AUTOMATED PRODUCTION OF PERSONALIZED VIDEO CONTENT FOR VISITORS OF THEMATIC PARKS

National Technical University of Athens
Dept. of Electrical & Computer Engineering
157 73 Zografou, Athens, Greece
Email: adoulam@cs.ntua.gr

Abstract:

The proposed platform aims to develop an intelligent cross-media platform for personalised leisure and entertainment in thematic parks or venues. The system we develop allows the visitor to be the real protagonist in the venue. The platform caters for a great variety of media needs of the visitors: from recording their visit to provide a high-quality, customised souvenir, to embedding personalised electronic content into their activities in real time.

The platform incorporates new and innovative solutions that cover and improve digital content acquisition and creation, management and processing, access, retrieval, manipulation and delivery of digital content. The platform is equipped with innovative imaging technologies for real time detection, localisation and tracking of “human content”, i.e., the human visitor within the recording being made in real-time by the system. No constraints are imposed on the variation of the environment.

1 Introduction

One of the visitor’s main concerns when he/she visits a thematic park and/or a venue, is to capture his/her visit, either by photographing or by videotaping the venue and members of their group. However the results are usually of an amateur quality standard, due to the visitor’s filming experience, to their equipment quality and configuration, to the necessarily limited effort they can make and time they can spare, etc.

The purpose of this paper is to provide an intelligent, personalized and unobtrusive solution in order to capture and customize a high-quality record of leisure and entertainment experiences for visitors of thematic venues. In addition, this paper also aims to address in an optimal and substantial way the need of people who are travelling or otherwise visiting theme-based entertainment venues to keep in touch with their friends and family and to share their experience with them.

The main scientific and technological objectives of the this paper are to i) identify important content, ii) automatically synthesise new digital content, iii) allow efficient content-based delivery, iv) develop new mechanisms for content based access and retrieval across different media platforms through efficient adaptation mechanisms.

As mentioned before, the first objective is to identify important content within multi-camera video data, i.e., individual visitors whose visit is being tracked, as well as important context which is necessary for content annotation and augmentation. The visitors, as part of the digital content are known from their registration time while the context is permanent and known in advance. No constraints are imposed on the variation of the environment and wholly natural scenes can be processed successfully.

Furthermore, the proposed platform will automatically synthesize new digital content for visitors as well as for e-visitors. The most information-rich views of a visitor will be detected (e.g. the system can find the camera with the best view). The content will be augmented with annotations, explanations (e.g. of what the visitor or e-visitor is observing) and animations.

Efficient content-based delivery of all supported media types is allowed to access a wide range of terminal devices (workstations, PCs, PDAs, Internet, mobile phone, etc), over a broad variety of channels and networks supporting standardized protocols (local high speed networks, Internet-Protocol (IP)-based network, wireless network, etc). Adaptive real time media delivery, streaming and casting will be supported in order to attract the interest of geographically dispersed people to the content of the venue.

We arrive to the realisation of the aforementioned objectives by developing a system for providing the following innovative research solutions, in the framework of personalised cross-media and entertainment platform through: (i) advanced imaging technologies for content extraction, identification and representation, performed in real time wherever applicable, (ii) development of integrated content programming schemes, which support the aforementioned technologies, (iii) solutions for access and retrieval of the generated and semantically annotated content, which will be operational across different media platforms.
1.1 Previous Approaches and Innovation

Some works have been proposed in the literature concerning the above mentioned issues.

In the framework of content acquisition and detection, several algorithms have been proposed in the literature. Most of them restrict the meaning of content to homogeneous regions of similar colour [1][2], motion [3][4][5], or depth [6] characteristics. In the recent years, approaches for semantic content detection and acquisition have also been applied by considering constraints on the object type, e.g., humans [7]. Although this is the type of semantic objects that this paper deals with, most of the currently presented approaches make high processing and memory demands, which renders them practically impossible to be applied in real time framework as the proposed platform requires. In addition, the main drawback of these algorithms is the fact that they are restricted to certain type of environment (e.g., low complicated content and motion, almost uniform background). Thus, they cannot be applied with high efficiency for the objectives of the proposed platform, where no constraints on the type of the environmental conditions, background complexity, motion characteristics, illumination conditions are considered. This is also enhanced by the fact that in a real commercial application, like the presented one, the users are highly demanded and they usually are not willing to pay for an inaccurate system.

All the aforementioned problems will be overcome in the framework of the this paper by proposing new, innovative solutions which guarantee, on the one hand, high efficiency and robustness and on the other, low processing and memory requirements.

In the proposed platform, human content tracking is considered. For human tracking numerous algorithms have been proposed in the literature [3][4]. The algorithms apply motion information, boundaries detection and active contours, snakes, deformable models, (parametric and non-parametric), simple classification schemes and so on. These approaches however present a number of limitations so that they cannot be applied with high efficiency in a complex environment like the one proposed in this paper. For instance, their performance highly deteriorates in abrupt changes of the environmental conditions, complicated background motion and content, and complicated non-rigid motion. In addition, they are not characterized by re-activated mechanisms which can re-act and re-apply the tracking algorithms in cases that deterioration of the tracking performance is encountered [8].

The proposed innovative technological solutions aim at overcoming all the aforementioned limitations. In

Figure 1. The basic modules of the proposed system.
particular, the platform will be equipped with efficient and low computationally demanded tracking algorithms, which a) can perceive changes of the environmental conditions and re-activate each time this is appropriate, b) they are robust to environment of complicated foreground/background motion, and c) they provide sufficient performance in complicated situations such as occluded foreground objects, illumination changes.

In the framework of this paper, advance multimedia technologies will be incorporated for content representation and description. Efficient content representation allows easy content manipulation, management, delivery, retrieval and access. Content representation is supported by the MPEG-7 standardization activities [9][10][11]. However, the standard provides a generic open framework for describing different solutions in content representation and not the algorithms for the content representation itself, which are left open to the developers [10][11]. In the developed platform, two different innovative aspects for content representation are addressed; the spatio-temporal and the semantic representation.

The spatio-temporal content representation describes the relationships of different acquired media content in space and time. Three dimensional graphs are developed for this reason, the links of which indicate the spatio-temporal relations, while the nodes the content properties/characteristics. Different content hierarchy is defined as the difference of the sum of the pixel values in equally sized adjacent rectangular regions; the regions may be of various numbers, sizes and configurations, thus forming simple or more complex features. The features can be computed at any location or scale in constant time using the idea of summed area tables, known from the field of computer graphics. The calculated features are 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 proper features.

For each feature \( j \), a classifier \( h_j \) using this single feature is trained, which returns 0 or 1 for negative or positive examples respectively. The error is given by

\[
e_j = \sum_i w_i | h_j(x_i) - y_i |
\]

where \( x_i \) is the training sample \((i=1,n), \ w_i \) the corresponding weight of the sample and \( y_i \) the ground truth. Then the classifier with the lowest error \( e_j \) is selected and the weights are updated according to

\[
w_i' = w_i \left( \frac{e_j}{1 - e_j} \right)^{-q_i}
\]

where \( e_i \) is 0 if \( x_i \) is classified correctly or 1 otherwise. The procedure is repeated until we select the \( T \) features that satisfy the performance and accuracy requirements of the classifier. The final classifier is applied to many rectangular regions in the image and gives 1 (true) if

\[
\sum_{i=1,T} \log \frac{1 - e_i}{e_i} \cdot h_i(x) \geq \frac{1}{2} \sum_{i=1,T} \log \frac{1 - e_i}{e_i}
\]

or 0 (false) otherwise. The early stage classifiers seek to minimize false negatives using fewer features due to performance.

The rotated Haar-like features and the post optimization procedures may increase recognition rates as shown by [14] The sensitivity of the method to big rotations is known and can be enhanced by using methods such as inserting the Haar-like facial features into deformable graphs.
For the purposes of our research a fraction of the identified face bounding rectangle is used for training the skin-color classifier (training region-of-interest) as presented in the following subsection.

4 Background Separation

For the general purpose of video processing, the background is usually considered as the scene without the presence of objects of interest, such as human objects or moving vehicles. Background is usually composed of non-living objects that remain passively in the scene. In a video about a general environment, the background can consist of both stationary and moving objects. The stationary background objects can be walls, doors, and furniture, paintings in an indoor scene, as well as buildings, vegetation, and ground surfaces in an outdoor scene. The moving background objects can be waving tree branches, flickering water surfaces, or screens of computers, wavering curtains, moving fans, running escalators, roller coasters and many more.

The background might be undergoing two types of changes over the time. One is the gradual changes caused by natural lighting variations, e.g., the change of illumination from day to night.

The other is the sudden changes. The global sudden changes may be caused by switching on/off some lights or the change of view angle of a camera, and the local sudden changes may be caused by removing or depositing the background objects, e.g., moving a chair to a different position. Besides, the foreground object might be converted to be a background object, such as a car moving into a parking lot. In some cases, a background pixel may have multiple states, such as sunny and cloudy scenes.

The following requirements must be satisfied:

- R1: Bootstrapping: Description: A training period absent of foreground objects is not available in some environments.
- R2: Time of day Gradual illumination changes alter the appearance of the background.
- R3: Sudden Local change: Background objects are moved. These objects should not be considered part of the foreground forever after
- R4: Sudden global change Sudden changes in illumination, other scene parameters or camera motion alter the appearance of the background.
- R5: Periodic change Backgrounds can vacillate, e.g., waving trees, in a manner which has a certain degree of periodicity.
- R6: Shadows Description: Shadow areas in the image are falsely considered as foreground causing problems in the recognition of shape.
- R7: Performance Description: The system should run at frame rates which are able to give the target motion without missing significant information.

Considering all the requirements mentioned in the previous section we propose in this section a non-parametric method to satisfy them.

In this section we define the features that we use for representing the background considering the ability to handle the above problems and for real time performance.

The color is a feature that is able to describe statically the individual pixels. It is an efficient and simple representation and can easily lead to multimodal distributions. Its disadvantage is that if used alone it does not consider the color change in time, which is very useful for identifying periodically changing background.

The color space ideally should have decoupled channels, so that we are able to examine separate phenomena in a decoupled manner. The HSV and YCrCb spaces cover these requirements.

The use of pixel color in current and previous moments is simple and efficient (with some additional memory cost) and is able to capture periodicity and also motion in an implicit way.

The use of motion data is able to represent motion in a direct way. However the calculation of motion through the optical flow and coherence constraints is quite costly in terms of performance. Additionally the motion calculation for moving objects that are homogeneous is problematic and wrong motion vectors are calculated “behind” the moving objects as they move.

The proposed approach has as follows:

For the representation at low level we employ a feature vector $f$ using the current and previous average color values of pixels in 3x3 blocks in the HSV color space. Specifically, $f = [h, s, v, h_{-1}, s_{-1}, v_{-1}]^T$, where each vector field takes values from 0 to 255. This offers a number of advantages:

- Simple representation since we use only six values to represent nine pixels
- The neighboring pixels are considered thus reducing salt-type noise
- The periodicity is considered
- The motion is captured implicitly
- The channels are decoupled
- Shadows can be handled more easily because of the relative immunity of the hue and due to the known effects in the luminance and saturation channel.

The background is represented as a series of $N$ $f$ feature vectors forming a temporal queue. In each frame cycle if the new vector is decided to be part of the background then the head of the queue is discarded and the new background vector is added to the tail of the queue. This handles the R2 requirement.

We seek to find the $k$- nearest neighbors of the $f$, using the $L_1$-distance ($k<<N$). For this purpose to reduce the computational load the search space has been partitioned into 6D blocks, so that a similar techniques with that used for the $k$-NN classification algorithm can
be applied such specifically the LSH described in [15]. The relatively small dimensionality makes this approach viable. If the $L_i$-distance for all of them is lower than a predefined threshold $T_{L_i}$ then the $f_i$ represents a background block and is added to the tail of the temporal queue and the head of the queue is discarded.

Some special care is required for calculating the hue difference, due to its cyclic nature. The difference in the hue fields is calculated by

$$d_{	ext{hue}} = \min \| f_{\text{hue1}} - f_{\text{hue2}} \|, \| f_{\text{hue1}} - 255 \|$$  (4)

It is obvious that the selection of $k$ and $T_{L_i}$ affect the quality of the results. Big values for $k$ and low values for $T_{L_i}$ make the system more sensitive to image differences, while the opposite choice makes a system that is more adaptable.

Apart from background representation there is a need for foreground representation in cases that for the same block the feature vectors are sequentially classified as foreground. This handles the R3 requirement (sudden local change) but also the bootstrapping requirement R1 (after making an arbitrary background initialization).

For this purpose we check for consistency the $L_i$-distance of the first three fields of $f_i$, between subsequent vectors. If we represent as $f_i = [h, s, v]^T$ and $f_{i-1}$ the average vector for the last $N$ frames then the test that we execute is given by:

$$d_{L_1}(a f_i + (1-a)f_{i-1}, f_{i-1}) < T_{L_1}/2 \text{ where } a = 1/N$$  (5)

If the distance is lower than the threshold then the block color values are converging to the same vector and we can assume that the block belongs now to the foreground. Then $k$ vectors of value $f_i$ are added to the queue tail and the same number of vectors is discarded from the head to signify the background update. A limitation here is that the new object that has entered the scene does not move periodically or that the object that has left the scene did not reveal periodically moving objects.

The method can be easily extended (and will be extended in the next steps) with bootstrapping for moving background by checking that each block has $k$ nearest neighbors with distance lower than $T$ that belong to the $N$ last vectors.

For handling the requirement R6 we perform some post processing in the foreground regions to suppress shadows. As mentioned before the hue is not significantly affected by shadows and saturation is lower as well as the luminance. Therefore a foreground region is considered to be shadow if the following three conditions hold simultaneously for each of the $k$ neighbors:

1. $T_{h_i} \leq v_i \leq T_{h_{\text{avg}}}$, where $T_{h_i}$ and $T_{h_{\text{avg}}}$ upper and lower thresholds $v_i$ the intensity values of the $i$-th neighbor
2. $s_i \leq T_{s_i}$, where $T_{h_i}$ a threshold value and $s_i$ the saturation value of the $i$-th neighbor
3. $h_i \leq T_{h_{i-1}}$, where $T_{h_{i-1}}$ a threshold value and $h_i$ the saturation value of the $i$-th neighbor

The same requirement has to hold also for neighboring pixels, since the shadow blocks are assumed to form regions and are not isolated.

5 The Proposed Human Tracking Algorithm

Tracking cameras (T-camera) are used to track the visitors throughout a venue. A T-camera has an overlap of the field of view with one or several other cameras of the same type. The human tracking module gets informed if in these overlapping view areas a person has been identified by a human identification module or if there is a handoff of an identified person from another tracking system.

The human tracking module processes the images of the T-cameras in real-time and determines the movement of identified visitors. The tracked locations of the human objects are saved in a MPEG-7 document.

5.1 Tracking Algorithm

There are many publications about real time tracking algorithms. A survey of methods used in visual surveillance can be seen in [16]. There are some publications from large research projects e.g. from the Visual Surveillance and Monitoring (VSAM) project [17] and the European Framework V project ADVISOR [18] about visual surveillance in metro stations. Often referenced systems are, among others, the real-time surveillance system Wlas [19] and the PFinder system [20] for recovering 3D descriptions of people.

The main processing steps of the implemented tracking algorithm are shown in Figure 2. Most of the processing steps can be found in nearly every visual surveillance system. But for each step a lot of approaches have been proposed in the literature.

The proposed algorithm is a particle filter based approach to the Bayesian tracking. As in Sampling Importance Resampling [26], the posterior is approximated by a set of samples (particles) which are propagated through time. This algorithm requires that the state can be partitioned into a set of sub-states. For each of those sub-states there is a corresponding measurement model. These models are probabilistically linked which means that if we don’t know the sub-state corresponding to a model we can evaluate its conditional probability given the rest of the sub-states. For each model one or more visual cues are used to define the likelihood. The models are arranged in increasing complexity order. The simpler ones, which might be a set of salient points within the object or a set of blobs with the same color or texture, are located in the beginning of the state vector. More complex models such as parametric curves or human models are located at the end of the vector. Simpler models are easier to update but they are not robust and do not offer a detailed target representation. Complex models in contrary are more difficult to update but they offer a very detailed target representation and when they are supported by multiple visual representation are very robust. Here we use one main model to define the target region, which is usually the most complex one. This
model is placed last at the feature vector. When a new frame arrives, the rest of the models are updated first and because they are linked to the main model they provide information about its expected position.

In the classical particle filtering approach to tracking the state evolution is used to produce the new samples. However, for models with many parameters many samples are required to sample adequately from the state evolution. A better proposal distribution using some sort of low level information from the current frame can improve the sampling efficiency [27]. Here we propose a model hierarchy, where the simpler models narrow the search space for the more complicated ones. The state can be written as:

\[ x = [x_{[1]}, x_{[2]}, \ldots, x_{[M]}] \]  

(6)

where M is the number of sub-states. Each sub-state, \( x_{[i]} \), has \( K_i \) parameters and corresponds to a measurement model \( z_{[i]} \):

\[ z = [z_{[1]}, z_{[2]}, \ldots, z_{[M]}] \]  

(7)

We assume that the conditional probabilities of a sub-state at time \( t \), given the others

\[ p(x_{[i]} | \{x_{[j]} : j = 1..M, j \neq i\}) \]  

(8)

are known. The likelihood \( p(z_{[i]} | x_{[i]} ) \) and the state evolution \( p(x_{[i]} | x_{[i-1]} ) \) for each sub-state are also known. The update of the particle set for each sub-state takes place in predefined sequential fashion. When the particle set for a sub-state is updated then it is used to update the subsequent sub-states in the vector. The steps of the algorithm are the following:

- Update using the classical Sampling Importance Resampling algorithm the state of the first sub-state. In the case that the first cue is a set of salient points a deterministic algorithm such as KLT [25] can also be used.
- For every other sub-state update the sample set by sampling from the conditional distribution given the previous sub-states in the state vector:
- Evaluate the performance of each model and remove from the state vector those who seem to have lost track. At this step models with low likelihood for every particle are removed. Using the current target representation maintained by the main model new models are trying to initialize if possible. This step may not take place at every frame but only when few models remain active.

The proposed algorithm breaks the initial problem into \( M \) sub-problems The sub-states are arranged in the vector in increasing complexity order so that the sub-states with the simpler features are updated first which then guide the more complicated features in relevant regions. This way the dimensionality of the search space is reduced. In addition, fewer particles are required to search efficiently the state space, leading to lower computational complexity and allowing the algorithm to be able to run in real time.

Here we propose two simple models, the first one being a set of salient points within the object and the second the contour of the tracked object. The state vector becomes: \( x = [x_{[SP]}, x_{[c]}] \) where \( x_{[SP]} \) are the coordinates of the mean positions of the salient points and \( x_{[c]} \) are the parametric curve parameters defining the contour.

Each of those models may use one or more visual cues to define the likelihood. In this work we use two cues for the contour, the edge information and the color histogram. We first describe how we calculate the likelihood for each of these models and afterwards we will link them together by showing how the conditional probabilities are calculated.

**Salient Points:** The corners of the tracked object are used. This feature has only two parameters which are the mean image coordinates \( x_{[SP]} = [i_1, i_2] \) of the tracked points. For each tracked object \( N_p \) points are used but each one of them is updated independently so the dimension of the search space does not increase with the number of points. The likelihood of this feature is determined by calculating the sum of squared differences of a rectangle around a candidate feature and the original.

**Contour:** The contour serves two purposes, it is a rich feature especially for complex-shaped objects and additionally it can be used to define the object being tracked. The curves used are the parametric B-Splines.

As in [27], the B-Splines are initialized with the object’s contour and then the algorithm updates their position. However, allowing the algorithm to alter the control point’s coordinates increases the dimension of the problem and destabilizes the tracker because after a few repetitions the curve’s shape could change completely. To overcome this problem, the parameters passed to the algorithm are not the control point’s coordinates but a parameter vector \( x_{[c]} \), containing six parameters. These allow affine transformations of the initial curve.

The likelihood for this feature is calculated by measuring the existence of edges in the image at the specified position. More specifically edge detection is performed on consecutive line segments perpendicular to the curve. The distance from the closest edge to the curve determines the edge likelihood for each segment. The edge likelihood for the whole curve is then calculated by multiplying the likelihoods for all segments.

Additionally, the histogram of the object contained by the curve is calculated and compared with the template’s histogram. The histogram likelihood is determined by the difference of the candidate and reference histogram. The measure used is the Bhattacharyya distance.

### 6 Experimental Results

The experiments are conducted for each of the above mentioned categories, i.e., the background extraction and tracking.

#### 6.1 Background Separation

We have performed some initial evaluation experiments of the approach compared to the other
algorithms using the test sets with ground truth provided by the CAVIAR IST project [21]. We have defined some metrics which are given by:

Tracker detection rate:

\[
TDR = \frac{\text{true positive}}{\text{true positive}} - \frac{\text{false negative}}
\]

False alarm rate: \( \text{FAR} = \frac{\text{false positive}}{\text{true positive}} + \frac{\text{false negative}} \)

Due to the limitations of the ground truth data these have been calculated considering all the pixels that belong in the bounding rectangle of the foreground regions and not only those that are identified as foreground. Although the datasets are not optimal they give an indication of performance.

For each algorithm we have experimented with parameters which give the best results, close to the ones proposed by the authors of the respective papers. We have used ten videos of approximately 7000 frames in total.

The results seem promising since our method has not been extensively tested and optimised yet and since the ground truth data are not very precise due to consideration of the whole bounding rectangle. Further tests are going to be executed on-site and are expected to lead to even higher performance after optimizations that we were not able to do due to time limitations.

The results were executed versus the algorithms presented in [22] and [23]. For reasons of brevity we call the first algorithm, Algorithm 1, the algorithm, Algorithm 2 and our algorithm is the Algorithm 3. We selected the dataset from CAVIAR, and especially videos Walk1,2 For the Algorithms 1,2 the parameters selections are the default used in OpenCV header files. The choice of parameters for the Algorithm 3 is T=20, N=40, k=5.

For the foreground objects computed we determine Minimum Bounding Rectangles. (MBRs) since Caviar provides them. The methodology followed for measurements is as follows. Given an image and bounding rectangles in it, we define as the positive region, PR, the region consisting of the union of the interior of the bounding rectangles. For a frame and a series of MBRs returned by an algorithm, given the ground truth, we define the true positive percentage, as the fraction of the positive region of the ground truth belonging to the positive region of the algorithm. As the false positive percentage, we define the fraction of the positive region of the algorithm belonging to the complement of the region of the ground truth. Figures 2-5 present the time evolution of TP and FP for each frame in the dataset.
The results of data tracking are shown in Figure 7. The results show a person walking in a complex background. The results have been recorded from a real-life application scenario, with extremely complex background and foreground objects.

We have depicted in white contour the regions of the tracked persons, while the background has been illustrated as it was in the original images. Accurate human tracking is observed in all cases.

7 Conclusions

In this paper, we proposed an innovative human face detection algorithm, accompanying with background separation schemes and human tracking performance. The schemes have been included under a generalized platform used for visitors’ detection in a venue.

The results have been tested and compared with other approaches using objective criteria. Excellent performance is accomplished in all cases.

8 Acknowledgement

This work has been funded through the European IST project Polymnia (IST-2-004357).

References