AUGMENTED TRANSITION NETWORKS AS VIDEO BROWSING MODELS FOR MULTIMEDIA DATABASES AND MULTIMEDIA INFORMATION SYSTEMS

Shu-Ching Chen

Florida International University
School of Computer Science
Miami, FL 33199
chens@cs.fiu.edu

Srinivas Sista, Mei-Ling Shyu, and R. L. Kashyap

Purdue University
School of Electrical and Computer Engineering
West Lafayette, IN 47907-1285
{*sista, shyu, kashyap*}*@ecn.purdue.edu*

Abstract

In an interactive multimedia information system, users should have the flexibility to browse and choose various scenarios they want to see. This means that two-way communications should be captured by the conceptual model. Digital video has gained increasing popularity in many multimedia applications. Instead of sequential access to the video contents, the structuring and modeling of video data so that users can quickly and easily browse and retrieve interesting materials becomes an important issue in designing multimedia information systems. An abstract semantic model called the augmented transition network (ATN), which can model video data and user interactions, is proposed in this paper. An ATN and its subnetworks can model video data based on different granularities such as scenes, shots and key frames. Multimedia input strings are used as inputs for ATNs. The details of how to use multimedia input strings to model video data are also discussed. Key frame selection is based on temporal and spatial relations of semantic objects in each shot. The temporal and spatial relations of semantic objects are captured from our proposed unsupervised video segmentation method which considers the problem of partitioning each frame as a joint estimation of the partition and class parameter variables. Unlike existing semantic models which only model multimedia presentation, multimedia database searching, or browsing, ATNs together with multimedia input strings can model these three in one framework.

Key words: *Augmented Transition Network (ATN), Multimedia Input String, Multimedia Browsing, Multimedia Database Systems.*

1. Introduction

Unlike traditional database systems which have text or numerical data, a multimedia database or information system may contain different media such as text, image, audio, and video. Video is popular in many applications such as education and training, video conferencing, video on demand, news service, and so on. Traditionally, when users want to search for certain content in videos, they need to fast forward or rewind to get a quick overview of interest on the video tape. This is a sequential process and users do not have a chance to choose or jump to a specific topic directly. How to organize video data and provide the visual content in compact forms becomes important in multimedia applications [19]. Therefore, users can browse a video sequence directly based on their interests so that they can get the necessary information faster and the amount of data transmission can be reduced. Also, users should have the opportunity to retrieve video materials by using database queries. Since video data contains rich semantic information, database queries should allow users to get high level content such as *scenes* or *shots* and low level content according to the temporal and spatial relations of semantic objects. A semantic object is an object appearing in a video frame such as a "car." Also, a semantic model should have the ability to model visual contents at different granularities so that users can quickly browse large video collections.

Many video browsing models propose to allow users to visualize video content based on user interactions [1, 5, 6, 10, 11, 14, 19]. These models choose representative images using regular time intervals, one image in each shot, all frames with focus key frame at specific place, and so on. Choosing key frames based on regular time intervals may miss some important segments and segments may have multiple key frames with similar contents. One image in each shot also may not capture the temporal and spatial relations of semantic objects. Showing all key frames may confuse users when too many key frames are displayed at the same time. To achieve a balance, we propose a key frame selection mechanism based on the number, temporal, and spatial changes of the semantic objects in the video frames.

The Augmented transition network (ATN), developed by Woods [18], has been used in natural language understanding systems and question answering systems for both text and speech. We use the ATN as a semantic

This work has been partially supported by National Science Foundation under contract IRI 9619812.

model to model multimedia presentations [2], multimedia database searching, the temporal, spatial, or spatio-temporal relations of various media streams and semantic objects [3, 12, 13]. As shown in [4], ATNs need fewer nodes and arcs to represent a multimedia presentation compared with Petri-net models such as OCPN [9]. Multimedia input strings adopt the notations from regular expressions [8] and are used to represent the presentation sequences of temporal media streams, spatio-temporal relations of semantic objects, and keyword compositions. In addition to using ATNs to model multimedia presentations and multimedia database searching, how to use ATNs and multimedia input strings as video browsing models is discussed in this paper. Moreover, key frame selection based on the temporal and spatial relations of semantic objects in each shot will be discussed. In previous studies, formulations and algorithms for multiscale image segmentation and unsupervised video segmentation and object tracking were introduced [15, 16, 17]. Our video segmentation method focuses on obtaining object level segmentation, i.e., obtaining objects in each frame and their traces across the frames. Hence, the temporal and spatial relations of semantic objects required in the proposed key frame selection mechanism can be captured. We apply our video segmentation method on a small portion of a soccer game video and use the temporal and spatial relations of semantic objects to illustrate how the key frame selection mechanism works. The details on how to use the recursive call property in ATNs to model user loops are also presented.

The organization of this paper is as follows. Section 2 discusses the use of ATNs and multimedia input strings to model video browsing. Key frame selection algorithm is introduced in section 3. Section 3 also gives an example soccer game video. Conclusions are presented in section 4.

2. Video Browsing Using ATNs

In an interactive multimedia information system, users should have the flexibility to browse and decide on various scenarios they want to see. This means that two-way communications should be captured by the conceptual model. Digital video has gained increasing popularity in many multimedia applications. Instead of sequential access to the video content, structuring and modeling video data so that users can quickly and easily browse and retrieve interesting materials becomes an important issue in designing multimedia information systems.

Browsing provides users the opportunity to view information rapidly since they can choose the content relevant to their needs. It is similar to the table of contents and the index of a book. The advantage is that

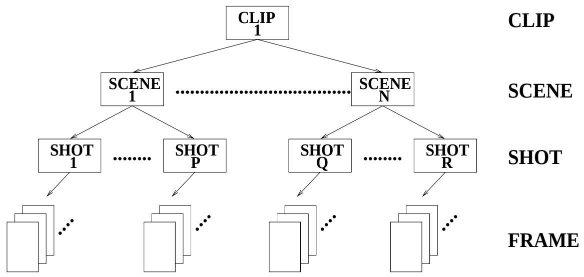


Figure 1: A hierarchy of video media stream

users can quickly locate the interesting topic and avoid the sequential and time-consuming process. In a digital video library, in order to provide this capability, a semantic model should allow users to navigate a video stream based on shots, scenes, or clips. The ATN can be used to model the spatio-temporal relations of multimedia presentations and multimedia database systems. It allows users to view part of a presentation by issuing database queries. In this paper, we further design a mechanism by using the ATN to model video browsing so that users can navigate the video contents. In this manner, querying and browsing capabilities can be provided by using ATNs.

2.1. Hierarchy for a Video Clip

As mentioned in [19], a video clip can be divided into *scenes*. A *scene* is a common event or locale which contains a sequential collection of *shots*. A *shot* is a basic unit of video production which captures between a record and a stop camera operation. Figure 1 is a hierarchy for a video clip. At the topmost level is the video clip. A clip contains several *scenes* at the second level and each *scene* contains several *shots*. Each *shot* contains some contiguous *frames* which are at the lowest level in the video hierarchy. Since a video clip may contain many video frames, it is not good for database retrieving and browsing. How to model a video clip, based on different granularities, to accommodate browsing, searching and retrieval at different levels is an important issue in multimedia database and information systems. A video hierarchy can be defined by the following three properties:

1. $V = \{S_1, S_2, \ldots, S_N\}$, S_i denotes the ith scene and N is the number of scenes in this video clip. Let $\mathrm{B}(S_1)$ and $\mathrm{E}(S_1)$ be the starting and ending times of scene S_1, respectively. The temporal relation $\mathrm{B}(S_1) < \mathrm{E}(S_1) < \mathrm{B}(S_2) < \mathrm{E}(S_2) < \ldots$ is preserved.

2. $S_i = \{T_1^i, \ldots, T_{n_i}^i\}$, T_j^i is the jth shot in scene S_i and n_i is the number of shots in S_i. Let $\mathrm{B}(T_1^i)$ and

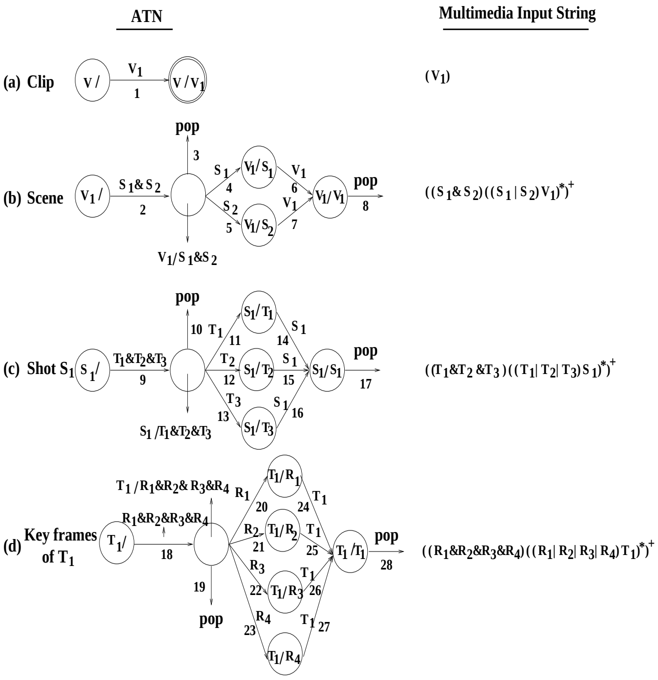


Figure 2: Augmented Transition Network for video browsing: (a) is the ATN network for a video clip which starts at the state V/. (b)-(d) are part of the subnetworks of (a). (b) is to model *scenes* in video clip V_1. (c) is to model *shots* in scene S_1. Key frames for shot T_1 is in (d).

E(T_1^i) be the starting and ending times of shot T_1^i where B(T_1^i) < E(T_1^i) < ... < B($T_{n_i}^i$) < E($T_{n_i}^i$).

3. $T_j^i = \{R_1^{i,j}, \ldots, R_{l_j}^{i,j}\}$, $R_1^{i,j}$ and $R_{l_j}^{i,j}$ are the starting and ending frames in shot T_j^i and l_j is the number of frames for shot T_j^i.

In property 1, V represents a video clip and contains one or more *scenes* denoted by S_1, S_2, and so on. *Scenes* follow a temporal order. For example, the ending time of S_1 is earlier than the starting time of S_2. As shown in property 2, each *scene* contains some *shots* such as T_1^i to $T_{n_i}^i$. *Shots* also follow a temporal order and there is no time overlap among shots so B(T_1^i) < E(T_1^i) < ... < B($T_{n_i}^i$) < E($T_{n_i}^i$). A *shot* contains some key frames to represent the visual contents and changes in each shot. In property 3, $R_k^{i,j}$ represents key frame k for shot T_j^i. The details of how to choose key frames based on temporal and spatial relations of semantic objects in each shot will be discussed in section 3.

2.2. Using ATNs to Model Video Browsing

An ATN can build up the hierarchy property by using its subnetworks. Figure 2 is an example of how to use an ATN and its subnetworks to represent a video hierarchy. An ATN and its subnetwork are capable of segmenting a video clip into different granularities and still preserve the temporal relations of different units.

In Figure 2(a), the arc label V_1 is the starting state name of its subnetwork in Figure 2(b). When the input symbol V_1 is read, the name of the state at the head of the arc (V/V_1) is pushed into the top of a push-down store. The control is then passed to the state named on the arc which is the subnetwork in Figure 2(b).

In Figure 2(b), when the input symbol X_1 (S_1&S_2) is read, two frames which represent two video scenes S_1 and S_2 are both displayed for the selections. In the original video sequence, S_1 appears earlier than S_2 since it has a smaller number. The "&" symbol in multimedia input strings is used to denote the concurrent display of S_1 and S_2. ATNs are capable of modeling user interactions where different selections will go to different states so that users have the opportunity to directly jump to the specific video unit that they want to see. In our design, vertical bars "|" in multimedia input strings and more than one outgoing arc in each state at ATNs are used to model the "or" condition so that user interactions are allowed. Assume S_1 is selected, the input symbol S_1 is read. Control is passed to the subnetwork in Figure 2(c) with starting state name $S_1/$. The "*" symbol indicates the selection is optional for the users since it may not be activated if users want to stop the browsing. The subnetwork for S_2 is omitted for the simplicity.

In Figure 2(c), when the input symbol T_1&T_2&T_3 is read, three frames T_1, T_2, and T_3 which represent three shots of scene S_1 are displayed for the selection. If the shot T_1 is selected, the control will be passed to the subnetwork in Figure 2(d) based on the arc symbol $T_1/$. The same as in Figure 2(b), temporal flow is maintained.

3. Key Frame Selections Based on Temporal and Spatial Analysis of Video Sequences

The next level under *shots* are key frames. Key frame selections play an important role to let users examine the key changes in each video shot. Since each shot may still have too many video frames, it is reasonable to use key frames to represent the *shots*. The easiest way of key frame selection is to choose the first frame of the shot. However, this method may miss some important temporal and spatial changes in each shot. The second way is to include all video frames as key frames and this may have computational and storage problems, and may increase users' perception burdens. The third way is to choose key frames based on fixed durations. This method is still not a good mechanism since it may give us many key frames with similar contents. Therefore, how to select key frames to represent a video *shot* is an important issue for digital library browsing, searching, and retrieval [20]. To achieve a balance, we propose a key frame selection mechanism 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 in this study. Therefore, spatio-temporal changes in each shot can be represented by these key frames. For example, in each shot of a soccer game, players may change positions in subsequent frames and the number of players appearing may change at the time duration of the shot.

3.1. Simultaneous Partition and Class Parameter Estimation (SPCPE) Algorithm

Let the set of semantic objects in the kth frame ($R_k^{i,j}$) of the jth shot T_j^i in the ith scene S_i be denoted by $O_k^{i,j}$. We define the key frame selections as follows:

Definition 1: Given two contiguous video frames $R_a^{i,j}$ and $R_b^{i,j}$ in T_j^i, let the sets of the semantic objects in these two video frames be $O_a^{i,j}$ and $O_b^{i,j}$. $R_b^{i,j}$ is a key frame if and only if any of following two conditions is satisfied:

(1) $O_a^{i,j} \cap O_b^{i,j} \neq O_a^{i,j} \cup O_b^{i,j}$;
(2) Any semantic object spatial location changes between $O_a^{i,j}$ and $O_b^{i,j}$.

As mentioned previously, the video segmentation method can provide the required information for the key frame selection mechanism. Therefore, the video segmentation method is applied to each frame before the above two conditions are checked. The method for partitioning a video frame starts with an arbitrary partition and employs an iterative algorithm to estimate the partition and the class description parameters jointly. So the minimum we obtain through our descent method depends strongly on the starting point or the initial partition. In a video, the successive frames do not differ much due to the high temporal sampling rate. Hence the partitions of adjacent frames do not differ significantly. Starting with the estimated partition of the previous frame, if we apply our descent algorithm on the current frame we may obtain a new partition that is not significantly different from the partition of the previous frame. For the first frame, since there is no previous frame, we use a randomly generated initial partition.

We treat the partition as well as the class parameters as random variables and pose the problem as one in joint estimation[15, 16]. Suppose we have 2 classes. Let the partition variable be $\boldsymbol{c} = \{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ and the classes be parametrized by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$. Now, the MAP estimates of $\boldsymbol{c} = \{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ are given by

$$\begin{aligned}(\hat{\boldsymbol{c}}, \hat{\boldsymbol{\theta}}) &= \operatorname{Arg}\max_{(\boldsymbol{c},\boldsymbol{\theta})} P(\boldsymbol{c}, \boldsymbol{\theta} \mid Y) \\ &= \operatorname{Arg}\max_{(\boldsymbol{c},\boldsymbol{\theta})} P(Y \mid \boldsymbol{c}, \boldsymbol{\theta}) P(\boldsymbol{c}, \boldsymbol{\theta}). \end{aligned} \tag{1}$$

With appropriate assumptions, this joint estimation can be simplified to the following form:

$$\begin{aligned}(\hat{\boldsymbol{c}}, \hat{\boldsymbol{\theta}}) &= \operatorname{Arg}\min_{(\boldsymbol{c},\boldsymbol{\theta})} J(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \\ J(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \sum_{y_{ij} \in \boldsymbol{c}_1} - \ln p_1(y_{ij}; \boldsymbol{\theta}_1) \\ &+ \sum_{y_{ij} \in \boldsymbol{c}_2} - \ln p_2(y_{ij}; \boldsymbol{\theta}_2). \end{aligned} \tag{2}$$

The joint estimation method is called the *simultaneous partition and class parameter estimation (SPCPE)* algorithm. The algorithm starts with an arbitrary partition of the data and computes the corresponding class parameters. Using 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. The details of the video segmentation method are shown in [17].

Given a video shot T_j^i, let K_j^i be the set of key frames selected for T_j^i and m a frame index. Initially the first frame is always selected so $K_j^i = \{R_1^{i,j}\}$.

1. Initialization:
   - $K_j^i = \{R_1^{i,j}\}$;
   - Execute SPCPE algorithm for the first frame;
2. for $m = 2$ to l_j
   - Execute SPCPE algorithm to get the temporal and spatial relations of the semantic objects;
   - if (($O_m^{i,j} \cap O_{m-1}^{i,j} \neq O_m^{i,j} \cup O_{m-1}^{i,j}$) OR Spatial_location_change($O_m^{i,j}$, $O_{m-1}^{i,j}$)) then $K_j^i = K_j^i \cup R_m^{i,j}$;

   endfor.

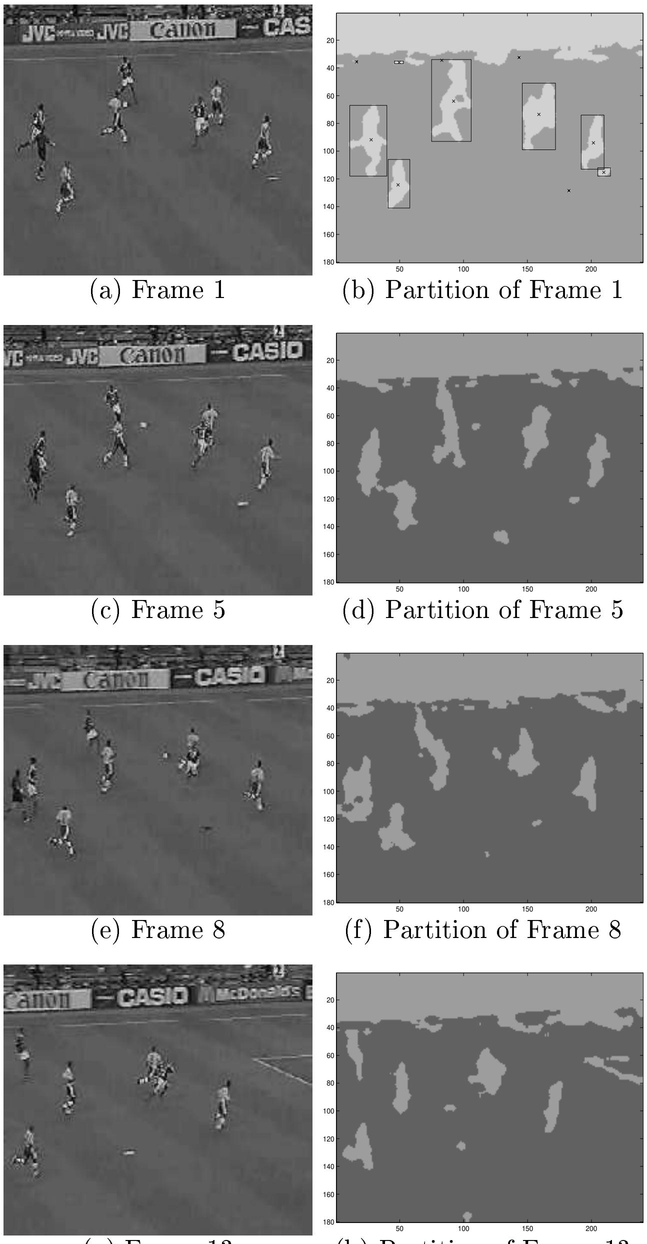
The first condition of definition 1 models the number of semantic object changes in two contiguous video frames at the same shot. The first part of the if-statement in the above solution algorithm is used to check this situation. The latter part of the if-statement checks the second condition of definition 1, which is to model the temporal and spatial changes of semantic objects in two contiguous video frames of the shot. Using the same definition of three dimensional relative positions for semantic objects as shown in [3], we choose one semantic object to be the target semantic object in each video frame. We adopt the minimal bounding rectangle (MBR) concept in R-tree [7] so that each semantic object is covered by a rectangle. In order to distinguish the relative positions, twenty-seven numbers are used to distinguish the relative positions of each semantic object relative to the target semantic object and are represented by subscripted numbers. The centroid point of each semantic object is used for space reasoning so that any semantic object is mapped to a point object. Therefore, the relative position between the target semantic object and a semantic object can be derived from these centroid points.

3.2. An Example Soccer Video

The example soccer video consists of 60 frames. It is a gray scale video that shows the part of the game where a goal is scored. Each frame is of size 180 rows and 240 columns. A small portion of the soccer video game is used to illustrate the way the proposed key frame selection mechanism works. Although we have several distinct regions in each frame of the video, only two of them are important from the content based retrieval perspective, namely the ball and the players. There are some important aspects in this video that make automatic object tracking difficult. They are as follows:

- The soccer ball vanishes between players for a few frames and reappears later.
- The regions corresponding to the players merge together and separate out after a few frames.
- Some spurious patches, typically on the ground, suddenly appear as blobs and disappear giving the impression of an object.

We will apply our video segmentation method to this data, assuming that there are 2 classes. The first frame is partitioned using the multiscale frame segmentation with 2 classes. The algorithm is initialized with a random starting partition. After obtaining the partition of the first frame, we compute the partitions of the subsequent frames. From the results on frames 1 through 60, a few frames – 1, 5, 8 and 13 – are shown in Figure 3 along with the original frames adjacent to them. As can be seen from Figure 3, the players, the soccer ball and the sign boards in the background (JVC, Canon, etc.) are all captured by a single class. The ground in the soccer field is captured by another class. Some of the players who are close together have been combined into a single segment. Similarly, the ball is merged into a single segment with two other players. For example, in frame 1, the ball and two players are part of one segment; whereas by the fifth



(a) Frame 1 (b) Partition of Frame 1

(c) Frame 5 (d) Partition of Frame 5

(e) Frame 8 (f) Partition of Frame 8

(g) Frame 13 (h) Partition of Frame 13

Figure 3: Figures (a),(c),(e),(g) are the original Frames 1,5,8,13 (on the left) and (b),(d),(f),(h) show their corresponding partitions (on the right). (b) shows the segments extracted from the first frame of the Soccer video. The centroid of each segment is marked with an 'x' and the segment is shown with a bounding box around it. The segments corresponding to the moving players and the ball are captured in every frame automatically.

frame, the soccer ball is far away so that it becomes a segment in itself. This continues until it goes in between two other players. Notice the patch on the ground which was near the right most player in the first frame, moves to the left uniformly owing to the camera panning to the right. In frame 5 we can see a spurious patch appearing out of nowhere. On the whole, the initial conditions from the previous frames seem to be guiding the segmentation of the current frame in an effective manner. There are some artifacts on the ground, specifically the one closest to the rightmost player, which show up as patches in the final partition. Inspection of the other frames shows that it is also present in them and not something spurious.

The segments of Frame 1, extracted by applying the seeding and region growing method are shown in Figure 3(b). There are 15 segments in this frame out of which only 5 correspond to the players and the ball. The ball and 2 players are merged into one segment, and there are 2 other segments where two players are put into a single segment. The rest of the two segments consist of one player in each segment. We have implemented the programs to find the bounding boxes and the centroids for the segments. Therefore, the segments are displayed with the bounding boxes around them and the centroids are marked with an 'x' in Figure 3(b). The small segments with only a centroid and without any apparent bounding box are the ones with very few pixels. Most of them are on the top of the frame and at the bottom of the sign boards. They arise out of smoothing the broken soccer boundary line.

Since only the ball and the players are important from the content based retrieval perspective, we use Figure 4 to simplify the segments for each frame. As shown in Figure 4, the ground (**G**) is selected as the target semantic object and the segments are denoted by **P** for the players or **B** for the soccer ball. As mentioned earlier, if two semantic objects are too close to each other, they are merged into a single segment. Hence, the soccer ball is put into a single segment only when it is far away from the players (in Frames 5 and 8) and each segment **P** may consist of multiple players and/or the soccer ball. In this example, each frame is divided into nine subregions. More or fewer subregions in a video frame may be used to allow more fuzzy or more precise queries as necessary. The corresponding multimedia input strings are on the right of Figure 4. In our design, each key frame is represented by an input symbol in a multimedia input string and the "&" symbol between two semantic objects is used to denote that the semantic objects appear in the same frame. The subscripted numbers are used to distinguish the relative positions of the semantic objects relative to the target semantic object "ground". Table 1 shows part of the three dimensional spatial relations

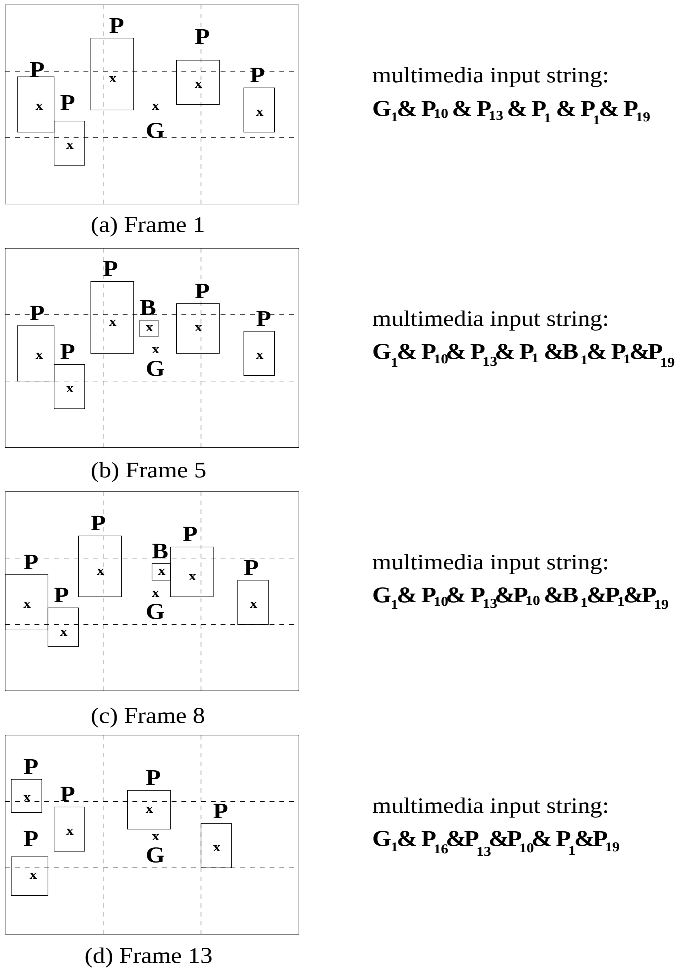


Figure 4: Segments with bounding boxes and centroids for Frames 1,5,8,13 in Figure 3 on the left and their corresponding multimedia input strings on the right. Each segment is displayed with the bounding box around it and the centroid is marked with an 'x'. Here, **G**, **P**, and **B** represent "ground", "players", and "soccer ball", respectively. The "ground" (**G**) is selected as the target semantic object and the subscripted numbers in a multimedia input string are used to distinguish the relative positions of the semantic objects relative to **G**. Each frame is divided into nine subregions and the centroid of each segment is used as a reference point for spatial reasoning.

Table 1: Part of the three dimensional relative positions for semantic objects: The first and the third columns indicate the relative position numbers while the second and the fourth columns are the relative coordinates. (x_t, y_t, z_t) and (x_s, y_s, z_s) represent the X-, Y-, and Z-coordinates of the target and any semantic object, respectively. The "$\approx$" symbol means the difference between two coordinates is within a threshold value.

Number	Relative Coordinates
1	$x_s \approx x_t, y_s \approx y_t, z_s \approx z_t$
10	$x_s < x_t, y_s \approx y_t, z_s \approx z_t$
13	$x_s < x_t, y_s < y_t, z_s \approx z_t$
16	$x_s < x_t, y_s > y_t, z_s \approx z_t$
19	$x_s > x_t, y_s \approx y_t, z_s \approx z_t$

introduced in [3]. (x_t, y_t, z_t) and (x_s, y_s, z_s) represent the X-, Y-, and Z-coordinates of the target and any semantic object, respectively. The "$\approx$" symbol means the difference between two coordinates is within a threshold value. Since two dimensions are considered in this example, $z_s \approx z_t$. The multimedia input strings can be used for multimedia database searching via substring matching. The details of multimedia database searching are shown in [3].

Assume Figures 4(a), (b), (c), and (d) are four key frames for shot T_1. The multimedia input string to represent these four key frames is as follows:
Multimedia input string:

$$\underbrace{(G_1\&P_{10}\&P_{13}\&P_1\&P_1\&P_{19})}_{M_1}\underbrace{(G_1\&P_{10}\&P_{13}\&P_1\&B_1\&P_1\&P_{19})}_{M_2}$$
$$\underbrace{(G_1\&P_{10}\&P_{13}\&P_{10}\&B_1\&P_1\&P_{19})}_{M_3}\underbrace{(G_1\&P_{16}\&P_{13}\&P_{10}\&P_1\&P_{19})}_{M_4}$$

As shown in the above multimedia input string, there are four input symbols which are M_1, M_2, M_3, and M_4. The appearance sequence of the semantic objects in an input symbol is based on the spatial locations of the semantic objects in the video frame from left to right and top to bottom. For example, Figure 4(a) is represented by input symbol M_1. G_1 indicates that **G** is the target semantic object. P_{10} means the first **P** is on the left of **G**, P_{13} means the second **P** is below and to the left of **G**, P_1 means the third **P** and the fourth **P** are at the same subregion as **G**, and P_{19} means the fifth **P** is on the right of **G**. Figure 4(b) is modeled by input symbol M_2 in which the soccer ball **B** appears at the same subregion as **G** and the rest of the semantic objects remain at the same locations. In this case, the number of semantic objects is increased from six to seven. This is an example to show how to use a multimedia input string to represent a number of semantic object changes. Figure 4(c) is represented by input symbol M_3. The third **P** moves from the same subregion of **G** to above and left of **G** so the associated number changes from 1 to 10 from which the relative spatial relations can also be modeled by the multimedia input string. Input symbol M_4 models Figure 4(d). In this situation, **B** disappears and the first **P** changes its spatial location from the left to above and left of **G** in Figure 4(c). So, the number associated with the first **P** changes from 10 to 16 and **B** does not exist in M_4. The order of these four key frames is modeled by four input symbols concatenated together to indicate that M_1 appears earlier than M_2 and so on.

4. Conclusions

Video data are widely used in today's multimedia applications such as education, video on demand, video conferencing and so on. Managing video data so that users can quickly browse video data is an important issue for the multimedia applications using video data. A good semantic model is needed if we want to meet the needs. In this paper, ATNs are used to model video hierarchy for browsing. Based on this design, users can view information quickly to decide whether the content is what they want to see. Key frames selection based on temporal and spatial relations of semantic objects is used in our design. The temporal and spatial relations of semantic objects are captured by the proposed unsupervised video segmentation method. From the soccer game video example, we can see that the players and the soccer ball are captured well. Since the first frame uses a random initialization and the subsequent frames use the results of the previous frames, the method is completely unsupervised. In addition, by incorporating the partition information of the previous frame into the segmentation process of the current frame, the temporal information is implicitly used. Under this design, these key frames preserve many of the visual contents and minimize the data size to mitigate the computation and storage problems in multimedia browsing environments. Moreover, based on the results of the segmentation, multimedia input strings are constructed. The multimedia input strings can be used for multimedia database searching via substring matching. Unlike the existing semantic models which only model presentation, query, or browsing, our ATN model provides these three capabilities in one framework.

5. References

[1] F. Arman, R. Depommer, A. Hsu, and M.Y. Chiu, "Content-based browsing of video sequences," *ACM*

Multimedia 94, pp. 97-103, Aug. 1994.

[2] Shu-Ching Chen and R. L. Kashyap, "Temporal and Spatial Semantic Models for Multimedia Presentations," in 1997 International Symposium on Multimedia Information Processing, pp. 441-446, Dec. 11-13, 1997,

[3] Shu-Ching Chen and R. L. Kashyap, "A Spatio-Temporal Semantic Model for Multimedia Presentations and Multimedia Database Systems," accepted for publication *IEEE Transactions on Knowledge and Data Engineering*, 1999.

[4] Shu-Ching Chen and R. L. Kashyap, "Empirical Studies of Multimedia Semantic Models for Multimedia Presentations," in 13th International Conference on Computer and Their Applications, pp. 226-229, March 25-27, 1998.

[5] Y. F. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-Oriented Concept Modeling of Video Data," IEEE Int'l Conference on Data Engineering, pp. 401-408, March 1995.

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *IEEE Computer*, Vol. 28, No. 9, pp. 23-31, September 1995.

[7] A. Guttman, "R-tree: A Dynamic Index Structure for Spatial Search," in Proc. ACM SIGMOD, pp. 47-57, June 1984.

[8] S.C. Kleene, "Representation of Events in Nerve Nets and Finite Automata, Automata Studies," *Princeton University Press*, Princeton, N.J., pp. 3-41, 1956.

[9] T.D.C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," *IEEE J. Selected Areas in Commun.*, Vol. 9, pp. 413-427, Apr. 1990.

[10] M. Mills, J. Cohen, and Y.Y. Wong, "A magnifier tool for video data," in Proc. ACM Computer Human Interface (CHI), May, 1992, pp. 93-98.

[11] E. Oomoto, and K. Tanaka, "OVID: Design and Implementation of a Video Object Database System," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 629-643, August 1993.

[12] Mei-Ling Shyu, Shu-Ching Chen, and R. L. Kashyap, "Information Retrieval Using Markov Model Mediators in Multimedia Database Systems," 1998 International Symposium on Multimedia Information Processing, pp. 237-242, Dec. 14-16, 1998.

[13] Mei-Ling Shyu and Shu-Ching Chen, "Probabilistic Networks for Data Warehouses and Multimedia Information Systems," submitted to *IEEE Trans. on Knowledge and Data Engineering*.

[14] S.W. Smoliar and H.J. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia*, pp. 62-72, Summer, 1994.

[15] R. L. Kashyap and S. Sista, "Unsupervised Classification and Choice of Classes: Bayesian Approach," Technical Report TR-ECE 98-12, School of Electrical and Computer Engineering, Purdue University, July 1998.

[16] S. Sista and R. L. Kashyap, "Bayesian Estimation for Multiscale Image Segmentation," IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing, Phoenix, Arizona, March 1999.

[17] S. Sista and R. L. Kashyap, "Unsupervised video segmentation and object tracking," to appear in IEEE Int'l Conf. on Image Processing, 1999.

[18] W. Woods, "Transition Network Grammars for Natural Language Analysis," *Comm. of the ACM*, **13**, October 1970, pp. 591-602.

[19] B-L Yeo and M.M. Yeung, "Retrieving and Visualization Video," *Comm. of the ACM*, Vol. 40, No. 12, December 1997, pp. 43-52.

[20] M.M. Yeung and B. Liu, "Efficient Matching and Clustering of Video Shots," in IEEE International Conference on Image Processing, Vol I, October, 1995, pp. 338-341.