01-MD

Shu-Ching Chen, Mei-Ling Shyu, Chengcui Zhang, and Jeff Strickrott, "Multimedia Data Mining for Traffic Video Sequences," the Second International Workshop on Multimedia Data Mining (MDM/KDD'2001), in conjunction with the Seventh ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 78-85, August 26, 2001, San Francisco, CA, USA.

Multimedia Data Mining for Traffic Video Sequences

Shu-Ching Chen[1], Mei-Ling Shyu[2], Chengcui Zhang[1], Jeff Strickrott[1]
[1]Distributed Multimedia Information System Laboratory
School of Computer Science, Florida International University, Miami, FL 33199
[2]Department of Electrical and Computer Engineering, University of Miami,
Coral Gables, FL 33124

ABSTRACT

In this paper, a multimedia data mining framework for discovering important but previously unknown knowledge such as vehicle identification, traffic flow, and the spatio-temporal relations of the vehicles at the intersections from traffic video sequences is proposed. The proposed multimedia data mining framework analyzes the traffic video sequences by using background subtraction, image/video segmentation, object tracking, and modeling with multimedia augmented transition network (MATN) model and multimedia input strings, in the domain of traffic monitoring over an intersection. The spatio-temporal relationships of the vehicle objects in each frame are discovered and accurately captured and modeled. Such an additional level of sophistication enabled by the proposed multimedia data-mining framework in terms of spatio-temporal tracking generates a capability for automation. This capability alone can significantly influence and enhance current data processing and implementation strategies for several problems vis-à-vis traffic operations. A real-life traffic video sequence is used to illustrate the effectiveness of the proposed multimedia data mining framework.

KEY WORDS: Multimedia data mining, spatio-temporal relationships, multimedia augmented transition network (MATN), object tracking.

1. INTRODUCTION

As computers have become more powerful, their role in everyday life has become more pervasive. Recent efforts [8,23,26] have begun to shift the traditional focus from user centric applications (i.e., word processors, browsers, etc.) to that of a ubiquitous tool that facilitates everyday activities. Projects like EasyLiving [23,26] and HAL [8] aim to develop smart spaces that can monitor, predict, and assist the activities of its occupants. These efforts at developing smart environments are not confined to homes or offices, but extend to that of the world around us. Municipalities [1,24] are installing video camera systems to monitor and extract traffic control information from their highways in real time. Issues associated with extracting traffic movement and recognizing accident information from real time video sequences are discussed in [10,11,20,21,22]. Two common themes exist in these works. First, the video information must be segmented and turned into objects. Second, the behavior of those objects is monitored (they are tracked) for immediate decision making purposes. What is missing in these efforts is to model and index the data for on-line analysis, storage or later pattern mining.

The analysis and mining of traffic video sequences to discover information, such as vehicle identification, traffic flow, and the spatio-temporal relations of the vehicles at intersections, provides an economic approach for daily traffic operations. In order to identify and track the temporal and relative spatial positions of vehicle objects in video sequences, it is necessary to have object-based representation of video data. For this purpose, attention has been devoted to segmenting video frames into regions such that each region, or a group of regions, corresponds to an object that is meaningful to human viewers [9,13,14]. While most of the previous works are based on low-level global features, such as color histogram and texture, our video segmentation method focuses on obtaining object level segmentation; obtaining objects in each frame and their traces across the frames. In [3-5] we have addressed the issues of unsupervised image segmentation; object modeling with multimedia input strings to capture the spatial–temporal behavior of the object, and the application of these techniques to the domain of traffic monitoring.

Similar approaches to our segmentation technique are discussed in [12,25]. In [25] the authors consider a Bayesian technique to segment images based on feature distributions. The histogram of features around a pixel neighborhood is considered as an estimate of the conditional probability distribution P(c|Y) versus the parametric equation in our approach (see Section 2.2). This technique models the texture in a neighborhood. DeMenthon et al. [12] utilize a Hidden Markov Model approach for low level image segmentation. Associated with each pixel are an observation vector and a hidden state. The observation vector is the set of parameters (of interest) associated with each pixel, such as color, or the average intensity of the image region centered on that

pixel. The hidden state is a label for that pixel. Computational time is $O(ns^3)$, where n is the number of pixels in the image and s is the number of states (regions) to segment the image. Segmentation in an image can also be modeled as a pixel-labeling problem, in which we must decide from which of M number of classes the pixel belongs. The membership in each class is formulated as a Bayesian conditional probability decision, where class membership is estimated from the intensity distributions of neighboring pixels. When the image segmentation problem is considered for a fixed camera domain, a classic technique to resolve the foreground objects is background subtraction [16]. This involves the creation of a background model that is subtracted from the input image to create a difference image. The new difference image only contains objects not in the background or new features that have not yet been incorporated into the background.

Various approaches to background subtraction and modeling techniques have been discussed in the literature [11,17,19,28], ranging from modeling the intensity variations of a pixel via a mixture of Gaussian distributions to simple differencing of successive images. In [29] the authors provide some simple guidelines and evaluation of the various techniques for background modeling. We are in the beginning phases of evaluating the performance benefits of background subtraction methods for the various domains of our image segmentation applications. To that aim we have evaluated the effectiveness of simple image averaging techniques over stationary (non-changing) portions of the image data set.

In this paper, a multimedia data mining framework for traffic video sequences is proposed. The proposed framework considers image/video segmentation with initial background subtraction, object tracking, and modeling with multimedia augmented transition network (MATN) model and multimedia input strings [2,7], in the domain of traffic monitoring over an intersection. The multimedia input strings are used to capture the spatio-temporal relationships of vehicle objects thereafter. The video segmentation method mentioned here is unsupervised. Another advantage is that it uses the segmentation result of the previous video frame to speed up the segmentation process of the current video frame. Experiments were conducted to illustrate the effectiveness of the proposed framework using a real-life traffic video sequence. The traffic video sequence was captured with a *Sony Handycam CCD TR64* and digitized with an inexpensive *Brooktree Bt848* based capture card on a Windows NT 2000 Celeron based platform. The original images are 640x480, 24 bit color and the video sequence was sampled at 5 frames per second.

The organization of this paper is as follows. In next section, the knowledge discovery process that includes background subtraction, the unsupervised segmentation algorithm, object tracking techniques, MATN model, and multimedia input strings are introduced. Experiment results and analysis of the proposed multimedia data mining framework are discussed in Section 3. Along with the discussion, an example real-life traffic video sequence is used. Conclusions are presented in Section 4.

2. MINING INFORMATION FROM TRAFFIC VIDEO SEQUENCES

Traffic video analysis can discover and provide useful information, such as queue detection, vehicle classification, traffic flow, and incident detection at the intersections. To the best of our knowledge, the current transportation applications and research work either do not connect to databases or have limited capabilities to index and store the collected data (such as traffic videos) in their databases. Therefore, those applications cannot provide organized, unsupervised, conveniently accessible and easy-to-use multimedia information to traffic planners. In order to discover and provide some important but previously unknown knowledge from the traffic video sequences to the traffic planners, multimedia data mining techniques need to be employed. The proposed multimedia data-mining framework includes background subtraction, vehicle object identification and tracking, multimedia augmented transition network (MATN) model and multimedia input strings. The additional level of sophistication enabled by the proposed framework, in terms of spatio-temporal tracking, generates a capability for automation. This capability alone can significantly influence and enhance current data processing and implementation strategies for several problems vis-à-vis traffic operations.

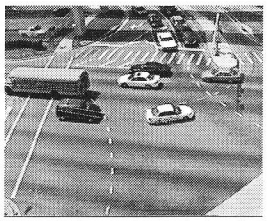
MATNs and multimedia input strings are used to model the temporal and relative spatial relations of the vehicle objects. An unsupervised video segmentation method, i.e., the SPCPE algorithm (see Section 2.2), can identify vehicle objects. In our framework, we introduce the technique of background subtraction to enhance the basic SPCPE algorithm to get better segmentation results, so that the more accurate spatio-temporal relationships of objects can be obtained. In the following subsections, we will first introduce the background subtraction technique, then give an overview of the SPCPE algorithm and the object tracking techniques, after that we will briefly describe how to use MATNs and multimedia input strings to model key video frames. A portion of the traffic video clips are used to demonstrate how video indexing is modeled by the MATNs and multimedia input strings.

2.1 Background Subtraction

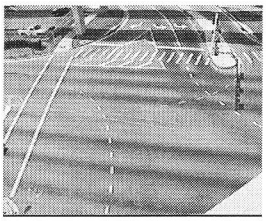
Background subtraction is a technique to remove non-moving components from a video sequence. The main

assumption for its application is that the camera remains stationary. The basic principle is to create a reference frame of the stationary components in the image. Once created, the reference frame is subtracted from any subsequent images. Those pixels resulting from new (moving) objects will generate a difference not equal to zero (i.e., difference $\neq 0$).

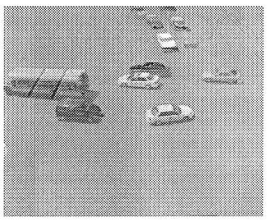
In this work, those video sequences containing non-moving objects were manually selected from the video data and then averaged together. The image sequence used consists of about 16 minutes of video from a traffic intersection with approximately constant lighting conditions (the sun was out and in Miami that means a bright day). Our approach is similar to that of [18] or [21], where a reference frame is constructed by accumulating and averaging images of the target area (the intersection in our case) for some time interval. As mentioned above, this is not a robust technique as it is sensitive to intensity variations [19]. That is, it can generate false positives since the detection of moving objects solely due to lighting changes. It can also generate false negatives due to the addition of stationary objects to the scene that are not part of the reference frame. [29] provides a good summary of the problems associated with background modeling. We use a simple averaging technique for this work as it allows us to quickly evaluate an upper limit on the performance improvement with our unsupervised segmentation algorithm.



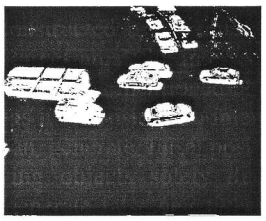
(a) Original frame 32



(b) Background reference frame



(c) Difference image



(d) Binary image

Figure 1: Example result of background subtraction

The difference image (as shown in Figure 1(c)) is created by subtracting the reference frame (as shown in Figure 1(b)) from the current image (as shown in Figure 1(a)). The results are scaled by $s = c\log(1+|d_{ij}|)$, where d_{ij} is the value for the difference at pixel ij. The scaling results in nonlinearly boosting the differences away from zero and towards 255 (the value of c will determine where saturation will occur). The results of the differencing step are fed to our unsupervised segmentation algorithm as the input images. Binary thresholding of the difference image can be used as an initial partition to improve the speed of converging (see Section 2.2) in our segmentation algorithm. Figure 1 gives an example result of background subtraction for frame 32.

2.2 Unsupervised Video Segmentation Method (SPCPE)

The SPCPE (Simultaneous Partition and Class Parameter Estimation) algorithm is an unsupervised video segmentation method to partition video frames. A given class description determines a partition. Similarly, a given partition gives rise to a class description, so the partition and the class parameter have to be estimated simultaneously. In practice, the class descriptions and their parameters are not readily available. An additional difficulty arises when images have to be partitioned automatically without the intervention of the user. Thus, we do not know a priori which pixels belong to which class. In the SPCPE algorithm, the partition and the class parameters are treated as random variables. The method for partitioning a video frame starts with an arbitrary partition and employs an iterative algorithm to estimate the partition and the class parameters jointly [6,27]. Since the successive frames in a video do not differ by much, the partitions of adjacent frames do not differ significantly. Each frame is partitioned by using the partition of the previous frame as an initial condition to speed up the convergence rate of the algorithm. A randomly generated initial partition, a learned partition for the domain or a binary image derived from the background difference is used for the first frame.

The mathematical description of a class specifies the pixel values as functions of the spatial coordinates of the pixel. The parameters of each class can be computed directly by using a least square technique. Suppose we have two classes. Let the partition variable be $c = \{c_1, c_2\}$ and the classes be parameterized by $\theta = \{\theta_1, \theta_2\}$. Also, suppose all the pixel values y_{ij} (in the image data Y) belonging to class k $(k=1,2)$ are put into a vector $\boldsymbol{Y_k}$. Each row of the matrix Φ is given by $(1, i, j, ij)$ and a_k is the vector of parameters $(a_{k0}, \ldots, a_{k3})^T$.

$$y_{ij} = a_{k0} + a_{k1} i + a_{k2} j + a_{k3} ij, \quad \forall (i, j) \; y_{ij} \in \boldsymbol{c_k}$$

$$\boldsymbol{Y_k} = \Phi \, a_k$$

$$\hat{a}_k = (\Phi^T \Phi)^{-1} \Phi^T \boldsymbol{Y_k}$$

We estimate the best partition as that which maximizes the a posteriori probability (MAP) of the partition variable given the image data Y. Now, the MAP estimates of $c = \{c_1, c_2\}$ and $\theta = \{\theta_1, \theta_2\}$ are given by

$$(\hat{c}, \hat{\theta}) = Arg \max_{(c,\theta)} P(c, \theta \mid Y)$$

$$= Arg \max_{(c,\theta)} P(Y \mid c, \theta) P(c, \theta)$$

We assume that the pixel values and parameters are independent and that the parameters are uniformly distributed. We also assume that the error function[1] of y_{ij} is represented by a Gaussian with mean 0 and variance 1. Let $J(c,\theta)$ be the functional to be minimized. With these assumptions the joint estimation can be simplified to the following form:

$$(\hat{c},\hat{\theta}) = Arg \min_{(c,\theta)} J(c_1,c_2,\theta_1,\theta_2)$$

$$J(c_1,c_2,\theta_1,\theta_2) = \sum_{y_{ij}\in c_1} -\ln p_1(y_{ij};\theta_1) + \sum_{y_{ij}\in c_2} -\ln p_2(y_{ij};\theta_2)$$

The minimization of J can be carried out alternately on c and θ in an iterative manner. Let $\hat{\theta}(c)$ represent the least squares estimates of the class parameters for a given partition c. The final expression for $J(c,\hat{\theta}(c))$ can be derived easily and is given by

$$J(c,\hat{\theta}(c)) = Arg \min_{(c_1,c_2)} \left\{ \frac{N_1}{2} \ln \hat{\rho}_1 + \frac{N_2}{2} \ln \hat{\rho}_2 \right\}$$

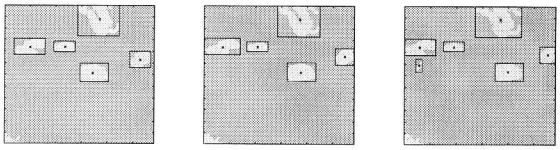
where $\hat{\rho}_1$ and $\hat{\rho}_2$ are the estimated model error variances of the two classes and N_1, N_2 are the number of pixels in each class. The algorithm starts with an arbitrary partition of the data and computes the corresponding class parameters. With these class parameters and the data, a new partition is estimated. Both the partition and the class parameters are iteratively refined until there is no further change in them.

2.3 Object Tracking

The first step for object tracking is to extract the segments in each class from each frame. Then the bounding box and the centroid point for each segment are obtained. For example, Figure 2(b) shows the segmentation results of the video sequence in Figure 2(a), where the vehicle objects belong to class 2 and the ground belongs to class 1. As shown in Figure 2(b), those segments corresponding to the vehicle objects are bounded by their minimal bounding boxes and represented by their centroid points.

The next step for object tracking is to connect the related segments in successive frames. The idea is to connect two segments that are spatially the closest in the adjacent frames [27]. In other words, the Euclidean distances between the centroids of the segments in the adjacent frames are used as the criteria to track the related segments. In addition, size restrictions are employed to determine the related segments in successive frames. A more sophisticated object tracking algorithm integrated into our framework is described in [5], which handles the situation of two objects overlapping under certain assumptions (e.g., the overlapped objects should have similar sizes). As shown in Figure 2 (case 1), there are two overlapped cars being identified as one segment because they are too close. In the algorithm in [5], if the two car objects have ever been separated from each other in the video sequence, then they can be split and identified as two objects, with their bounding boxes being fully recovered, since they have similar sizes.

(a) Example video sequence (frames 40, 41 and 42 from left to right).



(b) Segmentation mask maps and bounding boxes for (a)

Figure 2: Object tracking (case 1)

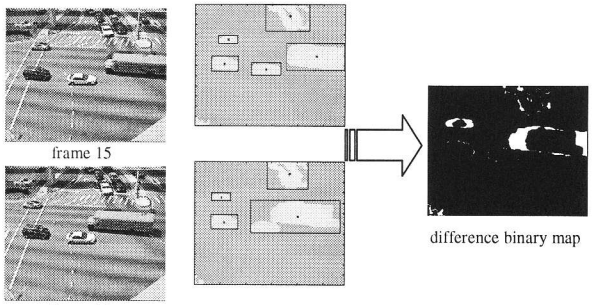


Figure 3: Object tracking (case 2)

On the other hand, in the situation that the overlapped objects have dissimilar sizes (case 2), for example the school bus and car in Figure 3, our existing algorithm [5] cannot find the school bus and car objects corresponding segments in the following frame (frame 16). In this example a large school bus and a small car that were detected as two objects in one frame (frame 15), were merged into a new big segment in the following frame (frame 16). However, from the new detected big segment in frame 16, we can reason that this is an '*overlapping*' segment that includes more than one vehicle object. A difference binary map knowledge discovery method is proposed to discover which objects the '*overlapping*' segment may include.

The idea is to obtain the difference binary map by subtracting the segment result of frame 16 from that of frame 15 and to compare the amount of differences between the two segmentation results of the consecutive frames. As shown in the difference binary map in Figure

[1] The model error is $e_{ij} = y_{ij} - (a_{k0} + a_{k1}i + a_{k2}j + a_{k3}ij)$.

3, the white areas in the difference binary map indicate the amount of differences between the segmentation results of the two consecutive frames. The car and school bus objects in frame 15 can be roughly mapped into the area of the big segment in frame 16 with relatively small differences. Hence, we can discover the vehicle objects in the big segment in frame 16 by reasoning that it is most probably related to the car and school bus objects from frame 15. In such a case, for the big segment (the '*overlapping*' segment) in frame 16, the corresponding links to the car and bus objects in frame 15 will be created.

2.4 Using MATNs and Multimedia Input Strings to Model Video Key Frames

A multimedia augmented transition network (MATN) model can be represented diagrammatically by a labeled directed graph, called a *transition graph*. A multimedia input string is accepted by the grammar if there is a path of transitions which corresponds to the sequence of symbols in the string and which leads from a specified initial state to one of a set of specified final states.

A MATN can build up a video hierarchy [7]. A video clip can be divided into *scenes*, a *scene* contains a sequential collection of *shots*, and each shot contains some contiguous frames that are at the lowest level in the video hierarchy [30]. It is advantageous to use several key frames to represent a shot instead of showing all these frames. Key frames play as the indices for a shot. The key frame selection approach proposed in [7] is based on the number, temporal, and spatial changes of the semantic objects in the video frames. Other features may also be possible for the key frame selections, but we focus on the number, temporal, and spatial relations of semantic objects. Therefore, these key frames can represent spatio-temporal changes in each shot. For example, in each shot of a traffic video sequence, the vehicles may change their positions in subsequent frames and the number of vehicles appearing may change at the time duration of the shot.

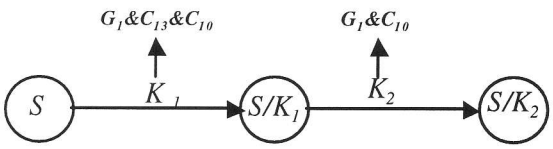
As introduced in [2], one semantic object is chosen as the target semantic object in each video frame and the minimal bounding rectangle (MBR) concept is used. In order to distinguish the 3-D relative positions, twenty-seven numbers are used [2]. In this paper, each frame is divided into nine sub-regions with the corresponding subscript numbers shown in Figure 4(a). Each key frame is represented by an input symbol in a multimedia input string and the "&" symbol between two vehicle objects is used to denote that the vehicle objects appear in the same frame. The subscripted numbers are used to distinguish the relative spatial positions of the vehicle objects relative to the target object "ground" (Figure 4(a)). For simplicity, two consecutive key frames are used to explain how to construct the multimedia input string and the MATN. The multimedia input string that represents these two key frames is as follows:

$$\underbrace{(G_1 \,\&\, C_{13} \,\&\, C_{10})}_{K_1}\ \underbrace{(G_1 \,\&\, C_{10})}_{K_2}$$

There are two input symbols, K_1 and K_2. The order of the vehicle objects in an input symbol is based on the relative spatial locations of the vehicle objects in the traffic video frame (from left to right and top to bottom). For example, the first key frame is represented by input symbol K_1. G_1 indicates that G is the target object. C_{13} *means* the first car object is on the left of and above G, and C_{10} means the second car object is on the left of G. For the next key frame, its multimedia input string is almost the same as that of frame 4 except that the car C_{13} that appeared in the first key frame has already left the road intersection in the next key frame. Hence, the number of vehicle objects decreases from two to one. This is an example to show how a multimedia input string can represent the change of the number of semantic (vehicle) objects.

13	4	22
10	1	19
16	7	25

(a) the nine sub-regions and their corresponding subscript numbers



(b) an example MATN model

Figure 4: MATN and multimedia input strings for modeling the key frames of traffic video shot *S*.

Figure 4(b) is the MATN for the above two key frames of the example traffic video sequence. The starting state name for this MATN is *S*/. As shown in Figure 4(b), there are two arcs with arc labels the same as the two input symbols (K_1 and K_2). The different state nodes in the MATN model the temporal relations of the selected key frames. The multimedia input strings model the relative spatial relations of the vehicle objects.

3. EXPERIMENT RESULTS AND DISCUSSIONS

A real life traffic video sequence is used to demonstrate the knowledge discovery process, i.e., spatio-temporal vehicle tracking, from the traffic video sequence using the proposed framework.

3.1 Experiment Setup

The traffic video sequence was captured with a *Sony Handycam CCD TR64* and digitized with a simple

Brooktree Bt848 based capture card on a *Windows NT 2000 Celeron* based platform. The video sequence consists of about 16 minutes of video from a traffic intersection with approximately constant lighting conditions. The original video frames were of size 480 rows×640 columns, 24 bit color and frame rate sampled at 5 frames per second. For simplicity and real-time processing purpose, we transform the color video frames to grayscale images and resize them to half of the original size (240 rows×320 columns). The traffic video sequence shows the traffic flow of an intersection on *US 1*, one of the busiest state roads in Miami, FL, USA.

A small portion of the traffic video is used to illustrate how the proposed framework can be applied to traffic applications to answer spatio-temporal queries like "Estimate the traffic flow of this road intersection from 8:00 AM to 8:30 AM." This query requires the use of multimedia data mining techniques to discover information such as the number of vehicles passing through the corresponding road intersection in a given time duration as well as the types of the vehicles (e.g., "car", "bus", etc.). This process can be done in real-time or off-line.

3.2 Experiment Results

The enhanced video segmentation method is applied to the video sequences by considering two classes. The first frame is partitioned into two classes using an initial random partition. After obtaining the final partition of the first frame (via SPCPE), we compute the partitions of the subsequent frames using the previous partitions as the initial partition parameter for the subsequent segmentation steps (since there is little significant difference between consecutive video frames). The convergence speed of the SPCPE algorithm is increased by using the previous partition results and thus provides support for real-time processing. The segmentation results for a few frames – 4, 9, 15, 16 and 35 – are shown in Figure 5 (end of paper) along with the original frames adjacent to them. These frames are the key frames after applying the key frame selection method introduced in [7]. As can be seen, the background of the traffic video sequence is complex. Related work has been done on the base of highway traffic videos [15,20] that have relatively simple backgrounds. Our framework, however can deal with more complex situations such as the traffic video for intersection monitoring.

In Figure 5, the frames in the leftmost column (Figure 5(a)) are the original frames. The second column (Figure 5(b)) shows the difference images after background subtraction. The final segmentation results are shown in the third column (Figure 5(c)). As can be seen from Figure 5(c), almost all of the vehicle objects are captured as separate segments (objects) except for those vehicles in the two lanes located in the upper part of the video frame (which has been captured as one segment because they appear too close together due to the shooting angle of the camera). From Figure 5(c), one can observe that the two-class partitioning schema can capture most of the relevant scene information (in regard to traffic applications). One class captures relevant vehicle information and the second class captures most of the ground information (the background non-vehicle information). Some of the vehicles have been combined with other objects into a single segment when they are closely located, for example, in frame 16, the school bus is overlapped with the car that was waiting in the middle of the intersection, while the school bus was moving westbound. Other cars in the main area of the intersection are successfully identified in all of these frames.

As only the vehicles are important for our application, we use the rightmost column in Figure 5(d) to show the relative spatial relationships of the vehicle segments for each frame. For the simplified segmentation results (Figure 5(d)), we use symbolic representations (multimedia input strings) to represent the spatial relationships of the vehicle objects in each frame. As shown in Figure 5(d), the ground ($\boldsymbol{G}$) is selected as the target object and the segments are denoted by $\boldsymbol{C}$ for cars or $\boldsymbol{B}$ for buses. For those cars combined together into a single segment (in the upper part of video frame), we use domain knowledge that there are two lanes located in the upper part of the scene where the vehicles are waiting before they enter the intersection. The use the symbol $\boldsymbol{W}$ for this special segment indicating that this is a '*waiting*' segment that may include more than one vehicle waiting to enter the intersection. Our data also contains vehicle objects in the main area of intersection that are combined into one segment. For example, the car object and the school bus were combined into one segment in frame 16, while they were separate segments in the preceding frame (frame 15). As discussed in Section 2, this occlusion situation can be detected by the proposed difference binary map knowledge discovery method. We use symbol $\boldsymbol{O}$ to denote an '*overlapping*' segment which has corresponding links to the related segments in the preceding frame.

As can be seen from Figure 5, the '*waiting*' segment always remains at the same location in the scene. In order to answer the query for traffic flow estimation, these '*waiting*' segments will not be counted. In the proposed symbolic representation, each vehicle segment is indexed in a multimedia input string based on the spatial relation of its centroid. The subscript numbers are used to denote the relative spatial relations of the vehicle objects with respect to the target object from the viewer's perspective. As mentioned earlier, $\boldsymbol{G_1}$ indicates that the ground ($\boldsymbol{G}$) is the target object and the subscript numbers have the same relative spatial meanings. In frames 4 and 9, two cars in the middle of the intersection ($\boldsymbol{C_{10}}$ and $\boldsymbol{C_1}$) were waiting to pass while another car ($\boldsymbol{C_4}$) was driving slowly through the upper part of the intersection westbound. In addition car ($\boldsymbol{C_{13}}$ in frame 4) was leaving the intersection

westbound. In frame 15, a school bus appeared as B_{19} from the east side; while in frame 16, the school bus and the white car (C_1 in frame 15) were combined into one *overlapping* segment (O_{19}). In frame 35, the school bus (B_{10}) was separated from the other cars and left the intersection on the west side, while the two cars (C_{10} and C_1 in frames 4, 9 and 15) made the left turn and moved towards the northeast bound so that their relative spatial locations changed to C_1 and C_{19} in frame 35.

As described above, it can be seen that the multimedia input strings can model not only the number of objects, but also the relative spatial relations. In this case, in order to estimate the intersection traffic flow, we can choose the east or west side of the intersection as a 'judge line' in the frame to determine the traffic flow of the specified direction (east↔west), and any vehicles passing through that line will be recorded. Using the information of centroid's position of each object, the traffic flow of a specified direction in the intersection area can be determined. Moreover, since the types of vehicles are also important for estimating the traffic flow, the sizes of the bounding boxes can be utilized to determine the vehicle types (such as 'car' and 'bus'). For those '*overlapping*' segments, since they have links to specific vehicle segments, the corresponding number and types of vehicles in an overlapping segment can be obtained in order to count the traffic flow. Besides answering the traffic flow query, the proposed framework also has the potential to answer other spatio-temporal related database queries.

4. CONCLUSION

Traffic video analysis can discover and provide useful information such as queue detection, vehicle classification, traffic flow, and incident detection at the intersections. Multimedia data mining techniques need to be employed in order to discover and provide important but previously unknown knowledge from the traffic video sequences to the traffic planners. In this paper, a multimedia data-mining framework that discovers the spatio-temporal relationships of the vehicle objects in the traffic video sequences is presented. The spatio-temporal relationships of the vehicle objects are discovered and captured via the unsupervised image/video segmentation method and the proposed object-tracking algorithm. The discovered spatio-temporal relationships of the vehicle objects are modeled by the multimedia augmented transition network (MATN) model and multimedia input strings. In order to eliminate the complex background information in the traffic video frames, background subtraction techniques are employed. Using the background subtraction technique, both the efficiency of the segmentation process and the accuracy of the segmentation results are improved achieving more accurate video indexing and annotation. This paper uses a real-life traffic video sequence on a state road intersection in Miami, FL, USA as the example video source. As shown in the results, the proposed framework can model complex situations such as the traffic video for intersection monitoring.

5. ACKNOWLEDGEMENT

For Shu-Ching Chen, this research was supported in part by NSF CDA-9711582.

REFERENCES

[1] Caltrans. Caltrans Live Traffic Cameras, http://video.dot.ca.gov/.

[2] Chen, S.-C. and Kashyap, R. L., "A Spatio-Temporal Semantic Model for Multimedia Database Systems and Multimedia Information Systems," *IEEE Transactions on Knowledge and Data Engineering, to appear.*

[3] Chen, S.-C., Shyu, M.-L., and Zhang, C., "An Unsupervised Segmentation Framework For Texture Image Queries," *The 25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC),* Chicago, Illinois, USA, Oct. 2000.

[4] Chen, S.-C., Shyu, M.-L., and Zhang, C., "An Intelligent Framework for Spatio-Temporal Vehicle Tracking," *4th International IEEE Conference on Intelligent Transportation Systems,* Oakland, California, USA, Aug. 2001.

[5] Chen, S.-C., Shyu, M.-L., Zhang, C., and Kashyap, R., "Object Tracking and Augmented Transition Network for Video Indexing and Modeling," *12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000),* Vancouver, British Columbia, Canada, pp. 428-435.

[6] Chen, S.-C., Sista, S., Shyu, M.-L., and Kashyap, R., "An Indexing and Searching Structure for Multimedia Database Systems," *IS&T/SPIE conference on Storage and Retrieval for Media Databases 2000,* San Jose, CA, U.S.A, pp. 262-270.

[7] Chen, S.-C., Sista, S., Shyu, M.-L., and Kashyap, R., "Augmented Transition Networks as Video Browsing Models for Multimedia Databases and Multimedia Information Systems," *11th IEEE International Conference on Tools With Artificial Intelligence (ICTAI'99),* Chicago, IL, U.S.A., Nov. 1999.

[8] Coen M, "The Future of Human-Computer Interaction or How I Learned to Stop Worrying and Love my Intelligent Room," *IEEE Intelligent Systems*, vol. 14, no. 2, pp. 8-10, Mar, 1999.

[9] Courtney, J. D., "Automatic Video Indexing via Object Motion Analysis," *Pattern Recognition*, vol. 30, no. 4, pp. 607-625, 1997.

[10] Cucchiara, R., Piccardi, M., and Mello, P., "Image Analysis and Rule-based Reasoning for a Traffic

Monitoring System," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119-130, June 2000.

[11] Dailey, D. J., Cathey, F., and Pumrin, S., "An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras," *IEEE Transactions on Intelligent Transportations Systems*, vol. 1, no. 2, pp. 98-107, Jun, 2000.

[12] DeMenthon, D., Stuckelberg, M., and Doermann, D., "Image Distance using Hidden Markov Models," *International Conference Pattern Recognition (ICPR 2000): Image, Speech and Signal Processing,* Barcelona, Spain, pp. 147-150, Sept. 2000.

[13] Fan, L. and Sung, K. K., "Model-Based Varying Pose Face Detection and Facial Feature Registration in Video Images," *8th ACM International Conference on Multimedia,* Los Angeles, CA, pp. 295-302, Oct. 2000.

[14] Ferman, A. M., Guensel, B., and Tekalp, A. M., "Object-based Indexing of MPEG-4 Compressed Video," *in Proceedings of SPIE: Visual Communications and Image Processing,* San Jose; CA, pp. 953-963, Feb. 1997.

[15] Friedman, N. and Russell, S., "Image Segmentation in Video Sequences: A Probabilistic Approach," *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI '97),* Providence; RI.

[16] Gonzalez, R. C. and Woods, R. E. *Digital image processing*, Reading, Mass: Addison-Wesley, 1993.

[17] Grimson, W. E. L., Stauffer, C., Romano, R., and Lee, L., "Using Adaptive Tracking to Classify and Monitor Activities in a Site," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Preceding,* pp. 22-31, 1998.

[18] Haritaoglu, I., Harwood, D., and Davis, L., "W 4 - Who, Where, When, What: A Real-Time System for Detecting and Tracking People," *IEEE Third International Conference on Face and Gesture Recognition,* Nara, Japan, pp. 222-227, 1998.

[19] Haritaoglu, I., Harwood, D., and Davis, L., "A Fast Background Scene Modeling and Maintenance for Outdoor Surveillance," *15th IEEE International Conference on Pattern Recognition: Applications, Robotics Systems and Architectures,* Barcelona, Spain, pp. 179-183, Sept. 2000.

[20] Huang, T., Koller, D., Malik, J., and Ogasawara, G., "Automatic Symbolic Traffic Scene Analysis Using Belief Networks," *Proceedings of the AAAI, 12th National Conference on Artificial Intelligence (AAAI '94),* Seattle, WA, pp. 966-972, July 1994.

[21] Kamijo, S., Matsushita, Y., Ikeuchi, K., and Sakauchi, M., "Traffic Monitoring and Accident Detection at Intersections," *IEEE International Conference on Intelligent Transportation Systems,* Tokyo Japan, pp. 703-708, Oct. 1999.

[22] Koller, D., Weber, J., and Malik, J., "Robust Multiple Car Tracking with Occlusion Reasoning," *3rd European Conference on Computer Vision, Eccv '94,* Stockholm Sweden, pp. 189-196, May 1994.

[23] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S., "Multi-Camera Multi-Person Tracking for EasyLiving," *3rd IEEE International Workshop on Visual Surveillance,* Dublin, Ireland, pp. 3-10, July 2000.

[24] Montgomery. Co. Department of Public Works Transportation. ATMS Video Monitoring System Live Traffic Camera Pictures, http://www.dpwt.com/jpgcap/camintro.html.

[25] Puzicha, J., Hofmann, T., and Buhmann, J. M., "Histogram Clustering for Unsupervised Image Segmentation," *IEEE Computer Society Conference Computer Vision and Pattern Recognition,* Fort Collins; CO, pp. 602-608, June 1999.

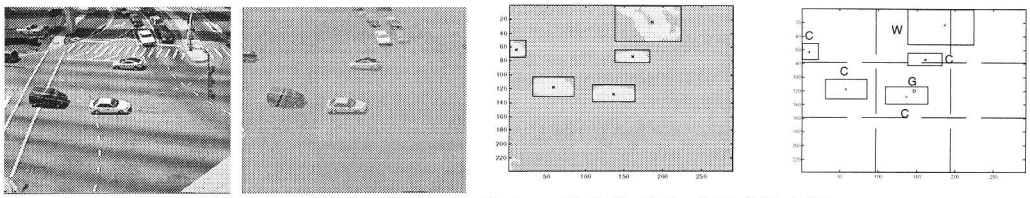
[26] Shafer, S., Krumm, J., Brumitt, B., Meyers, B., Czerwinski, M., and Robbins, D., "The New EasyLiving Project at Microsoft Research," *DARPA/NIST Workshop on Smart Spaces,* pp. 127-130, July 1998.

[27] *Sista,* S. and *Kashyap,* R. L., "Unsupervised Video Segmentation and Object Tracking," *Computers in Industry*, vol. 42, no. 2-3, pp. 127-146, June 2000.

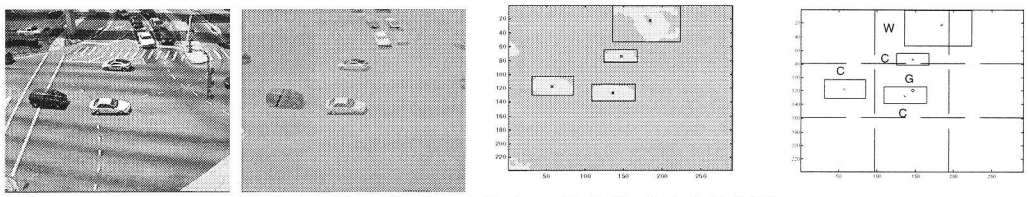
[28] Stauffer, C. and Grimson, W. E. L., "Adaptive Background Mixture Models for Real-Time Tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 246-252, 1999.

[29] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B., "Wallflower: Principles and Practice of Background Maintenance," *7th International Conference on Computer Vision (ICCV'99),* Held on the Island of Crete, pp. 255-261, Sept. 1999.

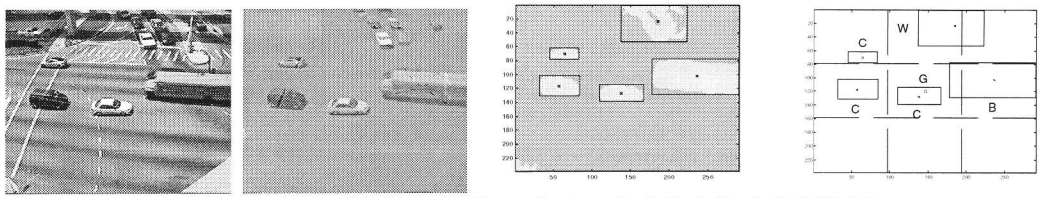
[30] Yeo, B.-L. and Yeung, M. M., "Retrieving and Visualizing Video," *Communications of the ACM*, vol. 40, no. 12, pp. 43-52, Dec. 1997.



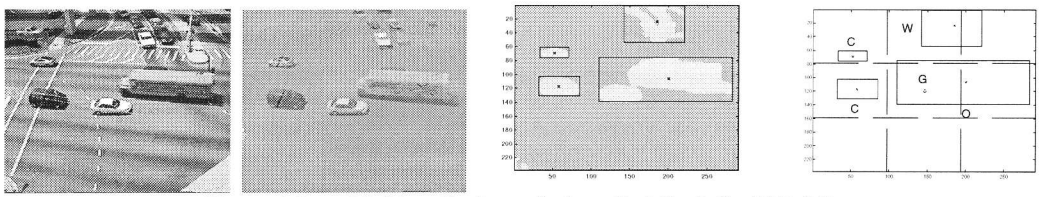
Frame 4 *Multimedia Input String:* $G_1\&C_{13}\&C_{10}\&C_1\&C_4\&W_4$



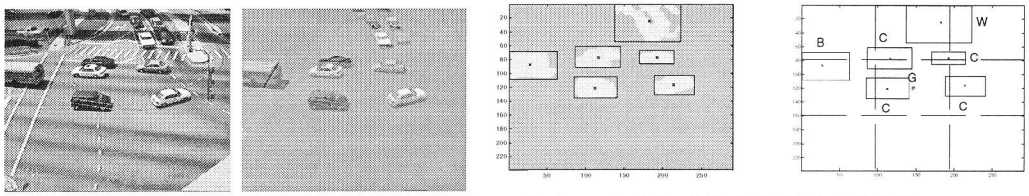
Frame 9 *Multimedia Input String:* $G_1\&C_{10}\&C_1\&C_4\&W_4$



Frame 15 *Multimedia Input String:* $G_1\&C_{13}\&C_{10}\&C_1\&W_4\&B_{19}$



Frame 16 *Multimedia Input String:* $G_1\&C_{13}\&C_{10}\&W_4\&O_{19}$



Frame 35 *Multimedia Input String:* $G_1\&B_{10}\&C_1\&C_4\&W_4\&C_4\&C_{19}$

(a) Original frames. (b) Difference frames. (c) Segmentation results. (d)Bounding boxes.

Figure 5: Segmentation results as well as the multimedia input strings for frames 4, 9, 15, 16 and 35. The leftmost column gives the original video frames; the second column shows difference images obtained by subtracting the background reference frame from the original frames; the third column shows the vehicle segments extracted from the video frames, and the rightmost column shows the bounding boxes of the vehicle objects.

01-MD

Multimedia Data Mining for Traffic Video Sequences

Shu-Ching Chen[1], Mei-Ling Shyu[2], Chengcui Zhang[1], Jeff Strickrott[1]
[1]Distributed Multimedia Information System Laboratory
School of Computer Science, Florida International University, Miami, FL 33199
[2]Department of Electrical and Computer Engineering, University of Miami,
Coral Gables, FL 33124

ABSTRACT

In this paper, a multimedia data mining framework for discovering important but previously unknown knowledge such as vehicle identification, traffic flow, and the spatio-temporal relations of the vehicles at the intersections from traffic video sequences is proposed. The proposed multimedia data mining framework analyzes the traffic video sequences by using background subtraction, image/video segmentation, object tracking, and modeling with multimedia augmented transition network (MATN) model and multimedia input strings, in the domain of traffic monitoring over an intersection. The spatio-temporal relationships of the vehicle objects in each frame are discovered and accurately captured and modeled. Such an additional level of sophistication enabled by the proposed multimedia data-mining framework in terms of spatio-temporal tracking generates a capability for automation. This capability alone can significantly influence and enhance current data processing and implementation strategies for several problems vis-à-vis traffic operations. A real-life traffic video sequence is used to illustrate the effectiveness of the proposed multimedia data mining framework.

KEY WORDS: Multimedia data mining, spatio-temporal relationships, multimedia augmented transition network (MATN), object tracking.

1. INTRODUCTION

As computers have become more powerful, their role in everyday life has become more pervasive. Recent efforts [8,23,26] have begun to shift the traditional focus from user centric applications (i.e., word processors, browsers, etc.) to that of a ubiquitous tool that facilitates everyday activities. Projects like EasyLiving [23,26] and HAL [8] aim to develop smart spaces that can monitor, predict, and assist the activities of its occupants. These efforts at developing smart environments are not confined to homes or offices, but extend to that of the world around us. Municipalities [1,24] are installing video camera systems to monitor and extract traffic control information from their highways in real time. Issues associated with extracting traffic movement and recognizing accident information from real time video sequences are discussed in [10,11,20,21,22]. Two common themes exist in these works. First, the video information must be segmented and turned into objects. Second, the behavior of those objects is monitored (they are tracked) for immediate decision making purposes. What is missing in these efforts is to model and index the data for on-line analysis, storage or later pattern mining.

The analysis and mining of traffic video sequences to discover information, such as vehicle identification, traffic flow, and the spatio-temporal relations of the vehicles at intersections, provides an economic approach for daily traffic operations. In order to identify and track the temporal and relative spatial positions of vehicle objects in video sequences, it is necessary to have object-based representation of video data. For this purpose, attention has been devoted to segmenting video frames into regions such that each region, or a group of regions, corresponds to an object that is meaningful to human viewers [9,13,14]. While most of the previous works are based on low-level global features, such as color histogram and texture, our video segmentation method focuses on obtaining object level segmentation; obtaining objects in each frame and their traces across the frames. In [3-5] we have addressed the issues of unsupervised image segmentation; object modeling with multimedia input strings to capture the spatial–temporal behavior of the object, and the application of these techniques to the domain of traffic monitoring.

Similar approaches to our segmentation technique are discussed in [12,25]. In [25] the authors consider a Bayesian technique to segment images based on feature distributions. The histogram of features around a pixel neighborhood is considered as an estimate of the conditional probability distribution $P(c|Y)$ versus the parametric equation in our approach (see Section 2.2). This technique models the texture in a neighborhood. DeMenthon et al. [12] utilize a Hidden Markov Model approach for low level image segmentation. Associated with each pixel are an observation vector and a hidden state. The observation vector is the set of parameters (of interest) associated with each pixel, such as color, or the average intensity of the image region centered on that

pixel. The hidden state is a label for that pixel. Computational time is $O(ns^3)$, where n is the number of pixels in the image and s is the number of states (regions) to segment the image. Segmentation in an image can also be modeled as a pixel-labeling problem, in which we must decide from which of M number of classes the pixel belongs. The membership in each class is formulated as a Bayesian conditional probability decision, where class membership is estimated from the intensity distributions of neighboring pixels. When the image segmentation problem is considered for a fixed camera domain, a classic technique to resolve the foreground objects is background subtraction [16]. This involves the creation of a background model that is subtracted from the input image to create a difference image. The new difference image only contains objects not in the background or new features that have not yet been incorporated into the background.

Various approaches to background subtraction and modeling techniques have been discussed in the literature [11,17,19,28], ranging from modeling the intensity variations of a pixel via a mixture of Gaussian distributions to simple differencing of successive images. In [29] the authors provide some simple guidelines and evaluation of the various techniques for background modeling. We are in the beginning phases of evaluating the performance benefits of background subtraction methods for the various domains of our image segmentation applications. To that aim we have evaluated the effectiveness of simple image averaging techniques over stationary (non-changing) portions of the image data set.

In this paper, a multimedia data mining framework for traffic video sequences is proposed. The proposed framework considers image/video segmentation with initial background subtraction, object tracking, and modeling with multimedia augmented transition network (MATN) model and multimedia input strings [2,7], in the domain of traffic monitoring over an intersection. The multimedia input strings are used to capture the spatio-temporal relationships of vehicle objects thereafter. The video segmentation method mentioned here is unsupervised. Another advantage is that it uses the segmentation result of the previous video frame to speed up the segmentation process of the current video frame. Experiments were conducted to illustrate the effectiveness of the proposed framework using a real-life traffic video sequence. The traffic video sequence was captured with a *Sony Handycam CCD TR64* and digitized with an inexpensive *Brooktree Bt848* based capture card on a Windows NT 2000 Celeron based platform. The original images are 640x480, 24 bit color and the video sequence was sampled at 5 frames per second.

The organization of this paper is as follows. In next section, the knowledge discovery process that includes background subtraction, the unsupervised segmentation algorithm, object tracking techniques, MATN model, and multimedia input strings are introduced. Experiment results and analysis of the proposed multimedia data mining framework are discussed in Section 3. Along with the discussion, an example real-life traffic video sequence is used. Conclusions are presented in Section 4.

2. MINING INFORMATION FROM TRAFFIC VIDEO SEQUENCES

Traffic video analysis can discover and provide useful information, such as queue detection, vehicle classification, traffic flow, and incident detection at the intersections. To the best of our knowledge, the current transportation applications and research work either do not connect to databases or have limited capabilities to index and store the collected data (such as traffic videos) in their databases. Therefore, those applications cannot provide organized, unsupervised, conveniently accessible and easy-to-use multimedia information to traffic planners. In order to discover and provide some important but previously unknown knowledge from the traffic video sequences to the traffic planners, multimedia data mining techniques need to be employed. The proposed multimedia data-mining framework includes background subtraction, vehicle object identification and tracking, multimedia augmented transition network (MATN) model and multimedia input strings. The additional level of sophistication enabled by the proposed framework, in terms of spatio-temporal tracking, generates a capability for automation. This capability alone can significantly influence and enhance current data processing and implementation strategies for several problems vis-à-vis traffic operations.

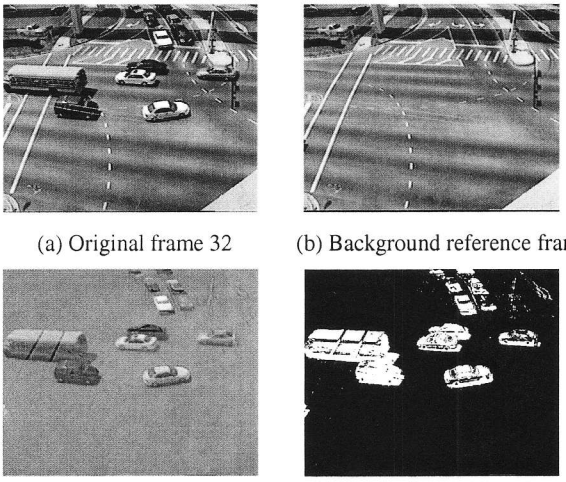
MATNs and multimedia input strings are used to model the temporal and relative spatial relations of the vehicle objects. An unsupervised video segmentation method, i.e., the SPCPE algorithm (see Section 2.2), can identify vehicle objects. In our framework, we introduce the technique of background subtraction to enhance the basic SPCPE algorithm to get better segmentation results, so that the more accurate spatio-temporal relationships of objects can be obtained. In the following subsections, we will first introduce the background subtraction technique, then give an overview of the SPCPE algorithm and the object tracking techniques, after that we will briefly describe how to use MATNs and multimedia input strings to model key video frames. A portion of the traffic video clips are used to demonstrate how video indexing is modeled by the MATNs and multimedia input strings.

2.1 Background Subtraction

Background subtraction is a technique to remove non-moving components from a video sequence. The main

assumption for its application is that the camera remains stationary. The basic principle is to create a reference frame of the stationary components in the image. Once created, the reference frame is subtracted from any subsequent images. Those pixels resulting from new (moving) objects will generate a difference not equal to zero (i.e., difference $\neq 0$).

In this work, those video sequences containing non-moving objects were manually selected from the video data and then averaged together. The image sequence used consists of about 16 minutes of video from a traffic intersection with approximately constant lighting conditions (the sun was out and in Miami that means a bright day). Our approach is similar to that of [18] or [21], where a reference frame is constructed by accumulating and averaging images of the target area (the intersection in our case) for some time interval. As mentioned above, this is not a robust technique as it is sensitive to intensity variations [19]. That is, it can generate false positives since the detection of moving objects solely due to lighting changes. It can also generate false negatives due to the addition of stationary objects to the scene that are not part of the reference frame. [29] provides a good summary of the problems associated with background modeling. We use a simple averaging technique for this work as it allows us to quickly evaluate an upper limit on the performance improvement with our unsupervised segmentation algorithm.



(a) Original frame 32 (b) Background reference frame

(c) Difference image (d) Binary image

Figure 1: Example result of background subtraction

The difference image (as shown in Figure 1(c)) is created by subtracting the reference frame (as shown in Figure 1(b)) from the current image (as shown in Figure 1(a)). The results are scaled by $s = c\log(1+|d_{ij}|)$, where d_{ij} is the value for the difference at pixel ij. The scaling results in nonlinearly boosting the differences away from zero and towards 255 (the value of c will determine where saturation will occur). The results of the differencing step are fed to our unsupervised segmentation algorithm as the input images. Binary thresholding of the difference image can be used as an initial partition to improve the speed of converging (see Section 2.2) in our segmentation algorithm. Figure 1 gives an example result of background subtraction for frame 32.

2.2 Unsupervised Video Segmentation Method (SPCPE)

The SPCPE (Simultaneous Partition and Class Parameter Estimation) algorithm is an unsupervised video segmentation method to partition video frames. A given class description determines a partition. Similarly, a given partition gives rise to a class description, so the partition and the class parameter have to be estimated simultaneously. In practice, the class descriptions and their parameters are not readily available. An additional difficulty arises when images have to be partitioned automatically without the intervention of the user. Thus, we do not know a priori which pixels belong to which class. In the SPCPE algorithm, the partition and the class parameters are treated as random variables. The method for partitioning a video frame starts with an arbitrary partition and employs an iterative algorithm to estimate the partition and the class parameters jointly [6,27]. Since the successive frames in a video do not differ by much, the partitions of adjacent frames do not differ significantly. Each frame is partitioned by using the partition of the previous frame as an initial condition to speed up the convergence rate of the algorithm. A randomly generated initial partition, a learned partition for the domain or a binary image derived from the background difference is used for the first frame.

The mathematical description of a class specifies the pixel values as functions of the spatial coordinates of the pixel. The parameters of each class can be computed directly by using a least square technique. Suppose we have two classes. Let the partition variable be $c = \{c_1, c_2\}$ and the classes be parameterized by $\theta = \{\theta_1, \theta_2\}$. Also, suppose all the pixel values y_{ij} (in the image data Y) belonging to class k $(k=1,2)$ are put into a vector $\boldsymbol{Y_k}$. Each row of the matrix Φ is given by $(1, i, j, ij)$ and a_k is the vector of parameters $(a_{k0}, \ldots, a_{k3})^T$.

$$y_{ij} = a_{k0} + a_{k1} i + a_{k2} j + a_{k3} ij, \quad \forall (i, j) \; y_{ij} \in \boldsymbol{c_k}$$

$$\boldsymbol{Y_k} = \Phi \, a_k$$

$$\hat{a}_k = (\Phi^T \Phi)^{-1} \Phi^T \boldsymbol{Y_k}$$

We estimate the best partition as that which maximizes the a posteriori probability (MAP) of the partition variable given the image data Y. Now, the MAP estimates of $c = \{c_1, c_2\}$ and $\theta = \{\theta_1, \theta_2\}$ are given by

$$(\hat{c}, \hat{\theta}) = Arg \max_{(c,\theta)} P(c, \theta \mid Y)$$

$$= Arg \max_{(c,\theta)} P(Y \mid c, \theta) P(c, \theta)$$

We assume that the pixel values and parameters are independent and that the parameters are uniformly distributed. We also assume that the error function[1] of y_{ij} is represented by a Gaussian with mean 0 and variance 1. Let $J(c,\theta)$ be the functional to be minimized. With these assumptions the joint estimation can be simplified to the following form:

$$(\hat{c},\hat{\theta}) = Arg \min_{(c,\theta)} J(c_1,c_2,\theta_1,\theta_2)$$

$$J(c_1,c_2,\theta_1,\theta_2) = \sum_{y_{ij}\in c_1} -\ln p_1(y_{ij};\theta_1) + \sum_{y_{ij}\in c_2} -\ln p_2(y_{ij};\theta_2)$$

The minimization of J can be carried out alternately on c and θ in an iterative manner. Let $\hat{\theta}(c)$ represent the least squares estimates of the class parameters for a given partition c. The final expression for $J(c,\hat{\theta}(c))$ can be derived easily and is given by

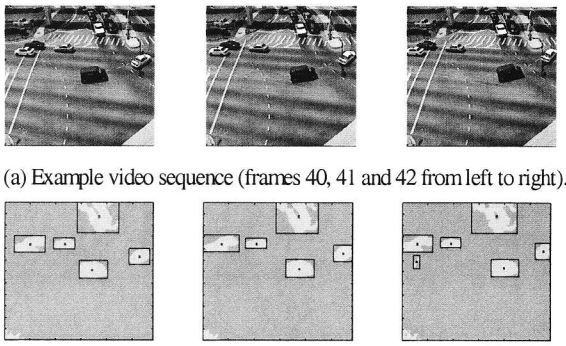
$$J(c,\hat{\theta}(c)) = Arg \min_{(c_1,c_2)} \left\{ \frac{N_1}{2} \ln \hat{\rho}_1 + \frac{N_2}{2} \ln \hat{\rho}_2 \right\}$$

where $\hat{\rho}_1$ and $\hat{\rho}_2$ are the estimated model error variances of the two classes and N_1, N_2 are the number of pixels in each class. The algorithm starts with an arbitrary partition of the data and computes the corresponding class parameters. With these class parameters and the data, a new partition is estimated. Both the partition and the class parameters are iteratively refined until there is no further change in them.

2.3 Object Tracking

The first step for object tracking is to extract the segments in each class from each frame. Then the bounding box and the centroid point for each segment are obtained. For example, Figure 2(b) shows the segmentation results of the video sequence in Figure 2(a), where the vehicle objects belong to class 2 and the ground belongs to class 1. As shown in Figure 2(b), those segments corresponding to the vehicle objects are bounded by their minimal bounding boxes and represented by their centroid points.

The next step for object tracking is to connect the related segments in successive frames. The idea is to connect two segments that are spatially the closest in the adjacent frames [27]. In other words, the Euclidean distances between the centroids of the segments in the adjacent frames are used as the criteria to track the related segments. In addition, size restrictions are employed to determine the related segments in successive frames. A more sophisticated object tracking algorithm integrated into our framework is described in [5], which handles the situation of two objects overlapping under certain assumptions (e.g., the overlapped objects should have similar sizes). As shown in Figure 2 (case 1), there are two overlapped cars being identified as one segment because they are too close. In the algorithm in [5], if the two car objects have ever been separated from each other in the video sequence, then they can be split and identified as two objects, with their bounding boxes being fully recovered, since they have similar sizes.



(a) Example video sequence (frames 40, 41 and 42 from left to right).

(b) Segmentation mask maps and bounding boxes for (a)

Figure 2: Object tracking (case 1)

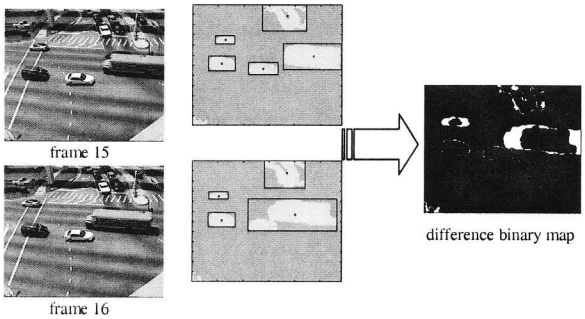


Figure 3: Object tracking (case 2)

On the other hand, in the situation that the overlapped objects have dissimilar sizes (case 2), for example the school bus and car in Figure 3, our existing algorithm [5] cannot find the school bus and car objects corresponding segments in the following frame (frame 16). In this example a large school bus and a small car that were detected as two objects in one frame (frame 15), were merged into a new big segment in the following frame (frame 16). However, from the new detected big segment in frame 16, we can reason that this is an '*overlapping*' segment that includes more than one vehicle object. A difference binary map knowledge discovery method is proposed to discover which objects the '*overlapping*' segment may include.

The idea is to obtain the difference binary map by subtracting the segment result of frame 16 from that of frame 15 and to compare the amount of differences between the two segmentation results of the consecutive frames. As shown in the difference binary map in Figure

[1] The model error is $e_{ij} = y_{ij} - (a_{k0} + a_{k1}i + a_{k2}j + a_{k3}ij)$.

3, the white areas in the difference binary map indicate the amount of differences between the segmentation results of the two consecutive frames. The car and school bus objects in frame 15 can be roughly mapped into the area of the big segment in frame 16 with relatively small differences. Hence, we can discover the vehicle objects in the big segment in frame 16 by reasoning that it is most probably related to the car and school bus objects from frame 15. In such a case, for the big segment (the '*overlapping*' segment) in frame 16, the corresponding links to the car and bus objects in frame 15 will be created.

2.4 Using MATNs and Multimedia Input Strings to Model Video Key Frames

A multimedia augmented transition network (MATN) model can be represented diagrammatically by a labeled directed graph, called a *transition graph*. A multimedia input string is accepted by the grammar if there is a path of transitions which corresponds to the sequence of symbols in the string and which leads from a specified initial state to one of a set of specified final states.

A MATN can build up a video hierarchy [7]. A video clip can be divided into *scenes*, a *scene* contains a sequential collection of *shots*, and each shot contains some contiguous frames that are at the lowest level in the video hierarchy [30]. It is advantageous to use several key frames to represent a shot instead of showing all these frames. Key frames play as the indices for a shot. The key frame selection approach proposed in [7] is based on the number, temporal, and spatial changes of the semantic objects in the video frames. Other features may also be possible for the key frame selections, but we focus on the number, temporal, and spatial relations of semantic objects. Therefore, these key frames can represent spatio-temporal changes in each shot. For example, in each shot of a traffic video sequence, the vehicles may change their positions in subsequent frames and the number of vehicles appearing may change at the time duration of the shot.

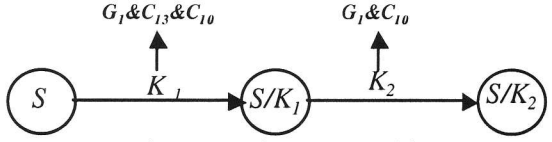
As introduced in [2], one semantic object is chosen as the target semantic object in each video frame and the minimal bounding rectangle (MBR) concept is used. In order to distinguish the 3-D relative positions, twenty-seven numbers are used [2]. In this paper, each frame is divided into nine sub-regions with the corresponding subscript numbers shown in Figure 4(a). Each key frame is represented by an input symbol in a multimedia input string and the "&" symbol between two vehicle objects is used to denote that the vehicle objects appear in the same frame. The subscripted numbers are used to distinguish the relative spatial positions of the vehicle objects relative to the target object "ground" (Figure 4(a)). For simplicity, two consecutive key frames are used to explain how to construct the multimedia input string and the MATN. The multimedia input string that represents these two key frames is as follows:

$$\underbrace{(G_1 \,\&\, C_{13} \,\&\, C_{10})}_{K_1}\ \underbrace{(G_1 \,\&\, C_{10})}_{K_2}$$

There are two input symbols, K_1 and K_2. The order of the vehicle objects in an input symbol is based on the relative spatial locations of the vehicle objects in the traffic video frame (from left to right and top to bottom). For example, the first key frame is represented by input symbol K_1. G_1 indicates that G is the target object. C_{13} *means* the first car object is on the left of and above G, and C_{10} means the second car object is on the left of G. For the next key frame, its multimedia input string is almost the same as that of frame 4 except that the car C_{13} that appeared in the first key frame has already left the road intersection in the next key frame. Hence, the number of vehicle objects decreases from two to one. This is an example to show how a multimedia input string can represent the change of the number of semantic (vehicle) objects.

13	4	22
10	1	19
16	7	25

(a) the nine sub-regions and their corresponding subscript numbers



(b) an example MATN model

Figure 4: MATN and multimedia input strings for modeling the key frames of traffic video shot *S*.

Figure 4(b) is the MATN for the above two key frames of the example traffic video sequence. The starting state name for this MATN is *S/*. As shown in Figure 4(b), there are two arcs with arc labels the same as the two input symbols (K_1 and K_2). The different state nodes in the MATN model the temporal relations of the selected key frames. The multimedia input strings model the relative spatial relations of the vehicle objects.

3. EXPERIMENT RESULTS AND DISCUSSIONS

A real life traffic video sequence is used to demonstrate the knowledge discovery process, i.e., spatio-temporal vehicle tracking, from the traffic video sequence using the proposed framework.

3.1 Experiment Setup

The traffic video sequence was captured with a *Sony Handycam CCD TR64* and digitized with a simple

Brooktree Bt848 based capture card on a *Windows NT 2000 Celeron* based platform. The video sequence consists of about 16 minutes of video from a traffic intersection with approximately constant lighting conditions. The original video frames were of size 480 rows×640 columns, 24 bit color and frame rate sampled at 5 frames per second. For simplicity and real-time processing purpose, we transform the color video frames to grayscale images and resize them to half of the original size (240 rows×320 columns). The traffic video sequence shows the traffic flow of an intersection on *US 1*, one of the busiest state roads in Miami, FL, USA.

A small portion of the traffic video is used to illustrate how the proposed framework can be applied to traffic applications to answer spatio-temporal queries like "Estimate the traffic flow of this road intersection from 8:00 AM to 8:30 AM." This query requires the use of multimedia data mining techniques to discover information such as the number of vehicles passing through the corresponding road intersection in a given time duration as well as the types of the vehicles (e.g., "car", "bus", etc.). This process can be done in real-time or off-line.

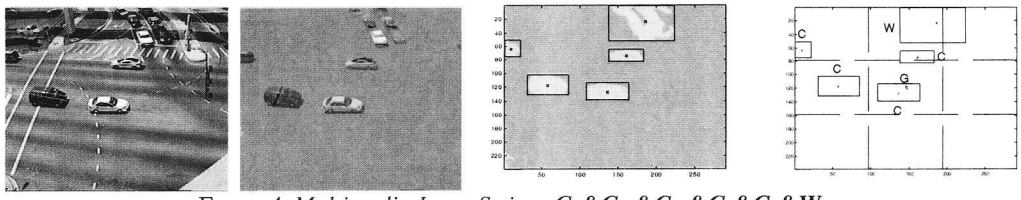
3.2 Experiment Results

The enhanced video segmentation method is applied to the video sequences by considering two classes. The first frame is partitioned into two classes using an initial random partition. After obtaining the final partition of the first frame (via SPCPE), we compute the partitions of the subsequent frames using the previous partitions as the initial partition parameter for the subsequent segmentation steps (since there is little significant difference between consecutive video frames). The convergence speed of the SPCPE algorithm is increased by using the previous partition results and thus provides support for real-time processing. The segmentation results for a few frames – 4, 9, 15, 16 and 35 – are shown in Figure 5 (end of paper) along with the original frames adjacent to them. These frames are the key frames after applying the key frame selection method introduced in [7]. As can be seen, the background of the traffic video sequence is complex. Related work has been done on the base of highway traffic videos [15,20] that have relatively simple backgrounds. Our framework, however can deal with more complex situations such as the traffic video for intersection monitoring.

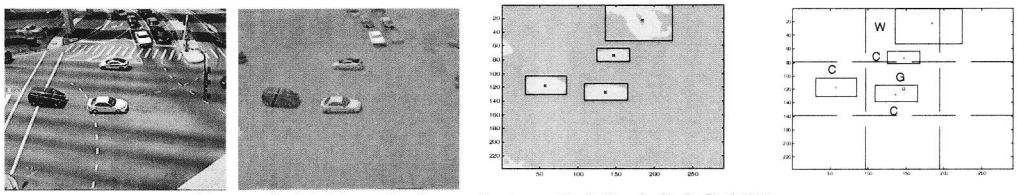
In Figure 5, the frames in the leftmost column (Figure 5(a)) are the original frames. The second column (Figure 5(b)) shows the difference images after background subtraction. The final segmentation results are shown in the third column (Figure 5(c)). As can be seen from Figure 5(c), almost all of the vehicle objects are captured as separate segments (objects) except for those vehicles in the two lanes located in the upper part of the video frame (which has been captured as one segment because they appear too close together due to the shooting angle of the camera). From Figure 5(c), one can observe that the two-class partitioning schema can capture most of the relevant scene information (in regard to traffic applications). One class captures relevant vehicle information and the second class captures most of the ground information (the background non-vehicle information). Some of the vehicles have been combined with other objects into a single segment when they are closely located, for example, in frame 16, the school bus is overlapped with the car that was waiting in the middle of the intersection, while the school bus was moving westbound. Other cars in the main area of the intersection are successfully identified in all of these frames.

As only the vehicles are important for our application, we use the rightmost column in Figure 5(d) to show the relative spatial relationships of the vehicle segments for each frame. For the simplified segmentation results (Figure 5(d)), we use symbolic representations (multimedia input strings) to represent the spatial relationships of the vehicle objects in each frame. As shown in Figure 5(d), the ground (G) is selected as the target object and the segments are denoted by C for cars or B for buses. For those cars combined together into a single segment (in the upper part of video frame), we use domain knowledge that there are two lanes located in the upper part of the scene where the vehicles are waiting before they enter the intersection. The use the symbol W for this special segment indicating that this is a '*waiting*' segment that may include more than one vehicle waiting to enter the intersection. Our data also contains vehicle objects in the main area of intersection that are combined into one segment. For example, the car object and the school bus were combined into one segment in frame 16, while they were separate segments in the preceding frame (frame 15). As discussed in Section 2, this occlusion situation can be detected by the proposed difference binary map knowledge discovery method. We use symbol O to denote an '*overlapping*' segment which has corresponding links to the related segments in the preceding frame.

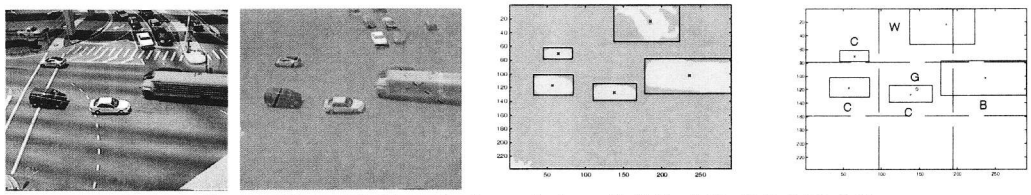
As can be seen from Figure 5, the '*waiting*' segment always remains at the same location in the scene. In order to answer the query for traffic flow estimation, these '*waiting*' segments will not be counted. In the proposed symbolic representation, each vehicle segment is indexed in a multimedia input string based on the spatial relation of its centroid. The subscript numbers are used to denote the relative spatial relations of the vehicle objects with respect to the target object from the viewer's perspective. As mentioned earlier, G_1 indicates that the ground (G) is the target object and the subscript numbers have the same relative spatial meanings. In frames 4 and 9, two cars in the middle of the intersection (C_{10} and C_1) were waiting to pass while another car (C_4) was driving slowly through the upper part of the intersection westbound. In addition car (C_{13} in frame 4) was leaving the intersection



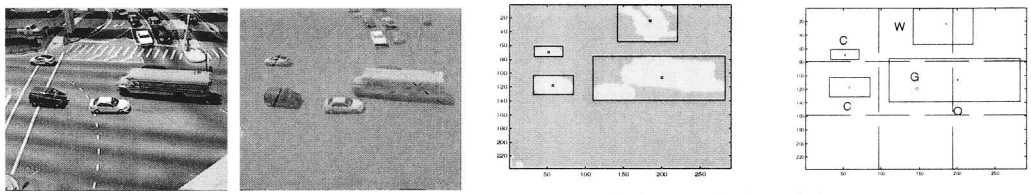
Frame 4 *Multimedia Input String:* $G_1\&C_{13}\&C_{10}\&C_1\&C_4\&W_4$



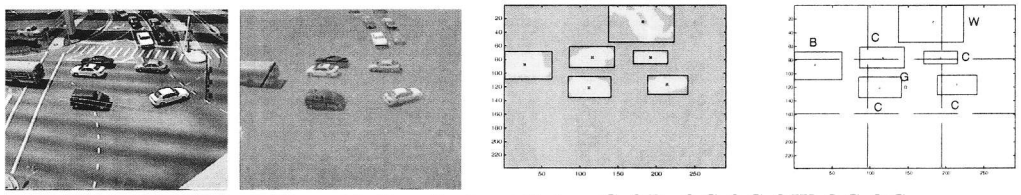
Frame 9 *Multimedia Input String:* $G_1\&C_{10}\&C_1\&C_4\&W_4$



Frame 15 *Multimedia Input String:* $G_1\&C_{13}\&C_{10}\&C_1\&W_4\&B_{19}$



Frame 16 *Multimedia Input String:* $G_1\&C_{13}\&C_{10}\&W_4\&O_{19}$



Frame 35 *Multimedia Input String:* $G_1\&B_{10}\&C_1\&C_4\&W_4\&C_4\&C_{19}$

(a) Original frames. (b) Difference frames. (c) Segmentation results. (d)Bounding boxes.

Figure 5: Segmentation results as well as the multimedia input strings for frames 4, 9, 15, 16 and 35. The leftmost column gives the original video frames; the second column shows difference images obtained by subtracting the background reference frame from the original frames; the third column shows the vehicle segments extracted from the video frames, and the rightmost column shows the bounding boxes of the vehicle objects.

westbound. In frame 15, a school bus appeared as B_{19} from the east side; while in frame 16, the school bus and the white car (C_1 in frame 15) were combined into one *overlapping* segment (O_{19}). In frame 35, the school bus (B_{10}) was separated from the other cars and left the intersection on the west side, while the two cars (C_{10} and C_1 in frames 4, 9 and 15) made the left turn and moved towards the northeast bound so that their relative spatial locations changed to C_1 and C_{19} in frame 35.

As described above, it can be seen that the multimedia input strings can model not only the number of objects, but also the relative spatial relations. In this case, in order to estimate the intersection traffic flow, we can choose the east or west side of the intersection as a 'judge line' in the frame to determine the traffic flow of the specified direction (east↔west), and any vehicles passing through that line will be recorded. Using the information of centroid's position of each object, the traffic flow of a specified direction in the intersection area can be determined. Moreover, since the types of vehicles are also important for estimating the traffic flow, the sizes of the bounding boxes can be utilized to determine the vehicle types (such as 'car' and 'bus'). For those '*overlapping*' segments, since they have links to specific vehicle segments, the corresponding number and types of vehicles in an overlapping segment can be obtained in order to count the traffic flow. Besides answering the traffic flow query, the proposed framework also has the potential to answer other spatio-temporal related database queries.

4. CONCLUSION

Traffic video analysis can discover and provide useful information such as queue detection, vehicle classification, traffic flow, and incident detection at the intersections. Multimedia data mining techniques need to be employed in order to discover and provide important but previously unknown knowledge from the traffic video sequences to the traffic planners. In this paper, a multimedia data-mining framework that discovers the spatio-temporal relationships of the vehicle objects in the traffic video sequences is presented. The spatio-temporal relationships of the vehicle objects are discovered and captured via the unsupervised image/video segmentation method and the proposed object-tracking algorithm. The discovered spatio-temporal relationships of the vehicle objects are modeled by the multimedia augmented transition network (MATN) model and multimedia input strings. In order to eliminate the complex background information in the traffic video frames, background subtraction techniques are employed. Using the background subtraction technique, both the efficiency of the segmentation process and the accuracy of the segmentation results are improved achieving more accurate video indexing and annotation. This paper uses a real-life traffic video sequence on a state road intersection in Miami, FL, USA as the example video source. As shown in the results, the proposed framework can model complex situations such as the traffic video for intersection monitoring.

5. ACKNOWLEDGEMENT

For Shu-Ching Chen, this research was supported in part by NSF CDA-9711582.

REFERENCES

[1] Caltrans. Caltrans Live Traffic Cameras, http://video.dot.ca.gov/.

[2] Chen, S.-C. and Kashyap, R. L., "A Spatio-Temporal Semantic Model for Multimedia Database Systems and Multimedia Information Systems," *IEEE Transactions on Knowledge and Data Engineering, to appear.*

[3] Chen, S.-C., Shyu, M.-L., and Zhang, C., "An Unsupervised Segmentation Framework For Texture Image Queries," *The 25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC),* Chicago, Illinois, USA, Oct. 2000.

[4] Chen, S.-C., Shyu, M.-L., and Zhang, C., "An Intelligent Framework for Spatio-Temporal Vehicle Tracking," *4th International IEEE Conference on Intelligent Transportation Systems,* Oakland, California, USA, Aug. 2001.

[5] Chen, S.-C., Shyu, M.-L., Zhang, C., and Kashyap, R., "Object Tracking and Augmented Transition Network for Video Indexing and Modeling," *12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000),* Vancouver, British Columbia, Canada, pp. 428-435.

[6] Chen, S.-C., Sista, S., Shyu, M.-L., and Kashyap, R., "An Indexing and Searching Structure for Multimedia Database Systems," *IS&T/SPIE conference on Storage and Retrieval for Media Databases 2000,* San Jose, CA, U.S.A, pp. 262-270.

[7] Chen, S.-C., Sista, S., Shyu, M.-L., and Kashyap, R., "Augmented Transition Networks as Video Browsing Models for Multimedia Databases and Multimedia Information Systems," *11th IEEE International Conference on Tools With Artificial Intelligence (ICTAI'99),* Chicago, IL, U.S.A., Nov. 1999.

[8] Coen M, "The Future of Human-Computer Interaction or How I Learned to Stop Worrying and Love my Intelligent Room," *IEEE Intelligent Systems*, vol. 14, no. 2, pp. 8-10, Mar, 1999.

[9] Courtney, J. D., "Automatic Video Indexing via Object Motion Analysis," *Pattern Recognition*, vol. 30, no. 4, pp. 607-625, 1997.

[10] Cucchiara, R., Piccardi, M., and Mello, P., "Image Analysis and Rule-based Reasoning for a Traffic

Monitoring System," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119-130, June 2000.

[11] Dailey, D. J., Cathey, F., and Pumrin, S., "An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras," *IEEE Transactions on Intelligent Transportations Systems*, vol. 1, no. 2, pp. 98-107, Jun, 2000.

[12] DeMenthon, D., Stuckelberg, M., and Doermann, D., "Image Distance using Hidden Markov Models," *International Conference Pattern Recognition (ICPR 2000): Image, Speech and Signal Processing,* Barcelona, Spain, pp. 147-150, Sept. 2000.

[13] Fan, L. and Sung, K. K., "Model-Based Varying Pose Face Detection and Facial Feature Registration in Video Images," *8th ACM International Conference on Multimedia,* Los Angeles, CA, pp. 295-302, Oct. 2000.

[14] Ferman, A. M., Guensel, B., and Tekalp, A. M., "Object-based Indexing of MPEG-4 Compressed Video," *in Proceedings of SPIE: Visual Communications and Image Processing,* San Jose; CA, pp. 953-963, Feb. 1997.

[15] Friedman, N. and Russell, S., "Image Segmentation in Video Sequences: A Probabilistic Approach," *Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI '97),* Providence; RI.

[16] Gonzalez, R. C. and Woods, R. E. *Digital image processing*, Reading, Mass: Addison-Wesley, 1993.

[17] Grimson, W. E. L., Stauffer, C., Romano, R., and Lee, L., "Using Adaptive Tracking to Classify and Monitor Activities in a Site," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Preceding,* pp. 22-31, 1998.

[18] Haritaoglu, I., Harwood, D., and Davis, L., "W 4 - Who, Where, When, What: A Real-Time System for Detecting and Tracking People," *IEEE Third International Conference on Face and Gesture Recognition,* Nara, Japan, pp. 222-227, 1998.

[19] Haritaoglu, I., Harwood, D., and Davis, L., "A Fast Background Scene Modeling and Maintenance for Outdoor Surveillance," *15th IEEE International Conference on Pattern Recognition: Applications, Robotics Systems and Architectures,* Barcelona, Spain, pp. 179-183, Sept. 2000.

[20] Huang, T., Koller, D., Malik, J., and Ogasawara, G., "Automatic Symbolic Traffic Scene Analysis Using Belief Networks," *Proceedings of the AAAI, 12th National Conference on Artificial Intelligence (AAAI '94),* Seattle, WA, pp. 966-972, July 1994.

[21] Kamijo, S., Matsushita, Y., Ikeuchi, K., and Sakauchi, M., "Traffic Monitoring and Accident Detection at Intersections," *IEEE International Conference on Intelligent Transportation Systems,* Tokyo Japan, pp. 703-708, Oct. 1999.

[22] Koller, D., Weber, J., and Malik, J., "Robust Multiple Car Tracking with Occlusion Reasoning," *3rd European Conference on Computer Vision, Eccv '94,* Stockholm Sweden, pp. 189-196, May 1994.

[23] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S., "Multi-Camera Multi-Person Tracking for EasyLiving," *3rd IEEE International Workshop on Visual Surveillance,* Dublin, Ireland, pp. 3-10, July 2000.

[24] Montgomery. Co. Department of Public Works Transportation. ATMS Video Monitoring System Live Traffic Camera Pictures, http://www.dpwt.com/jpgcap/camintro.html.

[25] Puzicha, J., Hofmann, T., and Buhmann, J. M., "Histogram Clustering for Unsupervised Image Segmentation," *IEEE Computer Society Conference Computer Vision and Pattern Recognition,* Fort Collins; CO, pp. 602-608, June 1999.

[26] Shafer, S., Krumm, J., Brumitt, B., Meyers, B., Czerwinski, M., and Robbins, D., "The New EasyLiving Project at Microsoft Research," *DARPA/NIST Workshop on Smart Spaces,* pp. 127-130, July 1998.

[27] *Sista,* S. and *Kashyap,* R. L., "Unsupervised Video Segmentation and Object Tracking," *Computers in Industry*, vol. 42, no. 2-3, pp. 127-146, June 2000.

[28] Stauffer, C. and Grimson, W. E. L., "Adaptive Background Mixture Models for Real-Time Tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 246-252, 1999.

[29] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B., "Wallflower: Principles and Practice of Background Maintenance," *7th International Conference on Computer Vision (ICCV'99),* Held on the Island of Crete, pp. 255-261, Sept. 1999.

[30] Yeo, B.-L. and Yeung, M. M., "Retrieving and Visualizing Video," *Communications of the ACM*, vol. 40, no. 12, pp. 43-52, Dec. 1997.