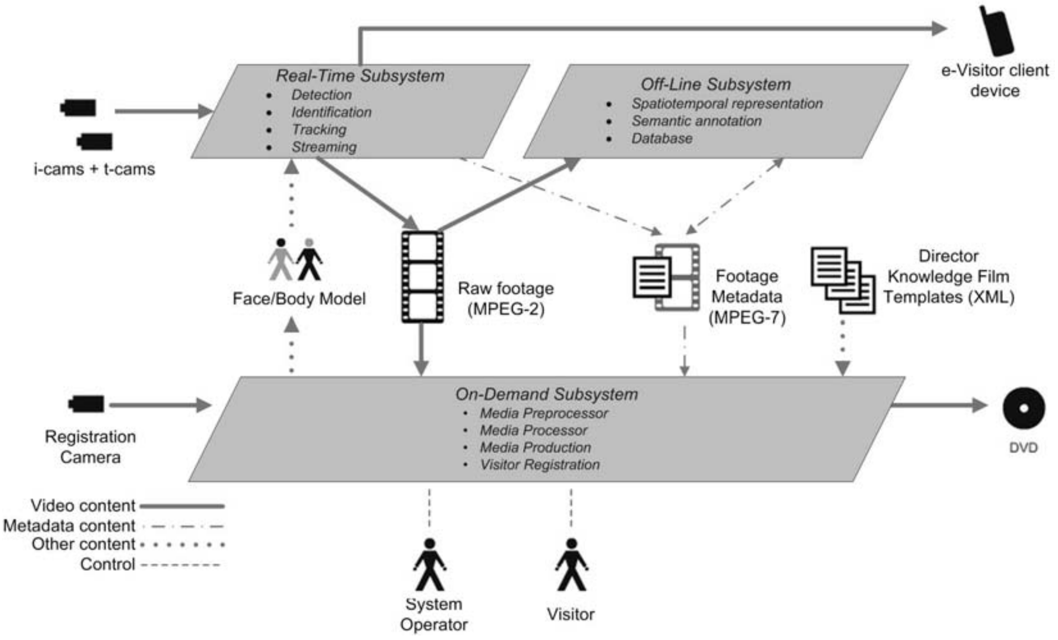

Fig. 1. The system architecture

2 Related work

As mentioned above, the architecture integrates several different research approaches in order to provide to the visitors of a venue professional mementos regarding their activities. We have stated in Section 1 that the current systems that are available for this purpose exploit Radio Frequency (RF) mechanisms for human identification and tracking. We extend the current state-of-the-art by using visual input. Thus, before presenting related research approaches, we recall that the system includes modules for a) human identification by exploiting the visual information, b) human tracking, c) automated video production. In the following subsections, we state the current techniques that provide the research background for the most important modules.

2.1 Human recognition

The main idea behind the proposed work is to build a system based on passive and non-intrusive sensors able to identify and track humans. The face recognition technology lends itself as a feasible solution for identifying humans in the environment of a leisure park. Over the last years, Automated Face Recognition technology has shown dramatic improvements. In [4] it is shown that there are now algorithms that surpass the human operators in both easy and difficult illumination conditions using the test sets and algorithms of the Face Recognition Grand Challenge 2006 (FRGC-06). In [5] the MID database was used. 29.2 percent of human subjects performed better than the best algorithm, while 37.5 per cent performed worse.

In [6] the results of the FRGC challenge are provided. Specifically, for recognition using frontal facial images under controlled conditions the recognition rates for the best algorithms were almost 100 percent. The results for all algorithms were better when four target and query images were used. The best algorithms using as target a single controlled image and as query a single uncontrolled still image provided success rates of about 80 percent. This experiment is closer to the general case of image acquisition in uncontrolled environment. As general comment the more controlled is the environment and the more images we use for target modeling and querying, the higher are the recognition rates.

Despite the progress made recently in face recognition techniques, the recognition process seems to benefit from employing additional non facial features, e.g., from body. In [7] different visual features have been compared for body recognition and the Color Structure Descriptor (CSD) achieved the highest recognition rate. In [8] people are recognized based on color and shape features and an SVM classifier is used. The use of Gaussian mixtures for modeling color is another attractive alternative. In [9] different distances between two Gaussian mixtures are compared (Bhattacharyya, Symmetric Kullback-Leibler, C2).

Specifically for outdoor environments the illumination variations may affect the results provided by the color model and therefore algorithms have been proposed for color constancy, i.e., for recognizing colors of objects independently of the color of the light source. Some of the most popular ones are the *Grey-world* assumption, which assumes that the spatial average of surface reflectances in a scene is achromatic and the *Gamut-mapping*, which recovers the transform that best projects the measured gamut into that of a canonical (see e.g., [10]). In [11] it is shown that both algorithms improve the color constancy for the visual surveillance scenario. The Grey-world method can be implemented in real time while the Gamut-mapping not. These methods must be used with caution since the changes of the image content may affect their performance. The problem of seeing the same color in two cameras is handled by using color calibration techniques (e.g., [12]).

As we are going to present in section 3 we have combined a face recognizer (using multiple images for modeling and querying) with a body recognizer.

We have tried to control the illumination and we have used color constancy and color calibration techniques.

2.2 Target tracking

To effectively address the problem of continuous, uninterrupted human tracking throughout a complex and extensive surveyed terrain (as dictated by our system requirements), a network of multiple cameras with overlapping FOV must be spanned across the area of interest. Such a system must incorporate effective algorithms (a) for tracking in the FOV of a single camera and (b) for tracking across cameras with overlapping FOV.

The taxonomy of the single-camera tracking methods includes point tracking, kernel tracking and silhouette tracking as categorized in [13]. Point tracking methods are adequate only for targets that are small and can be represented by points, but can be used for assisting purposes using meshes of points. Silhouette tracking methods provide the highest flexibility in the tracked shape but they do not handle explicitly the occlusions. The kernel - based methods assume the existence of a model for the tracked object (e.g., contour) and perform transformations on it to find a good match. Given that the kernel-based methods handle explicitly the occlusion problem we have focused on them.

Recently the SMC methods (Sequential Monte Carlo) (e.g.,[14]), also known as particle filters have been applied in kernel-based tracking. They can cope with multimodal distributions such as those emerging from a cluttered environment, they are relatively simple and they provide a framework to fuse different cues. 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 (internal target representation, e.g., contour, bounding box, human body model etc), a dynamic model (used to predict the next state given the current one) and an observation model, which links the object model to the data by calculating the likelihood of the object given the state.

The simple object models are fast but incomplete and thus difficult to track, while the more complex models provide better target representation but are difficult to initialize and are computationally expensive. 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 [15], the color and texture [16], [17], [18], and motion information [19], [20]. 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 and color based methods perform poorly in the existence of many similar colored objects. Several approaches for feature fusion have recently appeared [21], [22], [23], [24]. However, most of these works have high complexity and may still have problems with background clutter.

We use a novel Bayesian tracking method that overcomes the difficulties posed by the complex environment, by using several object models, which are updated hierarchically within the particle filtering framework. This approach enhances significantly the particle filters that use the same features in a non-hierarchical fashion. The algorithm is robust in various scene conditions. Each model uses several visual cues to define its likelihood function.

Regarding the target matching in overlapping FOV, there are several taxonomies of the related methods according to the used features and according to the requirement for camera calibration. A popular approach is to consider the targets as regions and then to use the features of the regions for matching in multiple views. Color is a popular feature and is modeled through color histograms, e.g., [25] or Gaussian color models, e.g., [26]. However, targets having similar colors may be poorly matched. Different viewpoints and lighting variations may cause the same target to be observed with different colors in different cameras. Inhomogeneous color may also cause problems if the same target exposes different colors in different cameras.

There are several approaches that use geometrical constraints, which may require either camera calibration or a homography constraint based on the ground plane. The 3D methods transform all points, e.g., target centroids into the common 3D coordinate system and perform matching based on the proximity of those points, e.g., [27]. Another approach is to use the epipolar constraint using only the relative pose of the cameras, e.g., [28].

Methods such as [29] assume that the target moves on a predominant ground plane. In that work, the homographies between each view and the ground plane are calculated: subsequently foreground likelihoods in all views are computed. It requires that the foreground blobs comprise the points where the subject touches the ground, however this is not always the case due to image noise and occlusions. Using the principal axis of a subject instead of a foreground likelihood map increases the robustness against noise. Since foreground pixels corresponding to a person are in general symmetrically distributed along the principal axis, the errors of monocular motion subtraction are also symmetrically distributed along the axis [30].

We have used a geometric approach due to the generality of the approach as opposed to methods making the ground plane assumption.

2.3 Automated Video production

Automated media production systems generally either perform pre-processing to create a new media item or run post-processing techniques to repurpose an existing media item.

In [31] automated video creation is used to select and combine camera shots into films according to predefined rules. The Auteur system [32], is a rule-based automatic editing system that takes a set of hand-annotated video shots and combines together sequences (or scenes) that are likely to suggest humorous content. Rules such as temporal association of an object (e.g., shot of banana skin) and result (e.g., someone slipping on the skin) are used to infer meaning (e.g., misfortune). The proposed system uses production rules but simplifies their creation by using a domain specific template, allowing rules to be more easily understood by non-technical film directors. This approach is more practically exploitable in a real system.

Systems such as [33],[34],[35] automatically classify video sequences according to automatically detected basic characteristics (e.g. pan, zoom, indoor, faces). These classifications provide the semantic labeling needed for efficient manual or automatic editing. The semantic annotation system in our system provides vision-based annotations that are made available to the rule-engine as MPEG-7 events.

Our approach to media production is interesting in that it processes semantically annotated footage (automatically extracted from live camera footage) and executes professional film production rules encoded as a film template. Our film template approach constrains the rule language into a simpler form and makes it practical to allow non-technical film experts to become involved in reviewing and developing the production rules.

3 Real time subsystem

3.1 Identification

Considering the state of the art and the related problems we have used a combination of face and body color modeling to recognise humans. The modeling is performed before entering the venue by capturing several face and body images from different angles.

The identification process is performed in four steps: (a) segmentation of the foreground from the background regions (b) face detection within the foreground regions (c) extraction of features from face and body and separate matching (d) fusion of facial and body features. By integrating the foreground extraction step we have two advantages: firstly we limit the application of the costly face detector to the foreground regions, which are small fractions of the total image; secondly we separate the body regions from the background and we use them for recognition.

The foreground extraction has been implemented using the algorithm described in [36]. In that work a pixel-level background model is maintained using a mixture of Gaussian distributions. The background parameters are updated automatically and the number of the used components per pixel is also automatically decided. Due to the use of multiple components the algorithm is applicable in outdoor scenes, which include periodic background motion.

In the extracted foreground region we seek for faces using the algorithm described in [37]. Generic Haar features are calculated and fed into a sequence of classifiers, which is used with the purpose of eliminating the largest number of negative inputs with little processing at the early stages (only positive results are further examined). The classifiers in the later stages are more accurate but combine more complex features. The training of each classifier proceeds according to the Adaboost algorithm which also selects the most appropriate features.

We adopt a fusion strategy for identifying the visitors. In particular, two different distance measures are used for the identification process. The first, denoted in the following as $D_1(u,v)$ expresses the distance of color distributions between the examined human object u and the v-th stored human object in the registration database. The second one, denoted as $D_2(u,v)$ expresses the distance of feature vectors that model human facial characteristics. These are described in the following.

3.1.1 Face and body distances

For the *body recognition* we have used Gaussian mixture models to represent body regions. Given that, we need a distance metric for matching the bodies of currently detected visitor u (represented by pdf $p(x)$) with the ones previously stored v (represented by pdf $p'(x)$). For this purpose we have used the following metric, as described in [9]:

$$D_1(u,v) = C(p,p') = -\log \frac{2\sum\limits_{i,j} \pi_i \pi'_j \sqrt{\frac{|V_{ij}|}{e^{k_{ij}}|\Sigma_i||\Sigma'_j|}}}{\sum\limits_{i,j} [\pi_i \pi_j \sqrt{\frac{|V_{ij}|}{e^{k_{ij}}|\Sigma_i||\Sigma_j|}}] + \sum\limits_{i,j} [\pi'_i \pi'_j \sqrt{\frac{|V_{ij}|}{e^{k_{ij}}|\Sigma'_i||\Sigma'_j|}}]} \quad (1)$$

$$V_{ij} = (\Sigma_i^{-1} + \Sigma_j^{'-1})^{-1} \tag{2}$$

$$k_{ij} = \mu_i^T \Sigma_i^{-1}(\mu_i - \mu_j^{'}) + \mu_j^{'T} \Sigma_j^{'-1}(\mu_j^{'} - \mu_i) \tag{3}$$

where π, $\pi^{'}$ the mixing weights, i and j are indexes on the Gaussian kernels, and, finally, μ, Σ and $\mu^{'}$, $\Sigma^{'}$ are mean and covariance matrices for the kernels of the Gaussian mixtures $p(x)$ and $p^{'}(x)$ respectively.

It is important to note that due to illumination variations it is necessary to build a look up table to establish a color correspondence between the colors of the cameras used in the registration and in the identification process. To minimize the effect of brightness variation in the same camera, color constancy techniques were used as will be mentioned in section 5.

For the *face recognition* functions we used a mainstream PCA - based approach similar to [38]. Each facial image is projected onto M dimensions by computing

$$F = [v_1, v_2, ..., v_M]^T \tag{4}$$

where the i-th coordinate of the facial image in the new space, which came to be the principal component v_i is given by $v_i = e_i^T w_i$ (e_i^T are the eigenvectors and $w_i = x_i$-m, x_i the current image, m the mean image). The distance $D_2(u, v)$ between current and training faces is calculated as the Euclidean distance of the face vectors as represented by (4). For survey of methods that could substitute our baseline method the reader is referred to works such as [6].

3.1.2 Fused distance

Two different approaches are adopted in this paper for fusing the distance metrics $D_1(q, j)$, and $D_2(q, j)$. The first approach linearly weighs each metric to obtain the overall metric used for ranking the humans' objects. The second uses a non-linear classifier, which takes as input the three distances and yields as output estimates of how close is the examined human object to the ones available in the registration database.

As far as the first approach is concerned, we optimally estimate the weights w_i, with $i = 1, 2$, which defines the importance of each metric to the overall object distance through a Mean Square Error (MSE) minimization. In this case the overall metric can be expressed as $D(u, v) = \sum_{i=1}^{2} w_i D_i(u, v)$. The weights w_i are calculated through training. If for an examined human u, the respective human in the registration database id v then, the ideal distance metric should be zero for these two objects u and v_u, (that is $ID(u, v_u) = 0$) while it should provide maximum values for all the rest ones. Thus, the learning strategy should estimate the optimal weights w_i so that the metric $D(u, v)$ is as close as possible to the ideal distance metric. That is,

$$\hat{w_i} : minE = min \sum_{\forall u} \sum_{\forall v} \epsilon^2(u, v) \tag{5}$$

where $\epsilon(u, v) = (D(u, v) - ID(u, v))$.

Minimization of equation (5) is accomplished by differentiating E with respect to the weights w_i and setting the derivative equal to zero, $\partial E/\partial w_i$=0,$\forall i$. Then, the optimal weights $\hat{w_i}$ are given by

$$\hat{\mathbf{w}} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{6}$$

where $a_{l,m} = \sum_{u,v} D_l(u, v) D_m(u, v)$ and $b_l = \sum_{u,v} D_l(u, v) ID(u, v)$.

The second approach relates the two distances using a non-linear relationship in order to fit the ideal distance. That is, the overall distance is now provided by a non-linear function which takes as inputs the two distances D_i, with $i = 1, 2$ and produces as output an approximation of the ideal overall metric. It is clear that relating with a non-linear function the three distances in order to produce the final overall distance metric would result in better performance rather than using a linear relationship. The main difficulty, however, in this case, is that the non-linear function that relates the inputs with the output is actually unknown. Non-linear function approximation can be achieved through a feed forward neural network classifier. The network uses a ground truth dataset of distances and is trained to approximate the ideal metric based on the results of the three distances. That is in this case, the approximate overall distance metric $ND(u, v)$ is given by

$$ND(u, v) = f_{nn}(D_1(u, v), D_2(u, v)) \tag{7}$$

where f_{nn} is the approximate of the non-linear function as provided by the neural network.

3.2 Detection and tracking

The first processing step for each captured image is the segmentation of moving image regions. Given the fact that we use static cameras we employ the background subtraction method described in [36] (a) to initialize our tracker and (b) to limit the regions in which we are searching for solutions to foreground areas. That method employs automatically resized Gaussian Mixture Model and thus it is appropriate for outdoor scenes, where the background may be moving periodically.

The initialization is performed by first classifying the person entering the scene as human using the method described in [39]. We project the foreground pixels on the horizontal axis, thus obtaining a histogram, which will have a peak close to the head region. We also check for curvature maxima around that region. After obtaining the head we fit a model for the upper part of the body and we start tracking it.

3.2.1 Bayesian Tracking/Particle Filters

In this section we provide the background for Bayesian tracking and the SMC methods which will be used to explain the proposed method. Let $\{\mathbf{x}_t; t \in N\}$ be an unobserved state of the target and $\{\mathbf{z}_t; t \in N\}$ the observations for every time step, t. The Bayesian tracking consists of calculating the posterior $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ at every step, given the measurements up to that step and a prior, $p(\mathbf{x}_0)$. The solution is expressed as:

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1})\frac{p(\mathbf{z}_t|\mathbf{x}_{0:t}, \mathbf{z}_{1:t-1})p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{8}$$

In most practical problems the state is considered a first order Markov process, i.e., $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{z}_{1:t-1})$=$p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the current measurements are considered independent of the previous measurements and states given the current state, i.e., $p(\mathbf{z}_t|\mathbf{x}_{0:t}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t|\mathbf{x}_t)$. Under these assumptions equation (8) becomes:

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1})\frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{9}$$

To recursively calculate the posterior, the terms involved in (9) have to be evaluated. The likelihood, $p(\mathbf{z}_t|\mathbf{x}_t)$, and the prior, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, are calculated using the selected measurement and dynamic model respectively. The evidence is given by: $p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t$

The particle filtering (PF) methods are used to approximate the above probabilities. They use samples (particles) to estimate the involved pdf's [40], [41]. Given N weighted particle trajectories $\{\mathbf{x}_{0:t-1}^{(n)}\}_{n=1}^{N}$ with importance weights $\{w_{t-1}^{(n)}\}_{n=1}^{N}$, up to time $t-1$, which approximate the distribution $p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1})$, the SMC methods compute N particles $\{\mathbf{x}_t^{(n)}\}_{n=1}^{N}$ which are combined with the previous trajectories to form $\{\mathbf{x}_{0:t}^{(n)}, w_t^{(n)}\}_{n=1}^N$, which approximates the posterior $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$, up to time t according to:

$$\hat{p}_N(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = \sum_{n=1}^{N} w_t^{(n)} \delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(n)}) \tag{10}$$

This approximation of $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ follows the importance sampling technique [14]. This technique relies on the use of another, so called proposal distribution $q(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$, from which it is easy to sample instead of sampling directly from $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$. Then the samples are weighted by:

$$w_t^{(n)} = \frac{p(\mathbf{x}_{0:t}^{(n)}|\mathbf{z}_{1:t})}{q(\mathbf{x}_{0:t}^{(n)}|\mathbf{z}_{1:t})} \tag{11}$$

The proposal distribution q is selected to factorize as (using the Markov assumption):

$$q(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) q(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1}) \tag{12}$$

to be able to update the weights recursively. The steps of the general Sampling Importance Resampling (SIR) [40] algorithm are:

- **Select** N samples $\{\mathbf{x}_t^{(n)}\}_{n=1}^N$ from the proposal $q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(n)}, \mathbf{z}_t)$.
- **Weight** each sample, n, using the following equation which results if we replace (9) and (12) in (11):

$$w_t^{'(n)} \propto w_{t-1}^{(n)} \frac{p(\mathbf{z}_t|\mathbf{x}_t^{(n)}) p(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)}, \mathbf{z}_t)} \tag{13}$$

- **Normalize** the weights so that their sum equals one:

$$w_t^{(n)} = \frac{w_t^{'(n)}}{\sum_{n_1=1}^{N} w_t^{'(n_1)}} \tag{14}$$

- **Resample** the resulting particle set $\{\mathbf{x}_{0:t}^{(n)}, w_t^{(n)}\}_{n=1}^N$ by multiplying or discarding particles according to their weight so that the new set will be unweighted and with the same number of particles.

A very common realization of this algorithm uses the prior $p(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)})$ as proposal distribution which if replaced in (13) results in the weights updated by the likelihood $p(\mathbf{z}_t|\mathbf{x}_t^{(n)})$ [15].

3.2.2 Proposed Tracking Algorithm

Our proposed algorithm uses a novel Bayesian tracking framework, an early version of which was briefly presented in [42]. The state (feature vector) $\mathbf{x}$ is composed by M parts $\mathbf{x}_{[i]}$, which correspond to probabilistically linked models of increasing complexity as follows:

$$\mathbf{x} = [\mathbf{x}_{[1]}^T, \mathbf{x}_{[2]}^T, \ldots, \mathbf{x}_{[M]}^T]^T \tag{15}$$

The simple models are updated first and then we are able to evaluate the conditional probability of the more complex models given the states of the simpler ones. For each model, one or more visual cues are used to define the likelihood. The last model (main model) is used to define the target area and is the one which needs to be estimated. The rest of the models are referred as auxiliary and their purpose is to provide better priors for the main model.

The steps of the algorithm for the t-th frame are:

For $i = 1$ to M do:

- Resample the particle set by selecting/discarding particles according to their weight so that the resulting set is unweighted and with the same number of particles.
- Update, the i-th model's particles by sampling from:

$$q(\mathbf{x}_{[i]t}|\mathbf{x}_{[1:i-1]t}, \mathbf{x}_{[i:M]t-1}, \mathbf{z}_{[i]t}) = p(\mathbf{x}_{[i]t}|\mathbf{x}_{[i]t-1})p(\mathbf{x}_{[i]t}|\mathbf{x}_{[i+1:M]t-1}, \mathbf{x}_{[1:i-1]t}) \tag{16}$$

- Weight the obtained samples by:

$$w_{[i]t}^{(n)} \propto w_{[i]t-1}^{(n)} \frac{p(\mathbf{z}_{[i]t}|\mathbf{x}_{[i]t}^{(n)})}{p(\mathbf{x}_{[i]t}^{(n)}|\mathbf{x}_{[i+1:M]t-1}, \mathbf{x}_{[1:i-1]t})} \tag{17}$$

In the above equations $p()$ indicates the probability density function and $\mathbf{x}_{[i]t}$, $\mathbf{z}_{[i]t}$ are the state and measurement for the i-th model at the t-th frame. $\mathbf{x}_{[1:i]t}$, $\mathbf{z}_{[1:i]t}$ are the states and measurements for all models 1 to i at the t-th frame. The $w_{[i]t}^{(n)}$, $\mathbf{x}_{[i]t}^{(n)}$ denote the weight and the state of the n-th particle for the i-th model and the t-th frame. The intuition behind eq. (16) is that the proposal is given by the product of probabilities of (a) the current model given its previous state and (b) the current model given the current states of the simpler ones and given the previous state of the more complex ones.

The auxiliary models use an adaptation procedure to re-initialize when they seem to lose track. For the adaptation procedure, we use a tracking confidence measure $f_{stc}(\cdot)$ which is calculated for each auxiliary model $i < M$ by measuring the compatibility to the main model. If this falls below a prede-

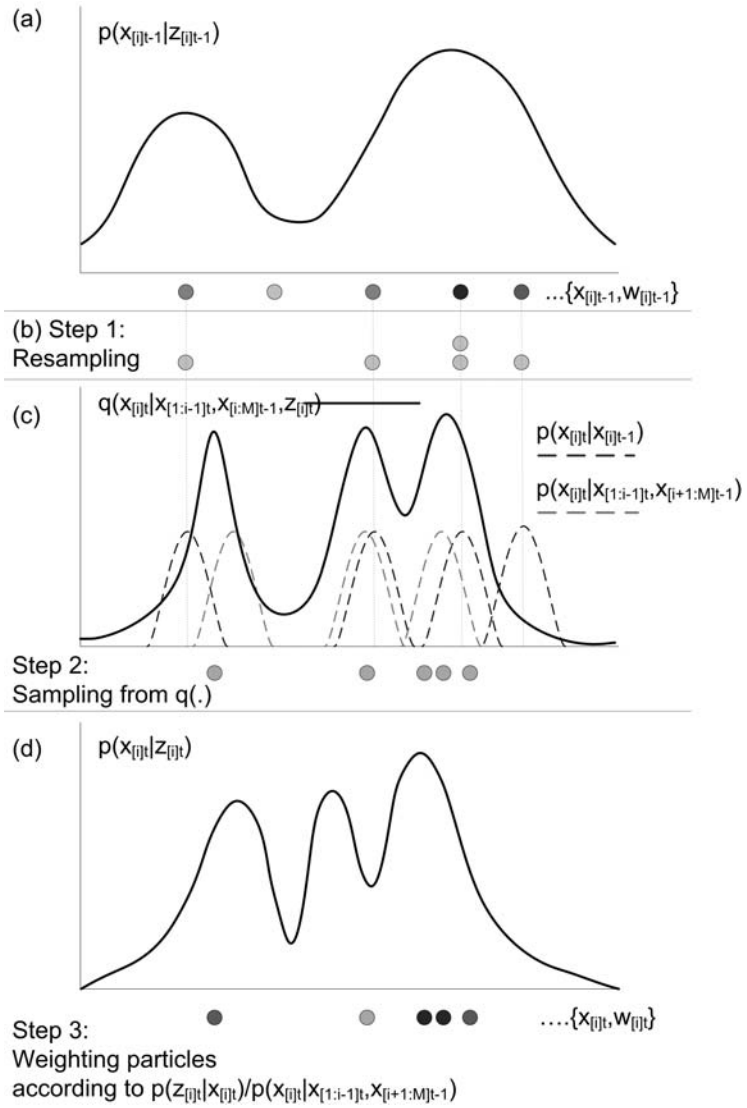

Fig. 2. Particle Update for model i at time t. (a) The pdf and particles at time $t-1$. The darker the color the higher the weight. (b) Resampling step, the particles are selected according to their weights. (c) The proposal distribution, from which the new particles are drawn, is formed by fusing information from the current model's previous particles and from the rest of the models (see eq. 16). (d) The new particles are weighted by the measurement model to approximate the posterior at time t (see eq. 17).

fined threshold the auxiliary model is deleted and a search for new features is performed to re-initialize it within the target area defined by the main model.

To satisfy the trade off between good tracking performance and efficiency we apply two models. The simple one is a set of salient points (corners) within the object; the more complex is the contour of the tracked object (main model). The combined state vector becomes: $\mathbf{x} = [\mathbf{x}_{[SP]}^T, \mathbf{x}_{[C]}^T]^T$ where $\mathbf{x}_{[SP]} = \mathbf{x}_{[1]}$ represents the centroid of all salient points and $\mathbf{x}_{[C]}$=$\mathbf{x}_{[2]}$ represents the contour spline curve through six parameters.

For the calculation of $\mathbf{x}_{[SP]}$ N_p points are used but each of them is updated independently so the dimension of the search space does not increase with the number of points. The model is determined by calculating the Sum of Squared Differences (SSD) in a rectangular mask around the candidate point and the original in the previous frame:

$$\mathbf{x}_{[SP]} = \sum_{j=1}^{N_p} \mathbf{x}_{SP_j} L_{SSD}(j) \tag{18}$$

where $\mathbf{x}_{SP_j}$ are the coordinates of the $j-th$ salient point.

$$L_{SSD}(j) = \frac{1}{\sigma_p\sqrt{2\pi}} \exp\left\{-\frac{d^2_{SSD}(j)}{2\sigma_p^2}\right\} \tag{19}$$

$$d_{SSD}(j) = \sqrt{\sum_{x,y}[T(x,y) - \overline{T}^{(j)}(x,y)]^2} \tag{20}$$

where $T(x,y)$ is a $n \times n$ mask around a candidate point, $\overline{T}^{(j)}$ is the $n \times n$ frame around the j-th original point and σ_p is experimentally defined.

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 (second term of product in eq. 16) and then sample from a Gaussian around that position (denominator in eq. 17) given by:

$$p(\mathbf{x}_{[C]t}/\mathbf{x}_{[SP]t}) = \frac{1}{\sigma_r\sqrt{2\pi}} \exp\left\{-\frac{[\mathbf{x}_{[SP]t} - D_{CP}]^2)}{2\sigma_r^2}\right\} \tag{21}$$

The adaptation metric for the salient points model is calculated by the following equation:

$$f_{stc}(\mathbf{x}_{SP_jt}, \mathbf{x}_{[C]t}) = \sum_{n=1}^{N} L^{(n)}_{SSD}(j) p(\mathbf{z}_{[C]}/\mathbf{x}^{(n)}_{[C]}) N(\mathbf{x}^{(n)}_{[C]t} + \mathbf{x}^{(n)}_{RSP_jt}, \sigma^2_{p_{sc}}) \tag{22}$$

where, $\mathbf{x}_{RSP_jt}$ is the initial relative position of the $j-th$ spot and the curve's center of mass, and $\sigma_{p_{sc}}$ is a variance parameter which is determined empirically. If this measure is below a predefined threshold the spot is deleted. The summation is over the particles. This metric takes low values for spots which have low likelihood particles corresponding to high likelihood curve particles and vice-versa.

For the calculation of $p(\mathbf{x}_{[C]t}/\mathbf{x}_{[C]t-1})$ (first term of product in eq. 16) we use a Gaussian around the previous contour position. Thus the $q(\mathbf{x}_{[C]t}/\mathbf{x}_{[SP]t-1}, \mathbf{x}_{[C]t-1}, \mathbf{z}_{SP}, \mathbf{z}_C)$ can now be calculated from equation 16.

The next step is the calculation of $p(\mathbf{z}_{[C]t}/\mathbf{x}^{(n)}_{[i]t})$ (nominator in eq. 17). For each sample we define:

$$p(\mathbf{z}_{[C]}/\mathbf{x}_{[C]}) = L_{CH} L_{CE} \tag{23}$$

The L_{CE} estimates how well the edges fit the current contour, while L_{CH} estimated how well the color of the surrounded region fits the template by calculating the corresponding Bhattacharyya distance:

$$L_{CH} = \frac{1}{\sigma_{c_{lh}}\sqrt{2\pi}} \exp\left\{ -\frac{d^2_{bht}(H(\overline{\mathbf{x}}_{[C]}), H(\mathbf{x}^{(n)}_{[C]}))}{2\sigma^2_{c_{lh}}} \right\} \tag{24}$$

where $H(\mathbf{x}^{(n)}_{[C]}))$ is the histogram for the n curve particle, $H(\overline{\mathbf{x}}_{[C]})$ is the curve's template histogram and $\sigma_{c_{lh}}$ is the variance. The $d_{bht}(\cdot)$ denotes the Bhattacharyya distance defined as:

$$d_B(H_1, H_2) = \sqrt{1 - \sum_{u=1}^{m} \sqrt{[H_1^{(u)} H_2^{(u)}]}} \tag{25}$$

where the summation is over the m histogram bins.

$$L_{CE} = \frac{1}{\sigma_{c_{le}}\sqrt{2\pi}} \exp\left\{ -\frac{f^2_{cl}(\mathbf{x}^{(n)}_{[C]})}{2\sigma^2_{c_{le}}} \right\} \tag{26}$$

Where $f_{cl}(\cdot)$ is a metric which determines how well the edges fit the current contour, and $\sigma_{c_{le}}$ is the variance.

3.3 Tracking Recovery

To overcome the difficulties posed by full or partial occlusions, we need an automatic recovery mechanism able to re-initialize the tracker each time that we consider its performance unacceptable. Tracking recovery is performed by *an object modeling* mechanism, which labels image regions with a probability of belonging to one of the available tracked objects in the scene. The color and texture properties are used as appropriate attributes of an image region since they can describe it adequately within small time windows. To address, however, color/texture fluctuations due, for example, to illumination variations, we introduce an *adaptable* object modeling algorithm which will be of higher computational complexity but also more accurate than tracking (due to its global nature) and will be executed once every k frames modeling the regions enclosed by the tracking curve. The model will be backprojected in the image each time the tracker posterior (9) falls below a predefined threshold (defined ad-hoc). This mechanism copes with the non-linearities that map color and texture properties of image regions to particular objects.

The color and texture properties of an image region are modeled using some of Discrete Cosine Transform (DCT) coefficients for each 8x8 image block. Let us denote as $\mathbf{a}_i(n)$ a feature vector which contains the DC and some AC coefficients of the DCT for the i-th image block of the n-th frame. Let us also denote as $\mathbf{y}(\mathbf{a}_i(n)) = [y_1(\mathbf{a}_i(n))\ y_2(\mathbf{a}_i(n)) \cdots y_K(\mathbf{a}_i(n))]^T$ a vector each element of which $y_m(\mathbf{a}_i(n))$ expresses the probability the i-th image block of the n-th frame to belong to the m-th tracked object.

Function $y_m(\cdot)$ is an unknown non-linear function, which maps the color texture properties of an image block to m-th object. However, using concepts from functional analysis, we can parameterize any non-linear function (with some assumptions on its continuity) as a series of known functional components,

$$y(\mathbf{a}_i(n)) \approx \sum_{j=1}^{N} c_j(n)\phi_j(\mathbf{a}_i(n)) \tag{27}$$

where in the previous equation we have omitted subscript m for simplicity. The $\phi_j(\cdot)$ refer to the known functional components, while $c_j(n)$ are the respective coefficients and N is the order of approximation for the n-th frame. For convenience, we can use functional components of the same type. One common choice regarding the type of the functional components $\phi(\cdot)$ is the sigmoid functions, defined as $1/(1+\exp(-x))$. This is due to the fact that sigmoid is bounded, monotonically increasing and continuous, which are the requirements that the functional components should satisfy. In this case, we need an additional parameter, say $\mathbf{q}_j(n)$ for scaling the $\phi_j(\cdot)$. This means that $\phi_j(\mathbf{a}_i(n)) = \phi(\mathbf{q}_j(n), \mathbf{a}_i(n))$. One common choice for scaling the sigmoid function is through the inner product between $\mathbf{q}_j(n)$ and $\mathbf{a}_i(n)$, that is equation

(27) is expressed as

$$y(\mathbf{a}_i(n)) \approx \sum_{j=1}^{N} c_j(n)\phi(\mathbf{q}_j(n)^T \cdot \mathbf{a}_i(n)) = \mathbf{c}^T(n) \cdot \mathbf{f}(\mathbf{n}) \qquad (28)$$

where $\mathbf{c}(n)$ is a vector that contains all the coefficients $c_j(n)$ and $f(n) = [\phi(\mathbf{q}_1(n)^T \cdot \mathbf{a}_i(n)) \cdots \phi(\mathbf{q}_N(n)^T \cdot \mathbf{a}_i(n))]^T$ is a vector valued function. If we form a matrix $\mathbf{Q(n)}$ that gathers all vectors $\mathbf{q}_j(n)$, that is $\mathbf{Q(n)} = [\mathbf{q}_1(n) \cdots \mathbf{q}_N(n)]^T$, then vector $\mathbf{f}(n)$ can be written as

$$\mathbf{f}(n) = \phi(\mathbf{Q(n)} \cdot \mathbf{a_i}(n)) \qquad (29)$$

In equation (28), the unknown parameters that should be estimated are the coefficients $c_j(n)$ and $\mathbf{q}_j(n)$. These parameters are computed by the use of a *recursive learning strategy* which is described in the following.

Let us assume that a reliable mask for the m-th tracked object has been extracted by tracking. Then, a set, O_m, is constructed which contains all image regions of the m-th object. Let us also denote as B the set of the background image regions. Then, the output of a classifier that recognizes O_m should be $y(\mathbf{a}_i(n)) = t_i$, where $t_i = 1, \forall i \in O_m$ and $t_i = 0, \forall i \in B$.

Since the color and texture properties either for the tracked objects or for the background slightly change from frame to frame, we can assume that the coefficients $\mathbf{c}(n)$ and $\mathbf{Q}(n)$ are derived from the previously estimated coefficients $\mathbf{c}(n-1)$ and $\mathbf{Q}(n-1)$ plus a small perturbation, that is $\mathbf{c}(n) = \mathbf{c}(n-1) + d\mathbf{c}$ and $\mathbf{Q}(n) = \mathbf{Q}(n-1) + d\mathbf{Q}$. Under such assumption, we can linearize equation (28) using a first order Taylor series expansion. In particular, equation (29) is written as

$$\phi(\mathbf{Q(n)} \cdot \mathbf{a_i}(n)) = \phi(\mathbf{Q(n-1)} \cdot \mathbf{a_i}(n)) + \mathbf{D} \cdot d\mathbf{Q} \cdot \mathbf{a_i}(n) \qquad (30)$$

where $\mathbf{D}$ is a diagonal matrix that contains the derivatives of the $\phi(\mathbf{q}_j(n-1)^T \cdot \mathbf{a}_i(n-1))$ with respect to the parameters $\mathbf{q}_j(n-1)$. Then, by combining $\mathbf{c}(n) = \mathbf{c}(n-1) + d\mathbf{c}$ and the previous equation and ignoring second order derivatives, we can conclude that

$$y(\mathbf{a}_i(n)) = y(\mathbf{a}_i(n)|\mathbf{Q}(n-1), \mathbf{c}(n-1)) + [d\mathbf{c} \;\; d\mathbf{q}_1 \cdots d\mathbf{q}_N]^T \cdot [\mathbf{u} \, \mathbf{r}_1 \cdots \mathbf{r}_N] (31)$$

where $y(\mathbf{a}_i(n)|\mathbf{Q}(n-1), \mathbf{c}(n-1))$ denotes the output of the classifier at the current image region but using the previous coefficients, that is

$$y(\mathbf{a}_i(n)|\mathbf{Q}(n-1), \mathbf{c}(n-1)) = \mathbf{c}^T(n-1) \cdot \phi(\mathbf{Q(n-1)} \cdot \mathbf{a_i}(n)) \qquad (32)$$

while vectors $\mathbf{u}$, $\mathbf{r_1}$,...,$\mathbf{r_N}$ are related with the previous coefficients as follows

$$\mathbf{u} = \phi(\mathbf{Q(n-1)} \cdot \mathbf{a_i}(n)) \tag{33}$$

and

$$\mathbf{r}_l = c_l(n-1)d_l\mathbf{a_i}(n) \tag{34}$$

where d_l is the l-th element of the diagonal matrix $\mathbf{D}$.

We recall that at the frame on which learning strategy is activated the output of the classifier is $y(\mathbf{a}_i(n))$ is known and equals t_i. In this case, we denote as $e_i(n)$ the difference between the actual target output and the one provided by the classifier using the coefficients before the adaptation, that is $e_i(n) = t_i - y(\mathbf{a}_i(n)|\mathbf{Q}(n-1), \mathbf{c}(n-1))$. The error $e_i(n)$ is related with a linear equation with the small perturbations $d\mathbf{c}$, $d\mathbf{q}_j$ (see equation (32)). In order to reliably estimate the coefficients for the adaptive non-linear model, we should take into account the effect of all image regions for the m-th object, and the background, i.e., for all elements of sets O_m and B. Usually, however, a precise estimation of the adaptive coefficients can be derived by also gathering information for the same object but at different frames. In this case, a set of linear equations are formed, and the perturbations of $d\mathbf{c}$ and $d\mathbf{q}_j$ are given by

$$[d\mathbf{c} \;\; d\mathbf{q_1} \cdots d\mathbf{q}_N] = (\mathbf{\Gamma}^T \cdot \mathbf{\Gamma})^{-1} \cdot \mathbf{\Gamma}^T \cdot \mathbf{e} \tag{35}$$

where vector $\mathbf{e}$ contains all the differences e_j for all image regions and frames and $\mathbf{\Gamma}$ a matrix which includes the respective values of $\mathbf{u}$, $\mathbf{r_1}$, $\mathbf{r_2}$,...,$\mathbf{r_N}$ for all error differences $e_j(n)$.

3.4 Multi-camera tracking

The multicamera tracking consists of two phases: the offline camera registration and the online tracking. The offline procedure is related to the calculation of camera geometry, while in the online phase the tracking is performed. The object tracking extracts the correspondences between the multiple views of a tracked object, using the obtained epipolar geometry information.

In the offline procedure we firstly estimate the camera geometry, which is expressed by the fundamental matrix (see, e.g., [43]). We have evaluated a number of different algorithms proposed for the estimation of the fundamental matrix. Firstly, we evaluated the regression-based approaches (ordinary least squares), in which the error is subsumed into one of the variables, and orthogonal regression, in which there is considered an error in all of the variables. We also tested robust algorithms, which relax the strong assumptions of regression-based methodologies: RANSAC and MAPSAC (see for example [43]).

The target matching is executed during the online phase (a) every time a visitor is identified so that tracking is initialized in the image of a t-cam given the position extracted from an i-cam and (b) when the visitor crosses the FOV of neighboring t-cams. In the source image the location of a visitor is defined by the image point, which indicates the top of the visitor's head (located in a manner similar to the one proposed in [39]). During visitor matching the tracking module that processes frames from the destination camera gets a message that contains the head point location and the visitor ID from the respective module that processes images from the source camera. Having available the fundamental matrix between source and destination camera we map the head point detected in source camera to a line in destination camera. Then the tracked head point from camera one, which has the minimal distance to this epipolar line, will be selected for the assignment.

4 On demand subsystem

In film making, there are some pretty basic rules [31] that can be applied to make an aesthetically pleasing film. The basic challenge in film making is in selecting and editing video clips according to these rules to produce something that looks 'right'. The system requires films to be made on-demand, in about 10 minutes, from video footage recorded during a visitor's day trip to a theme park or museum.

The media processor is made up of three distinct components. Firstly there is a media pre-processor step where MPEG-7 annotations of individual video clips are collated for a specific visitor. Secondly a media processor rule-engine executes a film template for this collection of MPEG-7 annotations and generated a SMIL edit decision list that follows the encoded directorial rules. Lastly a set of media production tools execute the SMIL film script, combines video clips in the right sequence and with the right duration, and burns a DVD souvenir.

4.1 Media pre-processor

Our media pre-processor extracts from the database the set of MPEG-7 descriptions for all camera footage where a specific visitor has appeared. The

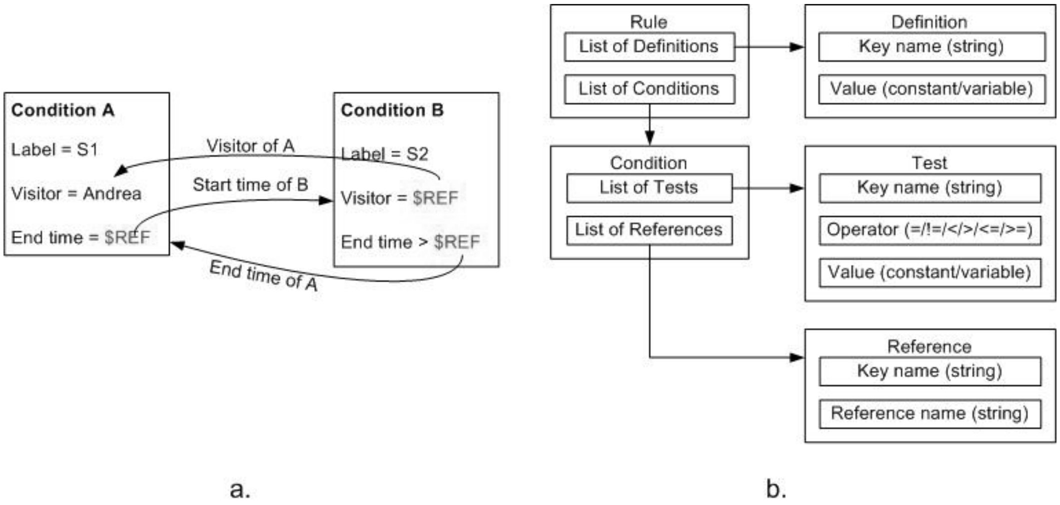

Fig. 3. a.Describing a combination of two elements with mutually dependent conditions b. Feature-augmented grammar for video editing

associated semantic annotations are then bundled together and sent to the media processor as a single MPEG-7 description. This is the shot list the media processor can select from to make the DVD. Semantic annotation includes camera ID, visitor ID, visitor bounding box, and simple semantics such as target position in the image (left, right, upper, or lower view border), entering/leaving camera FOV, position relative to known object.

4.2 Media processor rule-engine

The media processor service is a web service that takes a MPEG-7 document, which describes all the shots from a visitor's day at the theme park/venue and generates an 'edit decision list' for it based on some predefined directorial rules. These rules are encoded in a film template created by a professional director. The edit decision list is a film script which describes how the visitor's video clips should be used to make the film. We have used the SMIL edit decision list (see, e.g., [44]), which is a list of video clip elements where each element describes essential information about the video clip such as start time, duration (i.e. end time is equal to start time plus duration) and video source (i.e. which camera).

Our rule-based media processor's approach was inspired by previous work in computational linguistics [45] and expert systems [46]. The primary distinctions are that the proposed grammar is designed for video clip aggregation as opposed to word ordering as in text generation. Secondly, the grammar allows the use of mutually dependent conditions as opposed to conventional independent conditions. Two conditions can define tests that reference the properties of elements that are matched by the other condition (see figure 3a).

The proposed grammar operates on elements (see figure 3b). The grammar describes the creation of elements based on other elements. An element is a collection of properties where a property is a key-value pair. The key is a textual name and a value can be anything that is comparable, e.g., number, text and time. For instance, a video clip is an element with properties such as camera name, start/end time, duration, object type and object name.

Our rule engine is implemented using CLIPS [47] and runs via a GSOAP web service that the rest of the system can invoke. Our algorithm for rule application uses an element pool initialized with the observed events, e.g., a visitor was detected by Cam 1. Every rule is then applied to the pool to create a new set of elements which are added back to the pool, e.g., the shots, the short films and finally the complete film.

Set refinement is used to find all possible combinations of elements that satisfy the rule conditions in three steps. The first step applies context-free tests to eliminate clearly non-matching elements, e.g. to exclude all clips not about a specific visitor. The second step analyses the candidates for each condition to establish the range of property values in each group, thus enabling the application of context-dependant tests to further reduce the candidate set, e.g., to select the longest video clip. The final step applies the remaining tests to each candidate combination to identify complete matches which result in the creation of new elements that satisfy all the conditions.

4.3 Media production

The final module of the architecture produces automatically the final mementos. This is delivered in the form of a DVD that is given to the registered visitors of the venue upon their departure. The DVD production is performed automatically by taking into account the edit decision list as mentioned in the previous sub-section and interoperably described using the SMIL language. In this way, the most characteristic phases of the total recorded material are selected in a cinematographic fashion that permits the creation of a complete product of high quality not only with respect to the recordings but also to the synthesis of the new material. The DVD includes apart from video files selected images in order to produce e-souvenirs like e-photos, e-cards and e-albums.

5 Experimental evaluation : the bumper car ride

5.1 Scenario and setup

Several experiments have been performed in the premises of ALLOU entertainment park in Athens to prove the validity of our approach. More specifically the bumper car ride was used, which is undercover but this does not prevent illumination variations in certain parts of the ride. In such a scenario the RF sensors are not applicable due to the metal floor that causes reflections.

The scenario in this case has as follows: The visitor registers in the system before the ride, at some time later the visitor enters the ride, becomes recognized and then becomes tracked until entering one of the bumper cars. The visitor is associated with the selected bumper car and then the car starts to be tracked. To make the car tracking easier we have selected unique colors for each car. The system associates the frames where the associated car appears with the visitor. The system based on the predefined template selects the best shots, in which the visitor appears, to assemble the final video clip.

In our experimental setup we forced the visitors to pass from certain entry points, in which they could be seen by high resolution cameras, so that they could be recognized. Visitors followed a trajectory along which they could be seen by a set of cameras with overlapping FOVs, so that correspondences could be extracted. We also assumed that the visitors pass through at a walking pace, so that there was enough time to recognise the person and to initiate the tracking function. The illumination did not change significantly during the visitor presence, so that consistent tracking was possible. These above assumptions are realistic in a semi-controlled environment such as the bumper car ride.

Figure 4 displays one of the camera configurations we have experimented with. For an i-cam we have three t-cams used for the target tracking. The i-cam zoomed at the entrance, from which were sure that the visitors would pass. The resolution was relatively high, so that the required features for face and body matching could be extracted with higher reliability. On the contrary the t-cams covered wider areas for content acquisition purposes and high resolution was not a strict requirement, since the visitors could be tracked even at lower resolution. We also mounted 2 wireless cameras on each bumper car to take onboard shots.

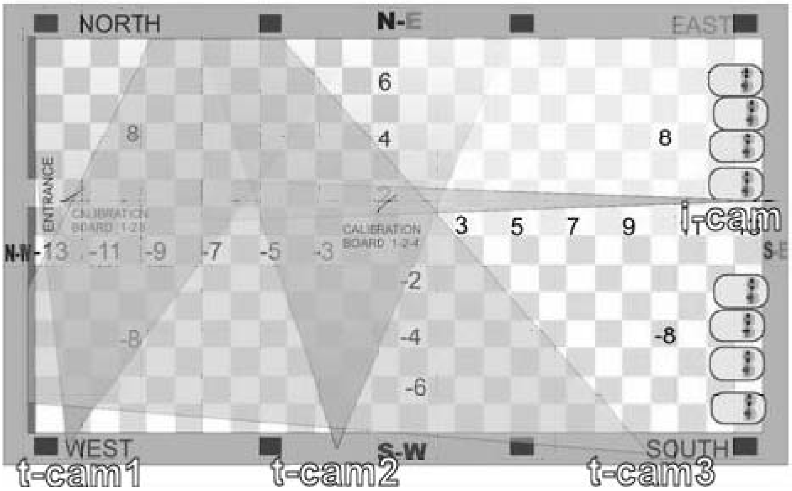

Fig. 4. The camera configuration

Distance Method	**False positive (%)**	**False negative (%)**
Facial Features	29.91	32.42
Body Color Features	22.22	25.64
Facial and Body Features (no weighting approach)	17.09	20.51
Facial and Body Features (weighted approach)	12.82	14.53
Non-Linear Fusion (Neural Network Model)	3.42	4.27

Table 1
Human identification using different features and fusion methods

5.2 Subsystems

5.2.1 Real-time

Table 1 presents the performance of the human identification process contacting during the registration phase. The results have been obtained using different feature fusion methods. In particular, initially, we use only facial features for the identification. Then, we measured the human similarity by exploiting only the color distribution of human body. Then, we investigated the effect of different feature fusion methods on human identification performance. Three different data fusion approaches were examined. The first was a simple concatenation of the facial / color body features. The second exploited the optimal weighted strategy described in section 3.1.2. In this case, facial and color body features were appropriately scaled to maximize human identification performance. The third approach non-linearly fused the features using a feed forward neural network architecture. As was observed, the best performance was encountered for the latter non-linear case. This was expected since the importance of facial and color body features on human identification do not follow a simple linear relation. The second best method was the optimal weighted linear strategy.

The identification results refer to all the frames acquired in a time period of about 3-4 seconds, which was the time required for a walking person to pass from the field of view of the i-cam. We were able to process frames for the recognition task at a rate of approximately 5 per second. Clearly there was significant improvement in the recognition rates if the fused information from face recognition and body region was used. In the experiments 14 people were registered and 23 were unregistered. The error factors in face recognition had to do with face detection (gave 1.5% false positive and 3.3% false negative by checking also for skin color in the returned regions), facial expressions, which altered the facial characteristics, and the relatively low resolution of the acquired video. As for the body recognition the illumination changes and the shadows were adversely affecting the results.

To achieve color constancy we have used the Grey-World assumption due to its real time capabilities. More specifically we have used parts of the image, which we knew that would always be unoccluded and then we had adapted the image accordingly. The camera used for registration and the i-cam have been calibrated to color using the Scarse tool [48] and a MacBeth color checker. For body modeling and recognition a mixture of Gaussians was used with three components in the RGB color space. The distance metric given in 5.2 has given the highest discrimination and was therefore used.

The unit test of the tracking module has been done using approximately 3600 frames annotated by a human operator. The measures that quantify the effectiveness of the algorithm for typical sequences that we tested are the 'Tracker Detection Rate' (TDR) and the 'False Alarm Rate' (FAR) defined as in [49]:

$$(TDR) = \frac{TP}{TP + FN}, (FAR) = \frac{FP}{FP + TP} \tag{36}$$

where TP, FN and FP denote the number of true positive, false negative and false positive pixels using hand-annotated ground truth bounding boxes for simplicity. We compared our method with the classical particle filter using the same models and features combined in a single state vector to show that by using the same features (salient points and edges) our hierarchical approach provides better results in terms of tracking accuracy. The experiments have been performed with both trackers sharing the same number of particles to equalize the computational resources. The number of particles were selected to allow for real-time tracking (≈25fps) while providing acceptable performance. Table 2 indicates the effectiveness of the tracking method for humans. The average values are positive considering the random localization of the bounding rectangle, which would give average TDR≤0.05 and average FAR≥0.95 for our t-cam sequences (ideally TDR=1, FAR=0). Although the bounding rectangle is not perfectly accurate, as proved by experiments given in section 5.3 the goal of extracting the target trajectory was achieved. Most of the errors are due to illumination variations, which could alter the perceived colors. We have used sequences captured from our system as well as standard PETS sequences (railway station) and the results (without recovery) are presented in table 2. An example from the PETS sequences as well as the aforementioned metrics are depicted in figures 5, 6 ,7.

The task of tracking bumper cars was much easier due to the fact that these are non-deformable objects, with different characteristic colors for each car, which are known in advance. After a person was entering a bumper car the car was associated with the visitor. Then we were searching for the characteristic color in the non static regions in each frame for each camera and we established spatial coherency in consecutive frames to minimize false positives. For approximately 80000 we had 0.05% false positives and 0.12% false negatives.

The tracking recovery tool has been tested in 30 occlusion cases (partial or total) where the tracker was losing the target and helped the tracker to recover in 25 of them, while the failures stemmed from targets having similar appearance. The modeling was done every 50 frames. The speed of the algorithm depends on the maximum number of iterations and the associated convergence threshold. Both were defined so that the algorithm would run at a rate of approximately 10 fps, given the size of the tracked objects, which gave reasonable performance and results not far from optimal. The output masks were also combined with masks from background subtraction for best results. A characteristic case is presented in figure (8), where the tracking with and without recovery is demonstrated. In figure 9 the respective mean likelihood for all particles is presented, which activates recovery if it falls below 0.4. The mean and not the maximum is used to minimize noise. The model parameters were dynamically updated each time a reliable mask was provided by the tracker for each object. The likelihood threshold depends on the specific view and the tracked objects. Instead, in case that tracking performance deteriorates, the object identification assists the tracker to recover objects' contours. To verify the good performance of the proposed adaptive identification process we compared the objects' masks for each object overall all frames with ground truth data, i.e., references masks of these objects. The average error of the region labeled by the classifier than the ground truth data was about 20%. This error includes both (a) misclassified image regions of that object to others and (b) misclassified regions of the other objects to the one considered.

For all experiments, we considered blocks of 8x8 pixels as image regions. Thus, object labeling was performed on an 8x8 image block. For each block, the DC and the 7 AC coefficients of the DCT transform were used. These 8 feature elements were extracted for each color component of the image in the Hue Saturation Value (HSV) color space. Thus, each image block was characterized by 24 features elements. These feature elements constituted the attribute vector $\mathbf{a}_i(n)$ which is involved in equation (27).

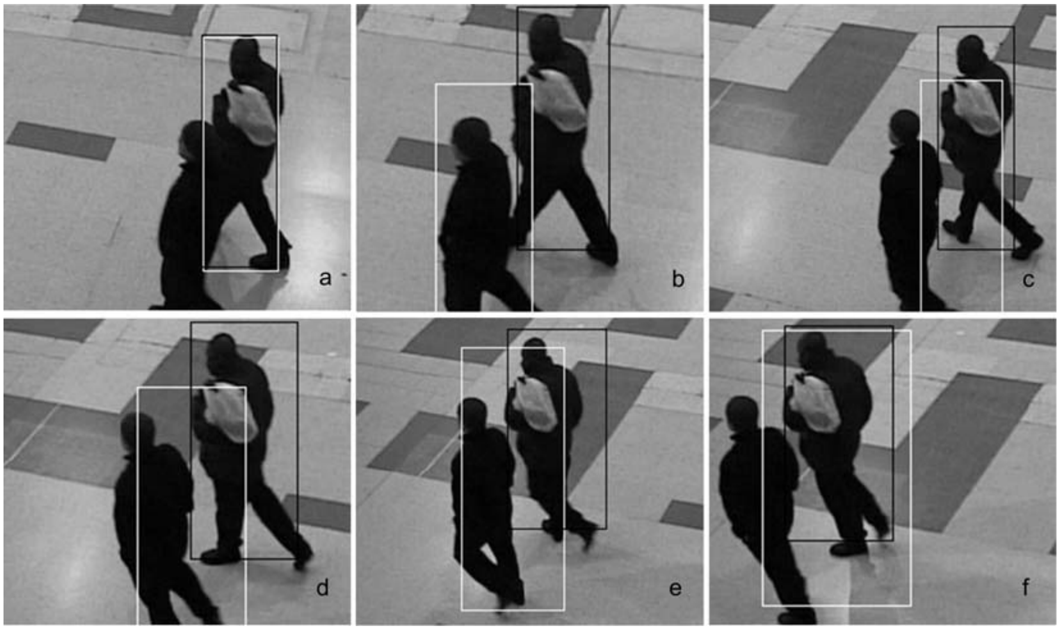

Fig. 5. Tracking Results - PETS Sequence: White PF, Black PM, 100 particles, frames 1,15,30,45,60,75.

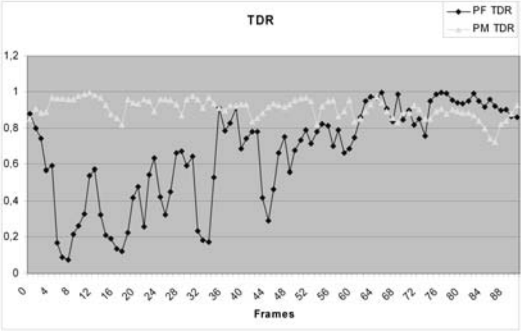

Fig. 6. Tracking Results Chart(TDR) - PETS Sequence.

The evaluation for the multicamera tracking was conducted by firstly estimating the epipolar geometry using a set of training points using a calibration pattern (visible from both cameras) and further evaluating it by using a set of corresponding test points. For each one of the source camera points of our test set we computed the corresponding epipolar line to the destination camera view, using the estimated epipolar geometry and we calculated the distance of its corresponding point from that line. The estimation of epipolar geometry using calibrated cameras was conducted estimating the essential matrix

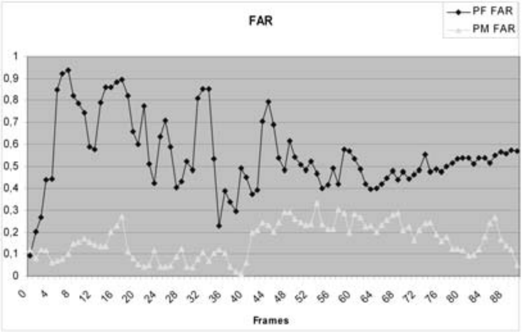

Fig. 7. Tracking Results Chart(FAR) - PETS Sequence.

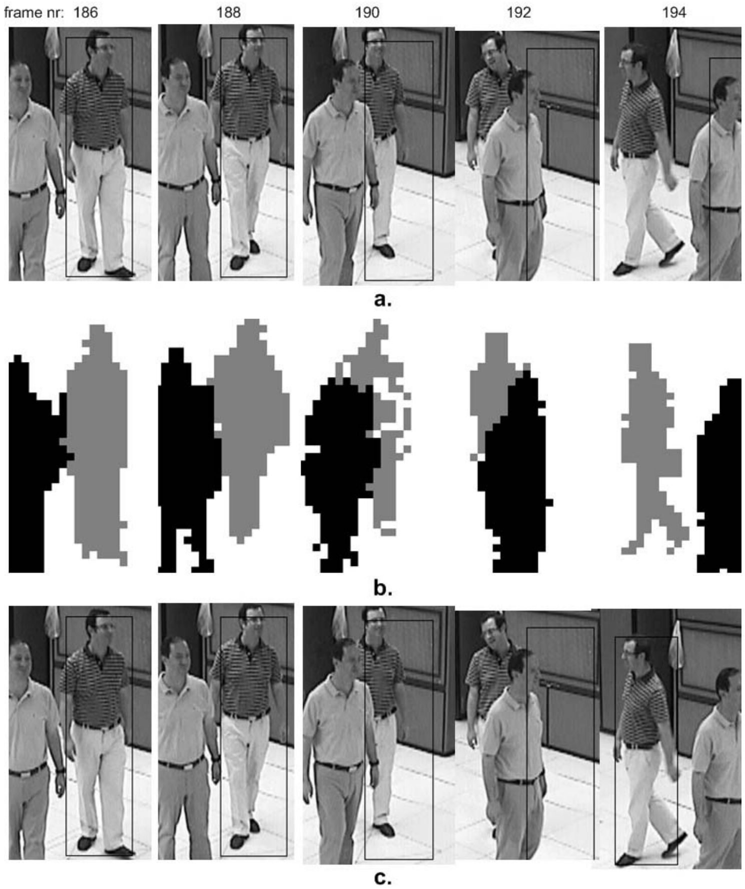

Fig. 8. Typical recovery procedure a. tracker of occluded target without recovery b. associated masks after training for both moving targets c. tracker of occluded target with recovery procedure

of the considered cameras [50]. The results are displayed in Table 3. As we can notice, RANSAC and MAPSAC algorithms yielded the best results, while epipolar geometry estimation using calibrated cameras yielded the poorest one, mainly due to the difficulty of the camera calibration task, which introduced a great deal of correlated error in the calculations. The precision of the mapping has been measured to be better than ten pixels. For the calibration,

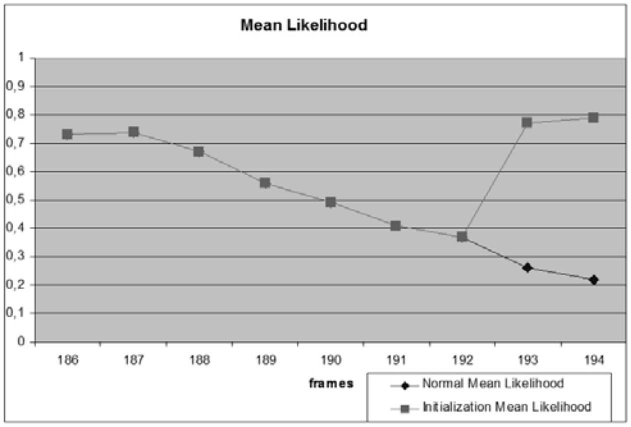

Fig. 9. The mean likelihood of all particles for the occlusion case in fig 8 with and without recovery activation

Method		PF		PM		
Seq	Frames	TDR	FAR	TDR	FAR	N
PETS station	1300	0.42	0.79	0.57	0.58	100
bumper car	2300	0.32	0.87	0.65	0.21	100

Table 2
Results using selected parts of the PETS-station sequences and sequences from our bumper car human tracking scenario. PF denotes the classical particle filter while PM our method (without recovery). TDR, FAR refer to average performance for the sequences. N is the number of particles which is proportional to the computational cost and is common for both PM and PF.

Method	**Mean error** (per cent)
Calibrated cameras	28.30
Ordinary least squares	11.37
Orthogonal regression	9.81
RANSAC	1.02
MAPSAC	1.01

Table 3
Evaluation of epipolar geometry estimation methods

a chess board in A0 format was used to extract the needed points. A result of this is optimized for a few meters close to the calibration map. There is also the limitation that the common area between two handover cameras should be enough to handle the object speeds and direction. A typical result of the multicamera tracking is displayed in figure 10.

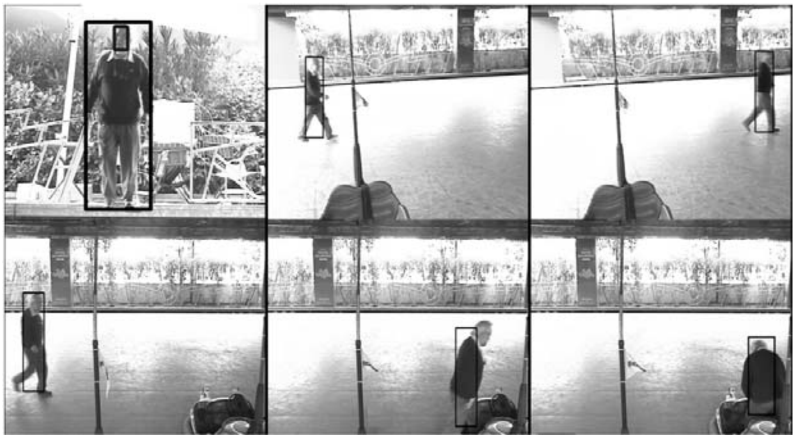

Fig. 10. Recognition and multicamera tracking. Initially the face is detected and the visitor recognised (i-cam). A hand-off between the i-cam and the t-cam1 camera takes place. The handoff between the t-cam1 and the t-cam2 follows. The visitor is associated to the bumper car in which he enters.

5.2.2 On-demand

The director's goal is to include the visitors in the video and to capture the attitude in the attraction, i.e., excitement in the case of the bumper car scenario, so that visitors can relive their experience. How this will be achieved, may be sometimes subjective. In our scenario a number of rules were run to assert a number of 'shot' facts that contain valid shots a film can choose from. These shots contained the visitor and some would overlap temporally (i.e. a choice must be made between two or more cameras). Metadata such as 'EnterView', 'CentreView', 'LeaveView' were present as attributes to a 'clip' fact that could be used to help decide which shots to use. This metadata was extracted from the MPEG-7 semantic annotations.

A film is a chain of selected shots in temporal order. Rules were created to turn 'clips' into 'shots' in a way compatible with the film's ethos. These rules were:

(1) The promotional material and stock content are added at the beginning of the video.
(2) At the first step only the shots associated with the specific visitor or associated bumper car are selected.
(3) Shots with duration less than one second are excluded.
(4) When the target leaves the FOV of a camera to enter of FOV of another one change view in the middle of the period that both cameras view the target.
(5) If there are time gaps, i.e., the visitor does not appear in a camera for a time period, e.g., due to occlusion or if entering a region not covered by the static cameras then we acquire footage from the onboard cameras.

Our director has spent approximately two man-days to find the appropriate camera positions and define the rules that produce the desired artistic result.

5.3 Integrated test

Each of the above production rules can be associated to certain types of erroneous decisions, which result in deviations from the optimal content. As for rule 1 it is the failure to automatically include promotional content in the produced video (practically never happens). Failures in 2 regard the selection of wrong frames (i.e., frames in which the target is not well visible or totally absent) and are calculated as the percentage of the total number of frames. Failures in 3 are a consequence of wrong labeling and they regard the false insertion (false positive) or omission (false negative) of video content due to incorrectly perceived duration (exceeding or not exceeding the cut-off duration threshold). They are calculated as percentage of the total number of frames. Failure in 4 regards the unnecessary hand off (false positive) and the failure to perform hand off (false negative). It is calculated as the percentage of the total number of hand offs. Failures in 5 regard the unnecessary insertions of shots from the onboard camera when the system believes that the target is not visible (false positives) and the omissions to insert such a shot (false negatives) when the system believes that the target is visible. It is calculated as percentage of the total number of frames.

To evaluate the system performance we have compared the system output with the output provided by a human editor who has to follow the same rules. The results are given in table 4. The discrepancies between the produced videos are due to processing errors in the system. Footage consisting of approximately 90000 frames from six cameras have been processed and evaluated to produce these results.

Generally the videos are aesthetically pleasing for the visitors (a) if they are included in them and (b) if the atmosphere in the attraction is reproduced. The (a) has been quantified by rule 2. Missing totally the visitors for some time duration does not create blank intervals in the final video, because clips from onboard (or overview) cameras are used so that the related shots can be relevant even in cases of such errors. As for (b), 19 out of 26 visitors who were surveyed thought that the atmosphere was successfully captured in the produced videos.

The produced final product had a duration of 3 minutes (as long as a ride) and about ten different videos were produced using the same scenario.

Rule	**Errors** (per cent)
Rule 1 (failure to insert intro)	0
Rule 2 (wrong frame label)	10.69
Rule 3 (wrong inclusion/exclusion)	1.41/1.30
Rule 4 (wrong handoff/failure to perform handoff)	16.15/19.85
Rule 5 (omit insert/false insert)	2.68/3.02

Table 4
Overall evaluation of application scenario

6 Discussion

Given the difficulty of the task the experimental results are promising. The target is not always successfully tracked, however the acquired content can be always relevant, provided that the initial identification has been done correctly. This is achieved by defining appropriate rules. For example when the target visitor is lost (in the bumper car scenario) then the system uses the clips captured from onboard cameras, or from cameras viewing the whole ride thus ensuring that the visitor is viewed.

The requirements for operating a personalized video acquisition system in venues or theme parks are very demanding. The human identification has to be executed in a very reliable manner, so that the whole processing chain can be correctly initiated. It is obvious that the face recognition technology as it is today does not fully cover this requirement despite the significant progress made recently. We have achieved higher success rates by (a) taking many images from various viewpoints during registration (b) allowing recognition through a sequence of frames and not a single frame (c) using information from body colors (d) controlling illumination as much as possible and compensating for brightness changes. The more progress is made in the face recognition field, the more applicable our approach will be.

The proposed template-based automated content creation can be applied to several domains, provided that the digital content can be automatically associated with semantics, e.g., by using techniques like the one proposed in [51]. Provided that the rules are appropriate and that the recognition/retrieval of content is acceptable such a method is expected to provide aesthetically pleasing results.

We have seen that with appropriate tuning of the number of particles for tracking and the number of iterations for modeling and recovery it is possible to achieve real time performance having acceptable results. At the rather small scale described in this work the production stage (DVD burning) is the main bottleneck. The scalability of the proposed recognition approach in large venues with many more visitors is still to be proven, however the use of cameras in combination with other sensors, e.g., RF, would help in decreasing significantly the search space. The difference of that case with the approach proposed in [2] is that the RF sensors will be used only for identification and not for localisation and camera triggering through tracking, which can be problematic near metal structures and for trajectories that are not predetermined.

A serious concern with the proposed approach is the illumination variations, which may affect identification and tracking performance. Thus the investigation of the light conditions between the registration area and the identification area, as well as in the tracking area is one of the key aspects. In our experiments the lighting conditions themselves were improved by providing standalone lighting that was consistent and adequate. We experienced difficulties with the sunlight levels changing during the day and invalidating the camera calibration settings. To resolve this we have used automatic camera calibration and color constancy techniques. The bumper car ride itself, once started, dimmed all lights and activated a number of flashing lights. These flashing lights were problematic but more consistent independent lighting helped.

The real-time subsystem was integrated with a streaming architecture that was evaluated and was found to be functional for PCs and mobile phones over a 3rd generation mobile network. There was a delay of several seconds (normally less than ten) between the actual time and the time that the e-visitors perceived. This was due to the inherent buffering mechanisms that the streaming servers employ to compensate for network problems. Nevertheless some streaming services can be based on the described scenario.

Setting up a demo area was the most difficult part of the integration. First thing to do was to define the area of the demo and mark all possible key areas in the demo - identification area, handover tracking sequence and registration. Then the cameras had to be configured to view the key points in the ride. The illumination needed to be optimized to minimize variations due to external light. Tracking handover (or human and car objects) required camera calibration to understand the same x-y coordinates of the object. Finally the appropriate rules needed to be defined so that the final DVD will be aesthetically pleasing.

7 Conclusion

A new system for identifying tracking using vision automatically generating personalized video content using non-intrusive technology has been presented.

Furthermore, we found that new services like online streaming for venue visitors are feasible for the proposed scenario. The system integrates some of the latest advancements and has been applied in a real environment of an entertainment park. The results have shown that the approach is feasible if the illumination is controlled and the environment carefully modified to match the capabilities of the identification and tracking algorithms.

In the framework of this application we have contributed a novel methodology for efficient tracking which outperforms the standard particle filter fusion scheme, a new fusion scheme for human identification which is better than face recognition and a feature-augmented grammar based system for automated production from annotated content.

The main difficulties of the overall approach have to do with identification and tracking under varying illumination, but as the related algorithms progress and the hardware costs decrease, our system, which has been deployed for research purposes, will come closer to commercial exploitation.

Acknowledgment

Earlier parts of this work have been performed in the framework of the STREP project ”POLYMNIA: Personalized Leisure and Entertainment over Cross Media Intelligent Platforms” (http://polymnia.pc.unicatt.it/). It has been partially funded by the EU 6th Framework Programme, grant number IST-2-004357.

References

[1] Disney, Disney photopass, Available on: http://www.disneyphotopass.com (2008).

[2] AltonTowers, Ride the rides, Available on: http://www.yourdayataltontowers.com (2008).

[3] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems Man and Cybernetics - Part C 34 (3) (2004) 334–352.

[4] A. O'Toole, P. J. Philips, P. Meer, F. Jiang, J. Ayyad, N. Penard, H. Abdi, Face recognition algorithms surpass humans matching faces over changes in illumination, IEEE Trans. Pattern Anal. Mach. Intell. 29 (9) (2007) 1642–1646.

[5] A. Adler, M. E. Schuckers, Comparing human and automatic face recognition performance, IEEE Transactions on Systems, Man and Cybernetics Part B 37 (5) (2007) 1248–1255.

[6] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, W. Worek, Preliminary face recognition grand challenge results, in: FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 2006, pp. 15–24.

[7] D. Klünder, M. Hähnel, K.-F. Kraiss, Color and texture features for person recognition, in: International Joint Conference on Neural Networks IJCNN 2004, Vol. 1, 2004, pp. 647–652, iSBN 0-7803-8360-5.

[8] C. Nakajima, M. Pontil, B. Heisele, T. Poggio, Full-body person recognition system, Pattern Recognition 36 (9) (2003) 1997–2006.

[9] G. Sfikas, C. Constantinopoulos, A. Likas, N. P. Galatsanos, An analytic distance metric for gaussian mixture models with application in image retrieval, in: ICANN (2), 2005, pp. 835–840.

[10] K. Barnard, V. Cardei, B. Funt, A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data, Image Processing, IEEE Transactions on 11 (9) (2002) 972–984.

[11] J.-P. Renno, D. Makris, T. Ellis, G. A. Jones, Application and evaluation of colour constancy in visual surveillance, in: 2nd Joint IEEE International Workshop on VS-PETS, IEEE Computer Society, Washington, DC, USA, 2005, pp. 301–308.

[12] Y.-C. Chang, J. F. Reid, Rgb calibration for color image analysis in machine vision, Image Processing, IEEE Transactions on 5 (10) (1996) 1414–1422.

[13] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Comput. Surv. 38 (4).

[14] C. Andrieu, N. de Freitas, A. Doucet, M. I. Jordan, An introduction to mcmc for machine learning, Machine Learning V50 (1) (2003) 5–43.

[15] M. Isard, A. Blake, Condensation – conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.

[16] P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I, Springer-Verlag, London, UK, 2002, pp. 661–675.

[17] K. Nummiaro, E. Koller-Meier, L. J. V. Gool, An adaptive color-based particle filter., Image Vision Comput. 21 (1) (2003) 99–110.

[18] E. Ozyildiz, N. Krahnstoever, R. Sharma, Adaptive texture and color segmentation for tracking moving objects., Pattern Recognition 35 (10) (2002) 2013–2029.

[19] H. Sidenbladh, M. J. Black, D. J. Fleet, Stochastic tracking of 3d human figures using 2d image motion, in: ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II, Springer-Verlag, London, UK, 2000, pp. 702–718.

[20] J. Odobez, D. Gatica Perez, S. Ba, Embedding motion in model-based stochastic tracking, IEEE Transactions on Image Processing 15 (11) (2006) 3514–3530.

[21] P. Perez, J. Vermaak, A. Blake, Data fusion for visual tracking with particles, Proceedings of the IEEE 92 (3) (2004) 495–513.

[22] M. Spengler, B. Schiele, Towards robust multi-cue integration for visual tracking, in: ICVS '01: Proceedings of the Second International Workshop on Computer Vision Systems, Springer-Verlag, London, UK, 2001, pp. 93–106.

[23] M. Isard, A. Blake, ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework, Lecture Notes in Computer Science 1406 (1998) 893–908.

[24] P. Li, F. Chaumette, Image cues fusion for contour tracking based on particle filter, in: J. Perales, B. Draper (Eds.), Int. Workshop on articulated motion and deformable objects, AMDO'04, Vol. 3179 of Lecture Notes in Computer Science, Palma de Mallorca, Spain, 2004, pp. 99–107.

[25] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, S. Shafer, Multi-camera multi-person tracking for easyliving, in: VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000), IEEE Computer Society, Washington, DC, USA, 2000, p. 3.

[26] A. Mittal, L. S. Davis, M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene, Int. J. Comput. Vision 51 (3) (2003) 189–203.

[27] P. H. Kelly, A. Katkere, D. Y. Kuramura, S. Moezzi, S. Chatterjee, An architecture for multiple perspective interactive video, in: MULTIMEDIA '95: Proceedings of the third ACM international conference on Multimedia, ACM Press, New York, NY, USA, 1995, pp. 201–212.

[28] Q. Cai, J. Aggarwal, Tracking human motion in structured environments using a distributed-camera system, PAMI 21 (11) (1999) 1241–1247.

[29] S. M. Khan, M. Shah, A multiview approach to tracking people in crowded scenes using a planar homography constraint, in: ECCV (4), 2006, pp. 133–146.

[30] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, S. Maybank, Principal axis-based correspondence between multiple cameras for people tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 663–671.

[31] S. Butler, A. P. Parkes, Film sequence generation strategies for automatic intelligent video editing, Applied Artificial Intelligence 11 (4) (1997) 367–388.

[32] F. Nack, A. Parkes, The application of video semantics and theme representation in automated video editing, Multimedia Tools Appl. 4 (1) (1997) 57–83.

[33] S. Yip, E. Leu, H. Howe, The automatic video editor, in: MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, ACM Press, New York, NY, USA, 2003, pp. 596–597.

[34] S. Bocconi, Semantic-aware automatic video editing, in: MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, ACM Press, New York, NY, USA, 2004, pp. 971–972.

[35] L. Chen, M. T. Özsu, Rule-based scene extraction from video, in: ICIP (2), 2002, pp. 737–740.

[36] Z. Zivkovic, F. van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognition Letters 27 (7) (2006) 773–780.

[37] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, Proc. CVPR 1 (2001) 511–518.

[38] M. Turk, A. P. Pentland, Eigenfaces for recognition., Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.

[39] I. Haritaoglu, D. Harwood, L. S. David, W4: Real-time surveillance of people and their activities, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 809–830.

[40] A. Doucet, S. Godsill, C. Andrieu, On sequential monte carlo sampling methods for bayesian filtering, Statistics and Computing 10 (3) (2000) 197–208.

[41] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking, IEEE Transactions on Signal Processing 50 (2) (2002) 174–188.

[42] A. Makris, D. Kosmopoulos, S. Perantonis, S. Theodoridis, A hierarchical feature fusion framework for adaptive visual tracking, in: ICIP '07: Proceedings of the 2007 IEEE International Conference on Image Processing, Vol. IV, 2007, pp. 289–292.

[43] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, ISBN: 0521540518, 2004.

[44] W3C, W3c synchronised multimedia homepage, Available on: http://www.w3.org/AudioVideo (2007).

[45] J. Allen, Natural language understanding (2nd ed.), Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1995.

[46] M. Mateas, P. Vanouse, S. Domike, Generation of ideologically-biased historical documentaries, in: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, AAAI Press / The MIT Press, 2000, pp. 236–242.

[47] J. C. Giarratano, Clips user's guide, found at url: http://www.ghg.net/clips/download/documentation/usrguide.pdf (2002).

[48] A. Frolov, Scarse: Scanner calibration reasonably easy url: //http://www.scarse.org/ (2007).

[49] D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R. B. Fisher, J. S. Victor, J. L. Crowley, Comparison of target detection algorithms using adaptive background models, 15-16 Oct. 2005, pp. 113–120.

[50] J.-Y. Bouguet, Camera calibration toolbox for matlab found at url: http://www.vision.caltech.edu/bouguetj (2007).

[51] B. Fasel, L. J. V. Gool, Interactive museum guide: Accurate retrieval of object descriptions, in: Adaptive Multimedia Retrieval, 2006, pp. 179–191.