A Cross Media Platform for Personalized Leisure & Entertainment: The POLYMNIA Approach

Vasilios Anagnostopoulos[1], Sotiris Chatzis[1], Constantinos Lalos[1], Anastasios Doulamis[1], Dimitrios Kosmopoulos[1] , Theodora Varvarigou[1], Helmut Neuschmied[2], Georg Thallinger[2], Stuart E. Middleton[3], Matthew Addis[3], , Eduardo Bustos[4] and Fabrizio Giorgini[5]

[1]*National Technical University of Athens Dept. of Electrical and Computer Engineering*
[2]*Joanneum Research, Austria*
[3]*IT Innovation Centre, University of Southampton, UK*
[4]*Telefonica I+D, Spain*
[5]*Giunti Interactive Labs, Italy*

E-mail: ***adoulam@cs.ntua.gr***

Abstract

The POLYMNIA project aims to develop an intelligent cross-media platform for personalised leisure and entertainment in thematic parks or venues. The system allows the visitors to be the real protagonist in the venue. Towards this goal, POLYMNIA platform is equipped with innovative imaging technologies for real time detection, localisation and tracking of "human content", i.e., the human visitor within the recoding being made in real-time by the system. No constraints are imposed on the variation of the environment. New, content-based media representation and organisation schemes will be developed to provide scalable, efficient and user-oriented description of the "human content", enabling efficient retrieval, access, and delivery across heterogeneous media platforms. In addition, adaptive mechanisms are employed to update the system response to the current users' information needs and preferences.

1. Introduction

One of the visitors' 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 visitors' filming experience, to their equipment quality and configuration, to the necessarily limited effort they can make and time they can spare, etc.

POLYMNIA provides 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, POLYMNIA 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 POLYMNIA project are to identify *important content*, to automatically synthesise new digital content, to allow efficient content-based delivery, to 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.

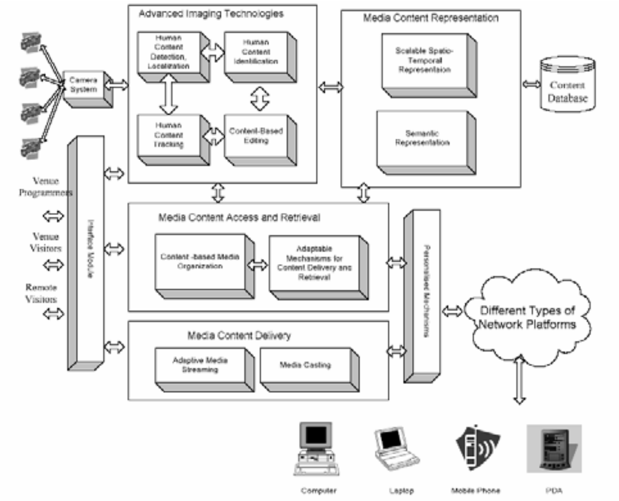
Furthermore, the POLYMNIA platform will automatically synthesize new digital content for visitors as well as for e-visitors. The most information-rich views of a visitor will be detected (e.g. the system can find the camera with the best view). The content will be augmented with annotations, explanations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

POLYMNIA arrives 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.

2. The POLYMNIA Architecture

An overview of the POLYMNIA approach is presented in the Figure 1. In particular, the POLYMNIA system comprises the following major architectural components



Advanced Imaging Technologies
Human Content Detection, Localization
Human Content Identification
Human Content Tracking
Content-Based Editing
Camera System
Media Content Representation
Scalable Spatio-Temporal Represenation
Semantic Representation
Content Database
Venue Programmers
Venue Visitors
Remote Visitors
Interface Module
Media Content Access and Retrieval
Content -based Media Organization
Adaptable Mechanisms for Content Delivery and Retrieval
Personalised Mechanisms
Different Types of Network Platforms
Media Content Delivery
Adaptive Media Streaming
Media Casting
Computer
Laptop
Mobile Phone
PDA

Figure 1. The basic modules of the proposed POLYMNIA system

2.1. Advanced Imaging Technologies

A camera system module controls an array of video cameras placed at various locations within the venue. Each camera locates the registered individuals as they enter that camera's zone. The cameras can pan, tilt and zoom to focus on the visitor(s) and may follow and record the visitor(s).

Apart form the camera system, the specific component uses advanced imaging technologies for real time human content extraction, localization and identification. The algorithms used are explained in more details in section 3.

2.2. Media Content Representation

The aim of this module is to describe: (a) the spatio-temporal relations of all the extracted human objects in a scenery (e.g., visitor named X stands next to visitor Y); (b) the relation of the extracted human objects in the context of the venue (e.g., the visitor stands before an exhibited item); (c) the semantic description and events of the extracted human objects in the scenery (e.g., John Smith is looking at the "Last Supper" of Leonardo Da Vinci, while his brother, Nick, stands next to him). Content representation of the extracted humans objects, enables, easy manipulation and management, content-based retrieval and access, content-based streaming and delivery.

2.3. Media Content Access and Retrieval

This specific component involves efficient and advanced multimedia technologies for accessing and retrieving the different types of media, extracted and identified by the detection and identification module. Media content access and retrieval exploits information based on the adopted content representation and description schemes. Two different units are included in this component.

The Content-based organization unit implements a non-linear organization of the created content;

The Adaptive Mechanisms are mainly utilized to fulfil the needs of e-visitors.

2.4. Media Content Delivery

The purpose of this module is to provide efficient algorithms for delivering the generated content, augmented with mixed reality and virtual reconstruction of the events or exhibits of the venue. This process involves techniques for real time media streaming and casting to terminal devices of different capabilities and cross networks of different characteristics.

In the following sections, we describe more analytical the above mentioned POLYMNIA components.

3. The Human Detection, Identification and Tracking Module

3.1 The Human Detection Unit

The detection-localization module attempts to provide a solution to the foreground classification problem compatible with the requirements outlined in [1].

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 $\mathbf{f}=(h_t,s_t,v_t,h_{t-1},s_{t-1},v_{t-1})$, where each vector field takes values from 0 to 255.

The background is represented as a series of N **f** feature vectors forming a temporal stack. There is also a test stack holding k ($k<<N$) foreground pixels. In each frame cycle if the new vector is decided to be part of the background then the new background vector is pushed into the temporal stack. For each new vector $\mathbf{f}_t$, we seek to find the k-nearest neighbors in the temporal queue using some norm, e.g. the L_2-distance. If the distance for all of them is lower than a predefined threshold T then the $\mathbf{f}_t$ represents a background block and is pushed into the temporal stack and the test stack is emptied. Else the $\mathbf{f}_t$ is pushed into the test stack.

In the case that the test stack is full we check for consistency the distance of the first three fields between subsequent elements. For the i element in the stack, let us represent as $\mathbf{c}_i=(h_i,s_i,v_i)$ and $\mathbf{c}$ the average value of $\mathbf{f}_{ci}$ for all the members of the test stack. Then the consistency test that we execute is given by $d(\mathbf{c}_i, \mathbf{c}) < T/2$, and $d(\mathbf{c}_i , \mathbf{c}_{i+1}) < T/2$.

If the test succeeds, then we can assume that the block belongs now to the background. For this reason, the k-th element of the test stack is pushed in the temporal stack, the the (k-1)-th and so on, and finally the test stack is emptied.

The results were executed versus the algorithms presented in [2] and [3]. 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.

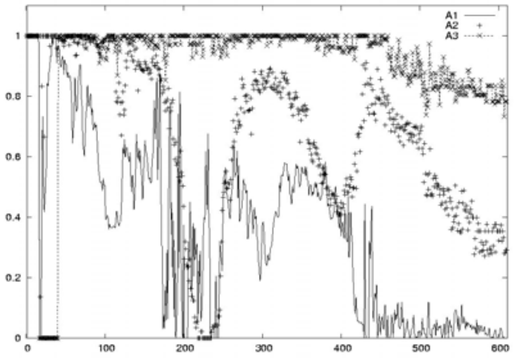


Figure 2. True positives for Walk 1

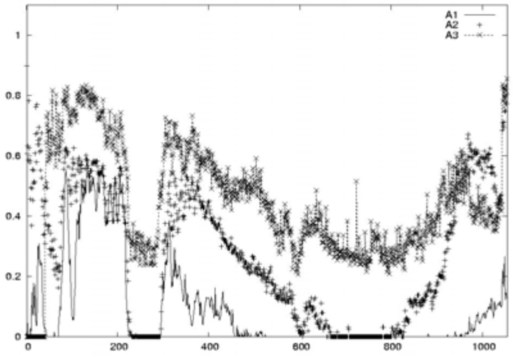


Figure 3. True positives for Walk 2

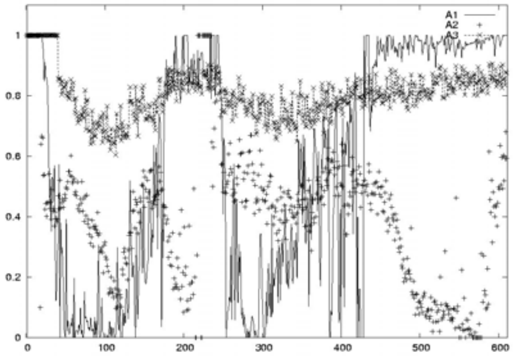


Figure 4. False positives for Walk 1

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.

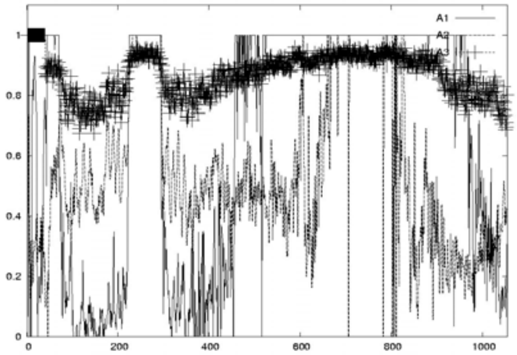


Figure 5. False positives for Walk 2

3.2 Human Identification Module

In the POLYMNIA specific environment it turned out that a combination of face recognition and visual feature recognition is the most promising solution to get reliable results of human recognition. Face recognition systems deliver good results, but they are not reliable against illumination and pose changes. Further challenges are low resolutions of face regions which we expect in the POLYMNIA application. To get better result even in these situations we additionally recognize the body (clothes) of the visitors.

The face recognition is done with the OpenCV library. The more reference images are captured from each person the better gets the recognition rate and the less influence has the accuracy of the segmentation of the face region.

The literature review of body recognition yields to a clear result that the Colour Structure descriptor (CSD) of MPEG7 [4] seams to be the most appropriate global visual feature for comparing body image regions. We also evaluate the CSD and we found out that the body recognition rate can be improved anymore by using a different distance function [5] as it is proposed in the MPEG-7 standard. An increase of the recognition rate of about 13% has been achieved when we use a combination (weighted sum) of the Euclid and the Jeffrey instead of the City Block distance function.

3.3 Human Tracking Module

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. 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.

3.3.1 Tracking Algorithm

There are many publications about real time tracking algorithms. A survey of methods used in visual surveillance can be seen in [6]. There are some publication from large research projects e.g. from the Visual Surveillance and Monitoring (VSAM) project [7] and the European Framework V project ADVISOR [8] about visual surveillance in metro stations. Often referenced systems are, among others, the real-time surveillance system W [9] and the PFinder system [10] for recovering 3D descriptions of people.

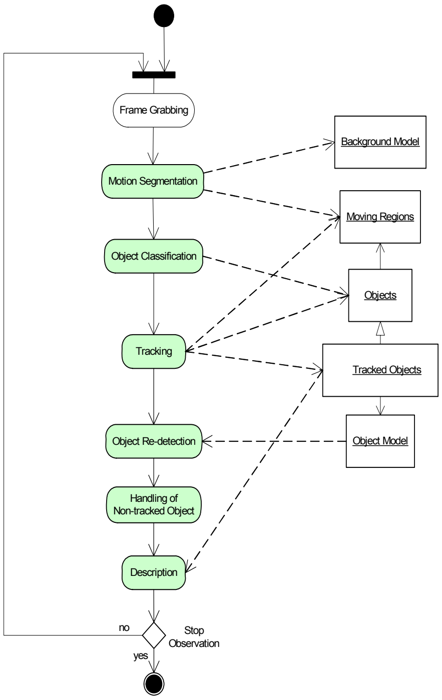


Figure 6. Processing steps for human motion tracking

The main processing steps of the implemented tracking algorithm are shown in Figure 6. 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 first processing step for each captured image is the segmentation of moving image regions. Common used approaches for motion segmentation are background subtraction and temporal differencing. For the tracking step we use the result of both of them. Additionally slow illumination changes can be detected if the two segmentation results are compared.

Once we have segmented the moving regions we have to detect which one are caused by moving peoples and where they are exactly located. An approved method is the detection of human heads by analysing the shape and the vertical projection of the pixel positions from the moving regions [7]. The potential head points are assumed at curvature maxima of the silhouette boundary and on the peaks of the vertical projection histogram

The movements of the persons are determined by tracking the detected head points and the moving regions itself. In principle it would be sufficient to track only the head points, but the information of the movement of the regions, when they split, and when they merge can be used to reduce wrong associations of head points from consecutive frames. In the tracking step the head and region location are predicted in the next frame. The predicted locations are matched and associated with the new detected ones.

For the prediction we use a combination of two methods, the Kalman filter and a simple first order motion model. The Kalman filter prediction works well if a person can not be detected for a longer time (e.g. due to an occlusion) and the first order motion model reacts very fast to abrupt motion changes. It could be that the association of a tracked head point to the new detected location is not unique. In this case we split the head point trajectory and calculate two predictions. If one of them performs better in the further tracking steps the other prediction will be rejected. For the prediction of the region location the median pixel position of the region is used.

The new and the predicted head positions are matched by calculating the Euclidian distance. Region matching is done by calculating the degree of overlap between the regions at the predicted location and the actual extracted regions.

Based on the matching results previous tracked head points and regions have to be assigned to the actual detected head points and moving regions. Thereby trajectories of head points and of region positions are created or updated. The used association method is Global Nearest Neighbor matching implemented by the auction algorithm [8].

If there is a movement in front of another foreground object no head points could be extracted from the background subtraction regions and therefore new points for the trajectories will not be found. In this case the head points extracted from the temporal differencing method are used (see Figure 7).

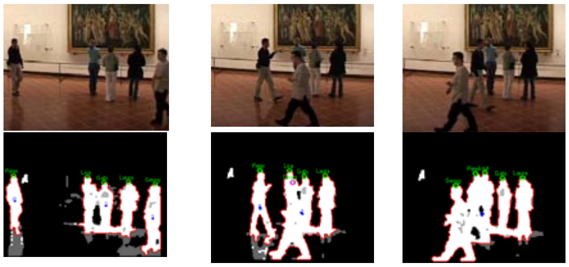


Figure 7. Example of tracking results

The extracted head points from temporal differencing are not as reliable as the head points received from background subtraction. To account this a higher threshold value for these point associations is used. This threshold value will be decreased after successful associations. If a person gets out or appears new in the camera view a handoff of the human location happens from identification or tracking systems whose camera views have an overlap with the current one. Unassigned objects are matched with previous occluded objects. If still no assignment can be made the new detected locations will be the start points of new human trajectories.

4. The Spatio-Temporal Module

The POLYMNIA system introduces a novel algorithm for the spatio-temporal content segmentation and representation of the acquired video sequences. In this way, events can be detected and subsequently used for data mining and retrieval procedures.

4.1 Segmentation of the video sequence

The first step of the proposed technique is to apply a video segmentation algorithm. In POLYMNIA framework, we apply a novel technique treating space-time as a 3D volume and analyzing the content in the extended domain by combining the information across all frames. Each pixel of the 3D space-time video sequence is mapped to a 7D feature point whose coordinates include *three color components*, *two motion angle components* and *two motion position components* (the latter two quantities will be defined in the following subsection). Then we apply a clustering procedure on these feature points utilizing the input from the real time POLYMNIA subsystem, the regions are classified into two categories: foreground objects and background objects.

4.1.1 Generation of the pixel feature vectors

Let us suppose a pixel $P_t = (x,y,t)$ at frame t and position (x,y) that belongs to a color segment. Let us also suppose that in frame $t + 1$ the pixel P_t of the pixel t has moved to $P_{t+1} = (x+u, y+v, t+1)$. Then, the corresponding motion vector of this movement is $(u,v,1)$. As we have stated in the previous subsection seven features are taken into account, (the *3 color* coordinates, the *two motion angle* coordinates and the *two motion position* components). The two motion angle coordinates describe the velocity vector of the pixel and are computed as,

$$a_x = 90^0 - \arctan u \quad a_y = 90^0 - \arctan v \qquad (1)$$

Similarly, the two *motion position* coordinates are computed as follows [11]:

$$D_x = (x - N/2) \sin(a_x) - (t - K/2) \cos(a_x) \qquad (2a)$$

$$D_y = (y - M/2) \sin(a_y) - (t - K/2) \cos(a_y) \qquad (2b)$$

where N, M refers to the frame dimension while K is the is the total number of frames of the sequence. We recall that (x,y) are the two co-ordinates of a pixel in the image frame.

4.2 Extraction of the regions

After computing the feature vectors of each pixel, our module proceeds to the video segmentation. Firstly the metadata generated by the POLYMNIA Real Time System (human detection, identification and tracking modules) are processed and the pixels belonging to human regions are determined.

For the computation of the rest regions we use a novel clustering algorithm on the video sequence pixels called the Adaptive Resonance Vector Quantization Algorithm (ARVQ). The ARVQ is a modification of the Learning Vector Quantization Algorithm (LVQ). The ARVQ algorithm, can be described as:

a. **(Phase 1).** The pixels of the a priori known regions (tracked human objects) are clustered together and the centroids of them are computed. The minimum resemblance (Euclidean distance) of a human region pixel with its region's centroid is computed,

$$\mathrm{RM} = \max_c \{ \max_{ci} \|\mathbf{x}_{ci} - \mathbf{w}_c\|\} \qquad (3)$$

where $\mathbf{x}_{ci}$ the i-th vector belonging to cluster c and $\mathbf{w}_c$ the cluster centroid or a heuristically defined proportion of it, can be used as resemblance threshold in phase 2.

b. **(Phase 2)**. A pixel series is made arraying the pixels of the video sequence randomly. Let us consider as $\mathbf{x}$ the feature vector of a pixel. Let us also denote as $\mathbf{w}_k$ the centroid of the k-th cluster. Then for each pixel in the pixel series

i. Find the cluster k for which

$$\|\mathbf{x} - \mathbf{w}_k\| = \min_i \|\mathbf{x} - \mathbf{w}_i\| \qquad (4)$$

ii. If cluster k is a human region cluster and pixel **x** belongs to it, proceed to (vi) step.

iii. If pixel **x** belongs to a human region represented by cluster k', different of k, then update the centroids of the clusters k and k' using the rule

$$\Delta\mathbf{w}_k = -\gamma_{error}(\mathbf{x} - \mathbf{w}_k) \qquad (5a)$$

$$\Delta\mathbf{w}_{k'} = \gamma_{error}(\mathbf{x} - \mathbf{w}_{k'}) \qquad (5b)$$

where γ_{error} is the error case learning rate. This way we perform a fine tuning of the cluster centroids so as to achieve a better representation of their characteristics through their centroids, aiming to avert the reoccurrence of such an error in the future.

iv. If cluster k is a human region cluster and pixel **x** does not belong to any human region, check whether there is a cluster k' fulfilling the following limitations:

k' must not represent a human region (6a)

$$\|\mathbf{x} - \mathbf{w}_{k'}\| = \min_i \|\mathbf{x} - \mathbf{w}_i\| \qquad (6b)$$

where i is any cluster not representing a human region

$$\|\mathbf{x} - \mathbf{w}_{k'}\| < \mathrm{RM} \qquad (6c)$$

Hence, we want to find out whether there is created, during the second phase, a cluster not representing a human region such that the Euclidean distance of its centroid from the pixel **x** is lower than the resemblance threshold RM.

If such a cluster exists, allocate pixel **x** to this cluster (k') and update its centroid using the rule

$$\Delta\mathbf{w}_{k'} = \gamma(\mathbf{x} - \mathbf{w}_{k'}) \qquad (7)$$

where γ is the allocation case learning rate.

Update also the human region cluster to which this pixel was erroneously allocated initially, by the rule (6a).

This fine–tuning of the human region's cluster is performed so as to avert the reoccurrence of a similar confusion between these two clusters in the future. If a cluster complying with (7) does not exist, create a new one and set **x** as its centroid.

v. If k is a non-human region cluster (thus generated in phase 2) and pixel x does not belong to human region, check whether the distance of pixel **x** from this cluster's centroid is lower that the resemblance threshold RM, i.e. holds

$$\|\mathbf{x} - \mathbf{w}_k\| < RM \qquad (8)$$

If (8) holds, allocate pixel x to cluster k and update its centroid by the rule

$$\Delta \mathbf{w}_k = \gamma\,(\mathbf{x} - \mathbf{w}_k) \qquad (9)$$

where γ is the allocation case learning rate.

Else, create a new cluster k' and set **x** as its centroid.

vi. If there are more pixels in the pixel series to process, take the next one (updating **x** with the new pixel's feature vector) and return to (i). Else finish.

The two learning rates, γ and $\gamma_{error,}$ must have a relationship of $\gamma >> \gamma_{error,}$ (e.g., 100/1). This way we avert a possible distortion of the centroids of the a priory known clusters.

Finally, the regions are classified into two categories: *background regions* and *foreground regions*, utilizing the input from the POLYMNIA Real Time system.

4.3 Graph-Based Spatio-Temporal Content Representation

The POLYMNIA system introduces a novel algorithm for the *Graph-Based* representation of the spatio-temporal relations between the extracted objects (regions) of the video sequence. The algorithm proposed comprises the generation of an undirected, attributed graph G(V,E) where the vertices represent the regions the video sequence consists of and the edges represent the spatio-temporal relations between them. Each graph vertex is integrated with a feature vector that describes the mean features of the region represented (it is the feature vector of the region's centroid).

The edges of the graph denote the pairs of regions that are spatio-temporally related. They connect a) pairs of spatially adjacent background objects in a series of frames, b) pairs of foreground – background objects, where the foreground object is spatially adjacent to the background one in a series of frames, c) pairs of foreground objects, both appearing in some frames, that either have a boundary between them or with the same background object (thus are also strongly spatio-temporally related).

4.4 Performance Evaluation

To evaluate the proposed representation scheme, we applied a series of tests based on an inexact graph matching model of the video retrieval problem. The problem is formulated as follows.

Given two undirected graphs with attributed vertices, the model graph $G_1(V_1,E_1)$ (representing the user's input) and the data graph $G_2(V_2,E_2)$ (representing a data piece inside the multimedia database being used), and a vertex resemblance function r: $\{V_1 \times V_2\} \rightarrow \Re$, check whether exists a function

$$f: V_1 \rightarrow V_2 \text{ such that} \qquad (10a)$$

$$f \text{ is 1-1} \qquad (10b)$$

$$r(u,f(u)) = \max\{r(u,w)\} \quad \forall\, u \in V_1,\, w \in V_2 \qquad (10c)$$

$$(f(u),f(v)) \text{ exists iff } (u,v) \text{ exists, where } u,v \in V_1 \qquad (10d)$$

As can be noticed, the problem formulation here deviates from the typical definition of the inexact graph matching problem, as we require the vertices of the model graph to be mapped to vertices of the data graph by a 1-1 function. To address problem (10) we introduce a novel algorithm comprising the following steps:

1. For each pair of vertices (u,w), $u \in V_2$, $w \in V_{1,}$ compute their resemblance $r(u,w)$.
2. For each model vertex w find the data vertex v that resembles to it most, i.e. $r(v,w) = \max\{r(u,w)\} \;\; \forall\, u \in V_{2.}$ Let us denote v as $f(w)$.
3. If a collision occurs, i.e. more than one model vertices are mapped to the same data vertex apply the following collision resolution algorithm:
   a. Let us suppose that the model vertices $a_1,a_2,\ldots,a_n$ are mapped the same data vertex b.
   b. Split the solution to a set of n different alternative solutions, where in the i-th solution the model vertex a_i is mapped to the data vertex b and the other model vertices are mapped to the second more similar to them data vertices.
   c. Apply collision resolution algorithm recursively to each one of the n sub-solutions until no collision exists.

Typically, the collision probability must be very low, since in this algorithm we map the model vertices to the data vertices, thus the model regions to the oversegmented regions and hence, the only case a collision could happen is the one in which the used resemblance metric in conjunction with the selected feature space coefficients cannot describe the region mappings successfully but, on the contrary, they cause errors.

The new approach yields significant results since it allows the detection of a) events, such as human congestion, absence of humans, etc, b) humans' actions such as interest in the same venue object, c) temporal and /or spatial relations of humans' actions, such as the duration that a particular human looks at a painting. All these "high level" concepts can be efficiently described using the proposed spatio-temporal graph representation.

Such a representation significantly enhances the image retrieval results since it allows for the detection and mining of "events". In addition, it assists the POLYMNIA platform in summarizing and non-linearly organizing the content information by discarding information that it is not relevant to the user's needs and preferences.

5. Real-time Controller

The real-time controller software has the role of orchestrating the data flow between modules, and initiating the control flow involved in coordinating module activities. We consider an event driven control flow strategy due to the pre-determined nature of our control flows. Performance must be as near to real-time as possible, capable of processing up to a possible 50 messages (frames) a second from our camera modules.

The controller framework provides a web service that all POLYMNIA modules can use to receive instructions about what to do at any given time. The controller service hosts a set of state machines, one for each module, that are capable of processing incoming events and generating control instructions for other modules; the control flow is encoded into these state machines. The controller software instructs modules to start specific processing activities and waits to receive regular module progress reports. Module progress reports are fed into the state machines as events and used to change state and generate new instructions for other modules to process.

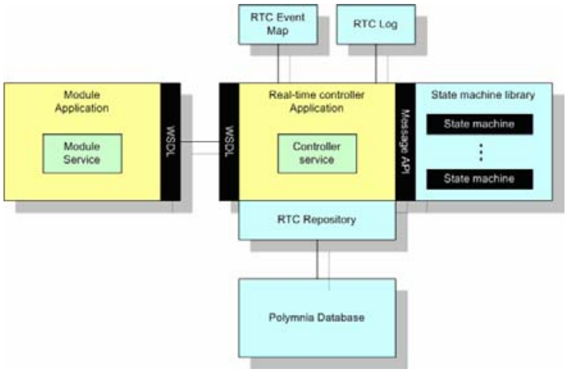


Figure 8. Real-time controller architecture

5.1 Controller framework

The real-time controller is based on a Windows C++ GSOAP [12] web services framework, upon which both the real-time controller and each module will expose a web service. The real-time controller is responsible for starting activities and processing the reports generated as part of these activities. Activity reports are fed to each module's state machine for processing, which results in a state transition and potentially some more activities to be started. The real-time controller architecture is shown in Figure 8. Each module provides a C++ state machine DLL that is capable of processing incoming events and generating control instructions for other modules.

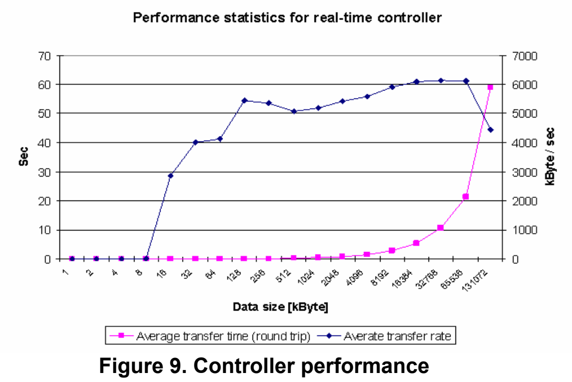


Figure 9. Controller performance

5.2 Computational Performance

Some performance statistics are shown in Figure 9, using the below hardware. Real controller usage will not execute activities sequentially, so will degrade depending on how many real-time controller invocations are being processed at the same time. The drop in performance at 128 Mbytes was due to the computer running near the limit of its memory, and needing to start caching etc.

5.3 Application example: switching live streaming sources to track visitors remotely

An example of application of the Real-time Controller in POLYMNIA is the control of the camera switching mechanism used to stream live footage from real visitors at venues.

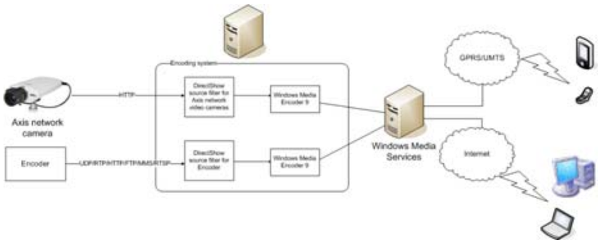


Figure 10. Switching sources mechanism

For the real time transmission and delivery of media content produced by the POLYMNIA infrastructure, adaptive streaming technology have been provided based on the Windows Media platform, allowing a broad portfolio of target devices ranging from PC's and PDA's to mobile phones (see Figure 10)

The tracking event from the Human Content Tracking subsystem is progressed to the Real Time Controller, which invokes a *RegisterChannel* Web Service method in the Adaptive Streaming module. This event results in the creation of a streaming URL that points to a *playlist* containing the address of the instance of the Windows Media Encoder that takes care of capturing images from that tracking camera.

If the visitor moves to a different location in the venue and is tracked by a different camera, this situation is notified to the Adaptive Streaming module by means of a *ChangeChannel* Web Service method, resulting in a switching of the input sources in the *playlist*. It is important to notice that this switching operation is performed without intervention of the e-visitor, who keeps on watching the visitor at the same publishing point URL.

6. Automated Media Production

In film making, there are some pretty basic rules [13] 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 POLYMNIA 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 film edits are predictable and as such possible to encode and automate. Film edit rules are encoded as film templates, with specific templates made for specific attractions at the theme park / museum. In POLYMNIA we have a professional film director within the project who will help us create each template. The media production service is the component in the POLYMNIA system that executes film templates.

6.1. Media Production Service

The POLYMNIA media production service is a GSOAP web service that takes a MPEG-7 [14] document containing all the shots from a visitors day at the theme park / museum and generates an 'edit decision list' for it based on some pre-defined directorial rules. The edit decision list is a film script which describes how the visitor's video clips should be used to make the film.

The production service uses a three phase rule engine to execute each film template. The MPEG-7 edit decision list 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 video tape) (Figure 11).

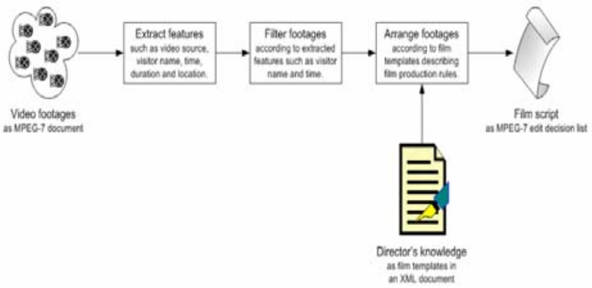

Extract features
such as video source, visitor name, time, duration and location.
Filter footages
according to extracted features such as visitor name and time.
Arrange footages
according to film templates describing film production rules.
Video footages
as MPEG-7 document
Film script
as MPEG-7 edit decision list
Director's knowledge
as film templates in an XML document

Figure 11. From Video footages to the edit decision list

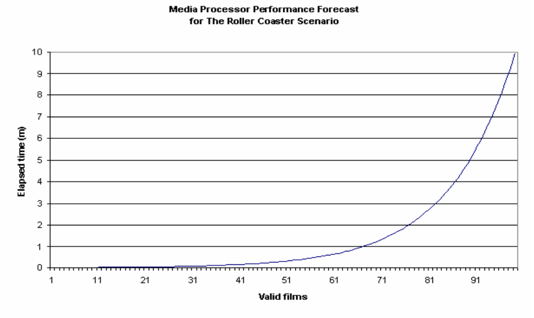


Figure 12. Rule engine performance

The POLYMNIA DVD production software takes a visitor's film footage and, according to the edit decision list, generates an avs script that specifies the frames composing the DVD. The avs script is then passed to the frame server (AviSynth) that extracts the selected frames. They are then given to the encoder module (QuEnc) for the creation of the MPEG-2 video and audio file. Finally, DVDAuthor is the program used to generate the DVD souvenir that the visitor can then purchase and take home. The same edit decision list is also used to select some pictures from the visitor's videos in order to produce e-souvenirs like e-photos, e-cards and e-albums.

The performance of this rule engine is important given the production time constraint of 10 minutes. Figure 12 shows how this rule-based approach scales up well with realistic numbers of theme park / museum attractions.

7. References

[1] K. Toyama, J. Krumm, B. Brummit and B. Mayers, "Wallflower: Principles and Practice of Background Maintenance," *ICCV*, Corfu, Greece, p. 255-261, September 1999.

[2] L.L.W. Huang, I.Y.H. Gu and Q. Tuan, "Foreground Object Detection from Videos Containing Complex Background", *in Proc. ACM Multimedia conf.*, ACM, Berkeley, CA, USA, November 2-8 2003.

[3] D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R. B. Fisher, J. Santos-Victor and J. L. Crowley, "Comparison of target detection algorithms using adaptive background models", *VisLab-TR 13/2005, Proc. 2nd Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, (VS-PETS)*, Beijing, October 2005.

[4] D. S. Messing, P. van Beek, J.H. Errico, "The MPEG-7 colour structure descriptor: image description using colour and local spatial information", *IEE-ICIP* Vol. 1., pages 670-673, 2001, Thessaloniki, Greece.

[5] Y. Rubner, J. Puzicha, C. Tomasi, and J. Buhmann, Empirical Evaluation of Dissimilarity Measures for Color and Texture,"*Computer Vision and Image Understanding* Vol. 84, pages 25-43, 2001.

[6] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors", *IEEE Transaction on Systems, Man and Cybernetics – Part C:* Applications and Reviews", vol. 34, nr. 3, pages 334-352, August 2004.

[7] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggis, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A System for Video Surveillance and Monitoring", *Carnegie Mellon Univ., Tech. Rep., CMU-RI-TR-00-12*, 2000.

[8] N. T. Siebel, St. J. Maybank, "The ADVISOR Visual Survaillance System", *Proceedings of the ECCV Workshop on Applications of Computer Vision*, Prague, May 2004.

[9] I. Haritaoglu, D. Harwood, L. S. Davis, "W^4: Real-Time Surveillance of People and Their Activities", *IEEE Trans on PAMI* , vol. 22, No. 8, pages 809-830, August 2000.

[10] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body", *IEEE Trans. on PAMI*, 19(7), July 1997.

[11] D. DeMenthon, "Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis," *Proc. Statistical Methods in Video Processing Workshop (SMVP)*, June 2002.

[12] GSoap web page http://www.cs.fsu.edu /~engelen/soap.html

[13] Mascelli, J., "The Five C's of Cinematography: Motion Picture Filming Techniques", *Silman-James Press*, 1998.

[14] The Moving Picture Experts Group (MPEG) a working group of ISO/IEC, "MPEG-7: The standard for multimedia for the fixed and mobile web"; http://www.chiariglione.org/mpeg/