Entity appearance model generation for multimedia events in surveillance videos

DOI:
10.1109/IS.2010.5548368
10.1109/is.2010.5548368

Entity appearance model generation for multimedia events in surveillance videos

Fahad Anwar, Ilias Petrounias, Tim Morris
University of Manchester
Manchester, UK
Fahad.Anwar@manchester.ac.uk
Ilias.Petrounias@manchester.ac.uk
Tim.Morris@manchester.ac.uk

Vassilis Kodogiannis
University of Westminister
London, UK
V.Kodogiannis@westminister.ac.uk

Abstract— **Traditionally, surveillance systems only focus on a small number of entities (such as humans, entrance and exit areas, etc.) and appearance models of these entities are uploaded manually in the system. However, as the end users are becoming more aware of the vision based technologies, there is ever growing demand for advanced surveillance systems which can detect complex abnormal events on different aspects of operational activities and can also provide intelligence to improve their operational management process. To achieve this goal, we proposed the event mining framework which explores the relationship between entity feature-sets and associated text strings to generate appearance models of all the entities automatically and can update them dynamically.**

Keywords-component; Multimedia mining, Object appearance modelling, Event modelling and detection

I. INTRODUCTION

Multimedia events can be defined as interactions between different entities over time and space. Since entities are one of the key elements in a multimedia event structure, it is important that any proposed event modelling/detection framework should have the capability to store appearance models of all the required entities in the given environment. Many event modelling frameworks have been proposed in the literature [1-3]; they mainly assume a limited number of entities in specific domain which can be uploaded manually in the system. However in certain environments, surveillances videos can contain a very large number of entities; one prime example is the retail store environment in which there can be thousands of different items (milk, orange juice, shampoo, fruits, vegetables etc.). Manual uploading of such large number of entities' appearance models is practically infeasible; hence, the system needs an optimised process to store all required entities' appearance models with minimum human effort. The problem of storing appearance models of entities in such an environment is further complicated due to the nature of surveillance videos, where entities may have different appearances in different video shots. This is because object appearance can differ due to changes in environmental conditions, such as different lighting condition, shadows, change in field of view (due to different entities overlapping each other), object rotation, camera angle etc). Therefore, it is important that the proposed framework should handle the above mentioned variations in field of view and environment conditions. In light of the above mentioned challenges it is important that any proposed framework should have the ability to upload appearance models of entities automatically and also update them dynamically. To achieve this goal, we utilise the different multimedia streams (text and video) in a surveillance system environment where CCTV video can be linked to text data (for example CCTV of till scanning process and ePOS data can be linked together). The main notion behind the proposed framework is to first define a few simple events through an event modelling module such as "Object crosses the entry line and then received text string (object name)" and then during event detection process store the entity features and corresponding text string [4]. Once data has been collected, our proposed mining framework will explore the relationship between different feature sets and entity text strings to generate appearance models of all the entities automatically (Figure. 1)

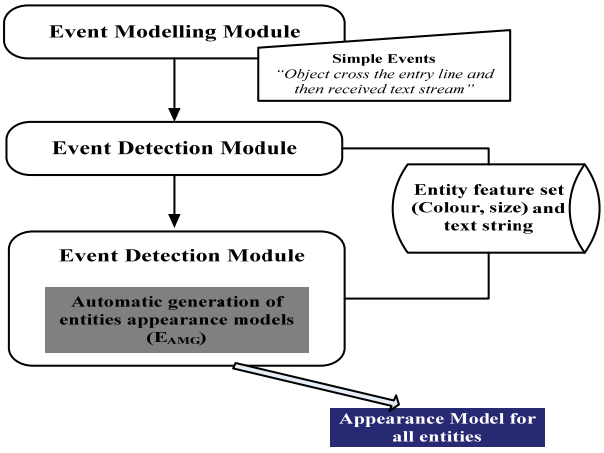

Figure 1. Proposed entity appearance model generation process

978-1-4244-5164-7/10/$26.00 ©2010 IEEE

II. II. PROBLEM DEFINITION FRAMEWORK

The problem of generating appearance models of entities from already detected simple events can be defined as: We have a database of already detected simple events "D_{EVENTS}" (such as "Object crosses the entry line and then received text string") spanning over the time domain "T", each record is a tuple of <E_{ID} (Event ID), E_T (Event Time), D_{OBJECT} (Detected Object), T_{STRING} (Text String)>; where each D_{OBJECT} is a set of different object features (f_i) such as size in pixels, object colour histogram, etc D_{OBJECT} = <f_1, f_2……. f_i…….f_n>. The problem to investigate is to generate appearance models of all the entities by discovering the optimal relationship between the Text String (T_{STRING}) and object features.

In order to provide a flexible and comprehensive problem definition we extend the problem definition framework by introducing three user defined parameters namely Time Period (T_P), Minimum Frequency Threshold (MIN_F_{TH}) and Temporal Boundaries (T_B).

It is quite possible that the user is very much interested in generating appearance models of entities during only specific time periods rather than in the whole time spectrum. The user defined parameter T_P can be use to reflect this concept; T_P represents the time period during which the appearance model of different entities needs to be generated. As discussed earlier, in surveillance videos entities may have different appearances in different video shots; this is because object appearance can differ due to object rotation and/or due to different entities overlapping each other. Therefore, it is important that the proposed system should be able generate all the frequent appearance models of entities which can be presented in a field of view. To confront this challenge, a user defined parameter MIN_F_{TH} is introduced here;

MIN_F_{TH} represents the minimum representation of object features (in percentage) before it can be considered as a new appearance model of the same entity. In surveillance videos, entities can have different appearances during different time intervals (e.g. morning, afternoon, evening and night); this is because of different lighting conditions during these time intervals. Hence, if the variations in lighting conditions are significant during different time interval; it is be important to store different appearance models representing different time intervals for the same entity. T_B is introduced here to represent user defined time intervals (T_{ITVL}). Each T_{ITVL} consists of two entries, Interval start time ($ITVL_{ST}$) and Interval end time ($ITVL_{ET}$); where $ITVL_{ST}$ and $ITVL_{ET}$ are a set of temporal granularities of hours, minutes and seconds.

$$T_B = < T_{ITVL}^{1}, T_{ITVL}^{2} \ldots T_{ITVL}^{i} \ldots\ldots T_{ITVL}^{n} >$$
$$T_{ITVL} = (ITVL_{ST}, ITVL_{ET})$$
$$ITVL_{ST}, ITVL_{ET} = <hh,mm,ss>$$

A. *Extended problem definition framework*

We have a database of already detected simple events "D_{EVENTS}" spanning over the time domain "T", each record is a tuple of <E_{ID} (Event ID), E_T (Event Time), D_{OBJECT} (Detected Object), T_{STRING} (Text String)>; where each D_{OBJECT} is a set of different object features (f_i) such as size in pixels, object colour histogram etc, D_{OBJECT} = <f_1, f_2……. f_i……, f_n> along with user defined parameters of Time Period (T_P), Minimum Frequency Threshold (MIN_F_{TH}) and Temporal Boundaries (T_B). The problem to investigate is to generate all appearance models of each entity by discovering the optimal relationship between the Text String (T_{STRING}) and object features.

III. APPEARANCE MODEL GENERATION PROCESS

The proposed model for generating appearance models of real world entities consists of two main phases, that is data filtering and discovery of optimal values for each feature (D_{FOV}) of a given entity. In this paper, we mainly focus on two features to be used for appearance model generation process that is Size and Colour; however, the proposed model is flexible and can include other features as well. This flexibility is based upon the main concept of the proposed model: that is to discover optimal values of each feature separately and then calculated optimal values of all features combined together to generate appearance models of each entity/object (Figure. 2).

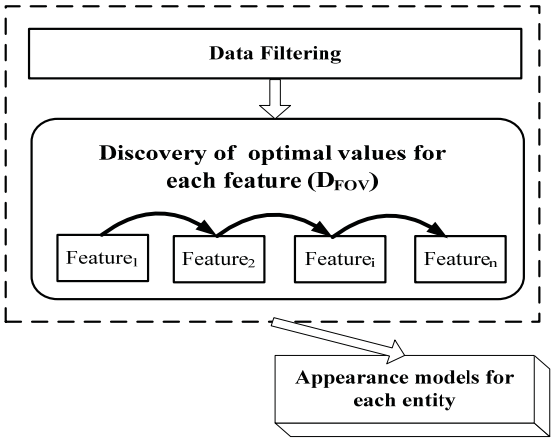

Figure 2. Automatic appearance model generation process

A. *Data Filtering*

In the data filtering phase the input data (D_{EVENTS}) is filtered in two steps; firstly only data which falls within the boundaries of user defined parameter of T_P is filtered out from the database. For example if T_P is defined as 1st Jan, 2009 to 30th March, 2009 then all D_{EVENTS} during this time period will be filtered out for on-word data processing. Secondly, the already filtered data is segmented into different data-sets according to the user-defined parameter of T_B. For example if

T_B is defined as T_B=((07:00:00-16:00:00), (16:00:01-20:00:00), (20:00:01-06:59:59)) then the data is segmented into three sets, each set contains D_{EVENTS} which fall within respective ITVLs of T_B (Figure 3).

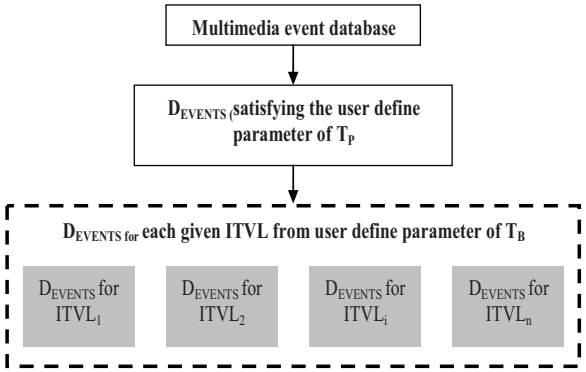

Figure 3. Data filtering process

B. Discovery of optimal values for each feature (D_{FOV})

Once D_{EVENTS} are segmented according to the user defined parameters of T_P and T_B (during the data filtering stage); the process of discovering optimal values for each feature proceeds during each segmented ITVLs. In each ITVL the process of D_{FOV} is carried out on each entity one after the other (Figure 4). In following sections we will discuss the process of finding optimal values for size and colour features for each given entity/object.

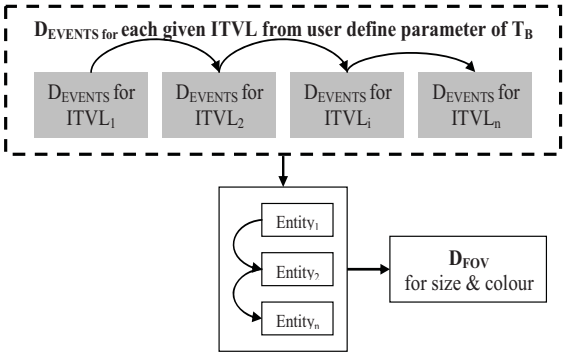

Figure 4. D_{FOV} for entity during different segmented ITVLs

1) Optimal value for entity size feature

For generating optimal values for entity size feature; the size feature values of the current entity are first extracted (Figure. 5); then a clustering process is executed on the extracted values (sizes in pixels). After that, clusters which have representation ≥ MIN_F_{TH} are extracted as a set of optimal values; for example if MIN_F_{TH} is set to 50% then all the clusters having representation ≥ 50% are extracted as different optimal values of size feature. If there is no cluster which has representation ≥ MIN_F_{TH}, then the densest part of the interval range is used as the optimal value of size feature for that specific entity. The extracted optimal values are then ranked according to their representation ratio against total number of values.

Object	Size (in pixels)
Coke Bottle	254
Coke Bottle	301
Milk Pack	542
Coke Bottle	198
Butter	142
White Sugar	169
Milk Pack	498
.....	.....
$Object_n$	$Size_n$

Object	Size (in pixels)
Coke Bottle	254
Coke Bottle	301
Coke Bottle	198

Figure 5. Extracted size feature for specific entity

In Figure. 6 and Figure. 7, clustering results of two example entities (coke bottle and white sugar pack) is presented. For example if the user-defined parameter MIN_F_{TH} is set to 30%, then for the "coke bottle" entity there are two clusters whose representation is greater or equal to MIN_F_{TH} that is cluster interval from (300-350) and (400-450); and their ranking in percentages will be 56% and 34% respectively. For the entity "white sugar pack" there is only one cluster (interval range 500-550) which has representation greater or equal to MIN_F_{TH} with 86% ranking/support value.

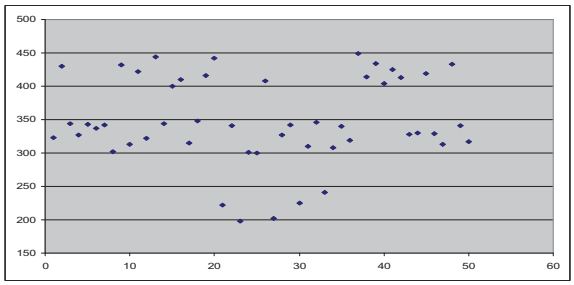

Figure 6. Clustering results on coke bottle entities

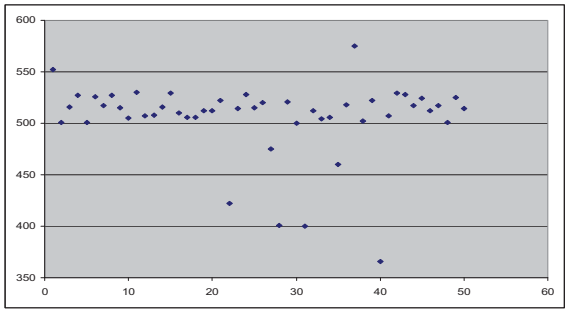

Figure 7. Clustering results on white sugar pack entities

2) Optimal value for entity colour feature

As discussed earlier, during the simple event detection process we store different features for each D_{OBJECT}, which includes 32 bin colour histogram in HSV colour space to represent the colour of that specific entity/object. For the sake of simplicity, we discuss the Hue element of the colour histogram here. To find the optimal histogram values for an entity, we first normalise the histogram values so that the values are between 0 and 1 and then categorise "bin" values into 10 different categories, after categorisation each bin is assigned an appropriate category number.

During the process of discovering the optimal colour histogram for a specific entity, the frequency of each bin's category is calculated. As we can see in Table 1, there are a total of 50 occurrences of the entity/object "White Sugar Pack" and for "bin_1" 60% representation is for category 04. Suppose MIN_F_{TH} is set to 30%, then category 04 representations for bin_1 (60%) is $\geq MIN\_F_{TH}$; hence, it is considered as optimal value for "bin_1" and for bin_1 category 04 is assigned to primary optimal colour histogram for this entity. For bin_2 there are two categories which have frequency $\geq MIN\_F_{TH}$; hence, both category 02 and 05 are consider as optimal values.

Now for bin_2, category 02 is assigned to primary optimal colour histogram and since there is a second optimal value of bin_2, the second optimal colour histogram is created with bin_1 value as category 04 and bin_2 value as category 05 (Table 2) and so on. Once all the possible optimal colour histograms are created, they are then ranked according to their representation ratio; since each bin has its own representation ratio, therefore, the average representation ratio of all the bins are used to rank each optimal colour histograms. For example in Table. 2 the representation ratios of the primary and second optimal colour histograms are 64% and 60% respectively.

Bin	Category	Frequency
1	4	30
1	6	10
1	7	07
1	1	03
2	2	24
2	5	18
2	1	08
…..	….	…
32	1	39
32	2	09
32	5	02

Table 1. Category frequency for each "bin"

Primary optimal colour histograms			**Second optimal colour histograms**		
bin	category	rep	bin	category	rep.
1	4	60%	1	4	60%
2	2	48%	2	5	36%
3	1	84%	3	1	84%

Table 2 Multiple optimal colour histograms

C. *Appearance Model generation*

Once optimal values of all the features have been discovered, the process of generating appearance models for entity is straight forward. Appearance models are basically generated by creating all possible combinations of already discovered optimal values of different features. For example in Table 3, there is one optimal value for size feature and two optimal histograms values for colour feature; therefore, by combining these three optimal feature values, two appearance models are generated for "White Sugar Pack" entity. The ranking of these appearance models are based upon average ratio of representation of all the features; moreover, since these appearance models are generated from $ITVL_1$ data-set, temporal information is also attached with the generated appearance models (Figure. 8).

$Feature_1$ Size	**$Feature_2$ Colour (Histogram)**
500-550 (86%)	1) First Colour Histogram (64%) 2) Second Colour Histogram (60%)

Table 3 Optimal feature values for entity

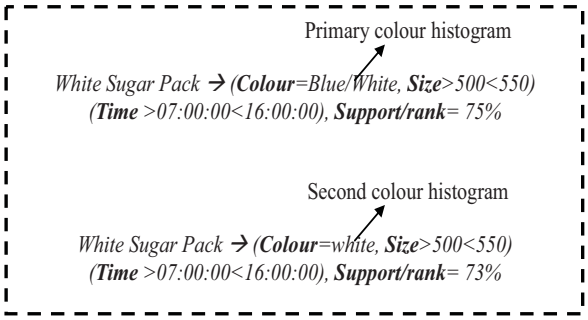

Figure 8. Multiple appearance models for entity

IV. VALIDATION FOR ENTITY APPEARANCE MODEL GENERATION TECHNIQUE

To validate proposed entity appearance model generation concept, we implement the algorithm in the Matlab programming environment and conduct the experiments by using "Columbia University Image Library (COIL-100)" object database [5]. The database consists of 100 images of different objects and for each object 72 different views covering the object from different angles are available. In our experiment we use these 72 views of the object to generate an

appearance model.

After segmenting the object from the background we then extracted the size (in pixels) and colour (in the HSV colour space) features of the object in each view and applied our proposed appearance generation technique. Once the appearance model of a specific object is generated we evaluate its strength in two aspects (specificity and sensitivity). In our first experiment we evaluate the specificity strength of the object appearance model (OAM); that is to evaluate the OAM strength to differentiate the actual object from other objects. For this purpose we randomly select one view of each object and use the OAM to retrieve the matching objects and calculate the probability that a non-relevant object is retrieved by the object matching process. To evaluate OAM sensitivity (that is to retrieve the same object from different views); we use the OAM on all 72 different views of the same object and calculate the probability that a relevant object is retrieved by the object matching process. The process of generating the appearance model of an object and then evaluating its specificity and sensitivity strengths with the above mentioned experiments was performed on all 100 objects (using 0.7 as threshold for object matching). The results presented in Figure. 9 & 10 show the overall high specificity and sensitivity of the OMA; however OMA fails to differentiate the objects where different objects' size and colour features are similar. In more complex environments other object features (such as shape or texture) can be used to make the appearance model more robust. In Figure. 10 (object 52, 68) we can see that if there is significant variation among different views of the object, the proposed method fails to generate an effective OAM, hence is not able to retrieve a high percentage of objects viewable from different angles. However, in a real world environment the appearance model can be further strengthened as the number of object views increases over time and contributes to the development of an effective OAM.

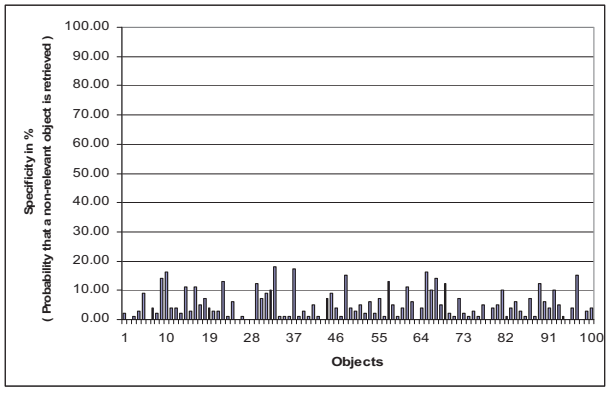

Figure 9. Probability that a non-relevant object is retrieved

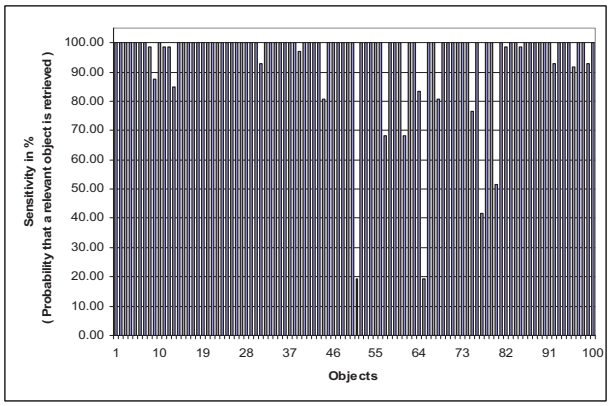

Figure 10. Probability that a relevant object is retrieved

V. CONCLUSION

In this paper proposed a framework to generate appearance models by utilising the multimedia streams of text and video. The main notion behind the process is to first define few simple events such as "Object crosses the entry line and then received text string" and then during the event detection process, store the entity features and corresponding text string. Once data has been collected, our proposed mining framework explores the relationship between different feature sets and entity text strings to generate appearance models of all the entities automatically. To validate the proposed entity appearance model generation concept, we implemented the proposed algorithm and conducted experiments by using the "Columbia University Image Library (COIL-100)" object database [5].

REFERENCES

[1] S. Hongeng and R. Nevatia, "Multi-agent event recognition," in *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001)*. vol. 2 Vancouver, Canada, 2001, pp. 84-91.

[2] A. Hakeem, Y. Sheikh, and M. Shah, "CASE^E: A Hierarchical Event Representation for the Analysis of Videos," in *The Nineteenth National Conference on Artificial Intelligence (AAAI)* San Jose, USA, 2004, pp. 263-268.

[3] P. Natarajan and R. Nevatia, "EDF: A framework for Semantic Annotation of Video," in *Proceedings of the Tenth IEEE International Conference on Computer Vision Workshops*: IEEE Computer Society, 2005.

[4] H. Zhang, S. Y. Tan, S. W. Smoliar, and G. Yihong, "Automatic Parsing and Indexing of News Video," *Multimedia Systems,* vol. 2, pp. 256-266, 1995.

[5] X. Zhu, X. Wu, A. K. Elmagarmid, Z. Feng, and L. Wu, "Video Data Mining: Semantic Indexing and Event Detection from the Association Perspective - Appendices," in *IEEE Transactions on Knowledge and Data Engineering*, 2005, pp. 665-667.