A Spatial User Interface for Browsing Video Key Frames

Mahendren A/L Munisamy[*]
Department of Social Informatics,
Graduate School of Informatics, Kyoto University,
Yoshida Honmachi, Sakyo,
Kyoto 606-8501, Japan
mahen@dl.kuis.kyoto-u.ac.jp

Kazutoshi Sumiya[†]
Department of Social Informatics,
Graduate School of Informatics, Kyoto University,
Yoshida Honmachi, Sakyo,
Kyoto 606-8501, Japan
sumiya@i.kyoto-u.ac.jp

Katsumi Tanaka[‡]
Department of Social Informatics,
Graduate School of Informatics, Kyoto University,
Yoshida Honmachi, Sakyo,
Kyoto 606-8501, Japan
ktanaka@i.kyoto-u.ac.jp

ABSTRACT

Key frames provide a visual summary for a user to grasp the contents of a video without having to view it. The most common method to extract keyframes from a video stream is known as scene change detection. We argue that this method is inappropriate when extracting keyframes from video images of sports footage such as a soccer game. This is because the inter-frame differences are very small and hard to detect in such video data. In this paper, we introduce a method to dynamically extract keyframes from video images of a soccer game. The keyframes were extracted based on the ball's positional information and changes in the video frame's pixel level intensity. We also propose a method to visualize the positional changes between keyframes using an animation sequence. Finally by using this animation sequence we developed a method to browse soccer video images after segmenting it into units we call plays.

Keywords

video database, keyframe extraction, keyframe browsing

1. INTRODUCTION

Within a video image, the frames that represent a particular shot are called keyframes. The method most commonly used to automatically extract keyframes from a video file is known as scene change detection. Here, the pixel intensity levels of two consecutive frames are compared and when the difference of the corresponding values exceeds a certain threshold, a scene change is declared. Therefore, whenever there is a sudden change in the contents of two consecutive frames (this usually occurs during camera breaks or when the overall color of a frame changes drastically), the latter frame is considered to be a keyframe.

In the case of sports footage, it may be difficult to detect scene changes within a video image. As an example, video images of soccer games capture a wide area on camera. The players belonging to the same team wear the same uniforms and the background color (the color of the soccer pitch) is constantly the same. It can be said that scene changes almost never occur in such cases. Therefore, it is inappropriate to use scene change detection to extract keyframes from such video images.

Another problem that occurs when extracting keyframes for soccer is that there exist a variation in the way people interpret the contents of a soccer game. A defensive minded person will give preference to defensive plays while a striker will give preference to offensive plays. It is therefore important to accommodate these variations in user preferences when extracting keyframes.

In this paper we propose a method to dynamically extract key frames using the positional information of an object within a video image. The method we propose takes into consideration variations that might exist in a particular user's positional interests. We also propose a method to browse soccer video images by mapping the keyframes on to a three dimensional representation of a soccer field. The keyframes are shown consecutively in the same position while the background is moved. This enables the user to visualize the positional changes that occur throughout the video stream.

In Section 3 of this paper presents a survey of existing technologies when segmenting video data. Section 4 describes the algorithm we developed in order to extract keyframes using an object's positional information and pixel intensity. In Section 5 we propose a method to visualize the positional

[*] PhD candidate, Graduate School of Informatics, Kyoto University

[†] Associate Professor, Graduate School of Informatics, Kyoto University

[‡] D. Eng. Professor, Graduate School of Informatics, Kyoto University

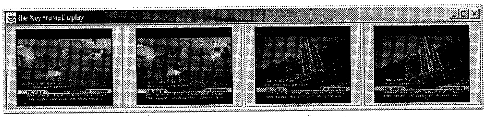


Figure 1: Example of abrupt scene change

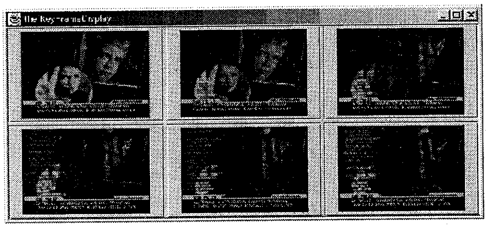


Figure 2: Example of gradual scene change

changes when browsing keyframes that have been extracted using the algorithm in Section 4. Section 6 describes works that are related to this paper. Finally in Section 7, we summarize the main contributions of this paper.

2. BASIC IDEAS

2.1 Key frame extraction

Content-based temporal sampling of video data is performed in order to identify significant video frames to achieve better representation, indexing, storage and retrieval of the video data. Automatic content-based temporal sampling is very difficult because the sampling criteria are not well defined, that is, whether a video frame is important or not is usually subjective. Moreover, it is usually highly application-dependent and requires high-level, semantic interpretation of the video contents.

The process of analyzing the visual contents of a video image and partitioning it into a set of basic units called shots is known as video data segmentation. Content-based sampling can thus be approximated by selecting one representing frame from each shot, since a shot is defined as a continuous sequence of video frames that have no significant inter-frame difference in terms of their visual contents. A single shot usually results from a single continuous camera operation. This partitioning is usually achieved by sequentially measuring inter-frame differences and studying their variances, for example, detecting sharp peaks. This process is known as scene change detection.

Scene change in a video can either be abrupt or gradual. Abrupt scene changes result from editing cuts. The process of detecting abrupt scene changes is known as cut detection. Gradual scene changes result from chromatic edits, spatial edits and combined edits. They include special effects like zoom, camera pan, dissolve and fade in/out. Examples of an abrupt scene change and a gradual scene change are shown in Figure 1 and Figure 2 respectively. Scene change detection is usually based on some measurements of the image frame, which can be computed from the information contained in the images.

It can therefore be said that present keyframe extraction methods are just a side effect of video cut detection. In other words, keyframes are the result of performing shot decompositions on a video file. Furthermore, present keyframe extraction methods only take a video image's signal information into consideration and ignore any form of metadata.

The actual purpose of extracting keyframes from a video file is to provide video content representation. It would then be possible to grasp the contents without having to view the entire video file. However by only using scene change detection methods, it is impossible to obtain keyframes that provide an adequate summary for certain types of video files like sports footage. As an example, in a soccer video file, players belonging to the same team wear the same uniforms and the background color (the color of the soccer pitch) is constantly the same. Therefore, these types of video images have very little significant inter-frame difference in terms of their visual contents and very few keyframes are extracted to represent the contents.

In order to overcome these problems, we developed a method to extract keyframes using a video's metadata and signal information. This increases the number of frames that are extracted and provides a better summary of the entire video file.

2.2 Browsing keyframes

Pesently, browsing a set of keyframes usually means viewing those keyframes in a tiled form on a monitor. Though it may be possible to grasp the sequence at which the keyframes were extracted, it is difficult to grasp the positions of each keyframe in relation to the entire video image. Furthermore, when browsing a series of tiled keyframes, confusion can occur whle trying to interpret an object's movements. As an example, Figure 4 shows a series of keyframes that were extracted from a video file of a soccer game. Though the sequence at which the keyframes were extracted may be obvious, it is unclear which part of the field each frame represents.

We therefore propose a method to browse keyframes using an animation sequence. Here, each keyframe is shown consecutively on a static position on the monitor. A three dimensional image of the keyframe's surroundings is moved accordingly to show the positional changes that occur to each keyframe that is being displayed. By using such an animation sequence, the user is able to grasp both the sequence and position of each keyframe in relation to the entire soccer file.

The extraction method introduced in the previous section and the browsing method mentioned here present a new way to extract and browse keyframes. These methods can also be considered as the first step to developing a new form of technology in order to automatically construct web data for easily understandable video summaries.

3. AUTOMATIC VIDEO SCENE ANALYSIS AND SEGMENTATION

3.1 Pixel-Level Change Detection

The changes between two frames can be detected by comparing the differences in intensity values of corresponding pixels in the two frames. The algorithm counts the number of pixels changed and a scene change is declared if the percentage of the total number of pixels changed exceeds a certain threshold. The difference in pixels and threshold calculation can be represented in the following equations.

$$DP_i(x,y) = \begin{cases} 1 & |F_i(x,y) - F_{i+1}(x,y)| > t \\ 0 & otherwise \end{cases} \qquad (1)$$

$$\frac{\sum_{x,y=1}^{X,Y} DP_i(x,y)}{X \times Y} \times 100 > T \qquad (2)$$

In equation(1), $F_i(x,y)$ is the intensity value of the pixel in frame i at the coordinates (x,y). If the difference between the corresponding pixels in the two consecutive frames is above a certain minimum intensity value t, then the difference picture, $DP_i(x,y)$ is set to one. In equation (2) the percentage difference between the pixels in the two frames is calculated by summing the difference picture and dividing by the total number of pixels in a frame. If this percentage is above a certain threshold T, a camera break is declared.

3.2 Likelihood Ration

In the likelihood ration approach, the frames are subdivided into blocks and then the blocks are compared on the basis of the statistical characteristics of their intensity levels. Equation (3) represents the formula that calculates the likelihood function. Let μ_i and μ_{i+1} be the mean intensity values for a given region in two consecutive frames and σ_i and σ_{i+1} be the corresponding variances. The number of the blocks that exceed a certain threshold t are counted. If the number of blocks exceed a certain value (dependent on the number of blocks) a segment is declared. A subset of the blocks can be used to detect the difference between the images so as to expedite the process of block matching.

$$\lambda = \frac{\left(\frac{\sigma_i+\sigma_{i+1}}{2} + \left(\frac{\mu_i-\mu_{i+1}}{2}\right)^2\right)^2}{\sigma_i \times \sigma_{i+1}} \qquad (3)$$

$$DP_i(k,l) = \begin{cases} 1 & if \lambda > t \\ 0 & otherwise \end{cases}$$

This method increases the tolerance against noise associated with camera and object movement. It is possible that even though the two corresponding blocks are different they can have the same density function. In such cases no change is detected.

3.3 Histogram Comparison

The sensitivity to camera and object motion can be further reduced by comparing the gray level histograms of the two consecutive frames. This is due to the fact that two frames with not much difference in their background and some amount of object motion have almost the same histograms. The histogram is obtained from the number of pixels belonging to each gray level in the frame.

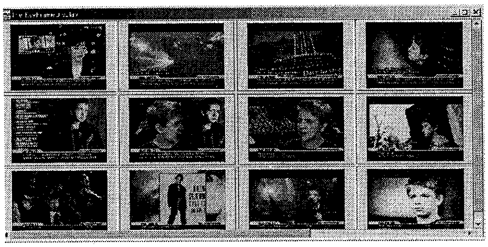


Figure 3: Example of keyframe extraction using scene change detection on a news scene

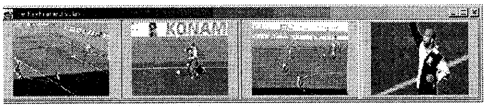


Figure 4: Key frame extraction using conventional scene change detection methods on a soccer video file

$$\sum_{j=1}^{G} |H_i(j) - H_{i+1}(j)| > t \qquad (4)$$

In Equation (4) the number of gray levels, j is the gray value, i is the frame number and $H_i(j)$ is the histogram for the gray level j. If the sum is greater than the given threshold t then a transition is declared.

4. KEYFRAME EXTRACTION USING POSITIONAL INFORMATION

4.1 Motivating Example

At present, the method most commonly used to extract keyframes from a video file is by using the scene change detection technique mentioned in Section 2.1. Figure 3 shows an example of key frame extraction using scene change detection on a news reel lasting 2 minutes 39 seconds long. We used a software called Giga Video Recorder (Version 1.1.1) developed by Sony Cooperation to extract key frames from the video stream. In this case, the contents of the entire video stream has been determined before it is captured. Therefore each scene change that occurred due to camera breaks or drastic overall color changes represents a meaningful semantic unit of the video file. A user is therefore able to grasp the contents of the entire video file by just viewing these keyframes.

However, during a live sports telecast, the events that are being captured on camera are occurring dynamically. Therefore, it is inappropriate to conclude that a scene change due to a camera break is semantically significant. Furthermore,

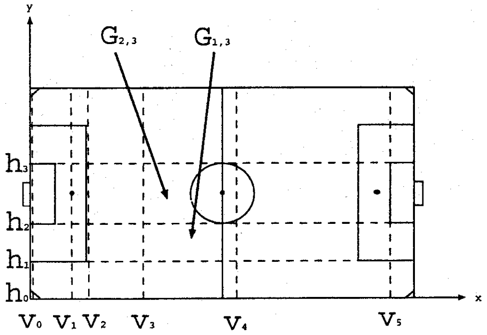


Figure 5: Example of a grid

in a sport like soccer, players will be wearing uniforms of the same color and the background color (color of the soccer pitch) is constantly the same. It is therefore difficult to detect any scene changes due to changes in pixel intensity. We used Sony Cooperation's Giga Video Recorder (Version 1.1.1) to extract keyframes over a 24 second soccer video stream. As shown in Figure 4, only four keyframes were extracted due to scene changes. Since it is difficult to detect color changes in a soccer video file the key frames in Figure 4 were extracted after a camera break occurs. It is impossible for a user to grasp the contents of this stream by only viewing the keyframes that have been extracted using this method. In other words, the keyframes that have been extracted through this method do not provide an adequate summary of this video stream.

One of the most important characteristics of a soccer game is the positional changes that occurs to the ball throughout the course of the game. However, present scene change detection techniques do not utilized a video object's spatial information when extracting key frames from a video file. We therefore propose a method to extract keyframes using both the positional information of a video object and a video image's metadata. The next section describes the algorithm we developed in order to extract keyframes in this manner.

4.2 Algorithm

We developed the following algorithm to extract keyframes from video images of a soccer game for a single play. This algorithm persumes that the ball's positional information within the video data has been given. Here we use Definition 1 to define a single unit of play.

Definition 1. A single unit of play is the duration during which the ball is brought into play after the whistle is blown until the time it leaves the perimeter of the field or until play is stopped due to a whistle blow.

As shown in Figure 5, the user partitions an image of the soccer field into multiple rectangular regions, which we will call "grids" here. The grids are constructed by drawing a series of horizontal lines $h_0, h_1 \ldots h_n$ and vertical lines $v_0, v_1 \ldots v_m$ which cut through the field. Then the grid $G_{i,j}$ is the rectangular region bounded by $h_i, h_{i+1}, v_j, v_{j+1}$ The size of each grid $G_{i,j}$ and the number of lines used to make the grids may vary as the user desires. Therefore, when the ball's trajectory crosses into a new grid, one of the subscripts of $G_{i,j}$ increases or decreases by one. For example, in Figure 5 if the ball is in grid $G_{2,3}$ and it crosses the horizontal line h_2 then the corresponding frame is in grid $G_{1,3}$.

1. The first frame of the video data is extracted and considered to be a keyframe.
2. Using the given positional information of the soccer ball within the video image, the system configures the trajectory of the soccer ball for each single unit of play.
3. Using Equation (5) the average pixel intensity level G_i for frame i is calculated.

$$G_i = \frac{\sum_{x,y=1}^{X,Y} F_i(x,y)}{X \times Y} \qquad (5)$$

   Here $F_i(x,y)$ is the intensity value of the pixel in frame i at the coordinates (x,y).
4. Step(3) is repeated for all frames within a particular grid.
5. The difference between the average pixel intensity levels of two consecutive frames T_i within a particular grid is calclulated using the following equation.

$$T_i = G_i - G_{i+1} \qquad (6)$$

6. All the values of $T_i$are compared. When T_i reaches it's highest value, frame $i+1$ is extracted and considered to be a keyframe.

4.3 Results

Figure 6 shows the result of positional keyframe extraction using the grid structure in Figure 5. The same video stream which was used to extract keyframes in the Section 4.1 is used again. By allowing the user to partition the soccer field we successfully extracted the keyframes dynamically based on a particular user's positional interests.

A total of seven keyframes were extracted using this method while the scene change detection method was only able to extract four keyframes. We used changes in a soccer ball's positional information along with changes in pixel level intensity as the basis for extracting keyframes from the video file. This is a better representation of the contents of a soccer game, as opposed to scene changes due to camera breaks or abrupt changes in pixel intensity only. Therefore, it can be concluded that a user will be able to have a better grasp of the contents of a soccer video stream by viewing the keyframes which have been extracted through this method.

4.4 Integrating Scene Change Detection and Positional Information

One of the disadvantages of keyframe extraction using positional information and pixel intensity changes occurs when

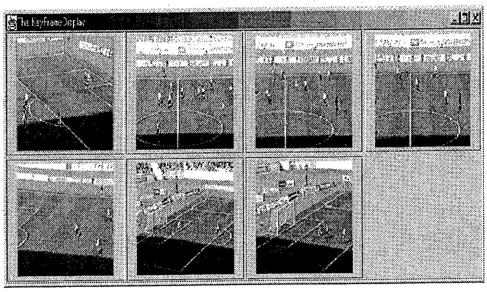


Figure 6: Key frame extraction using positional information

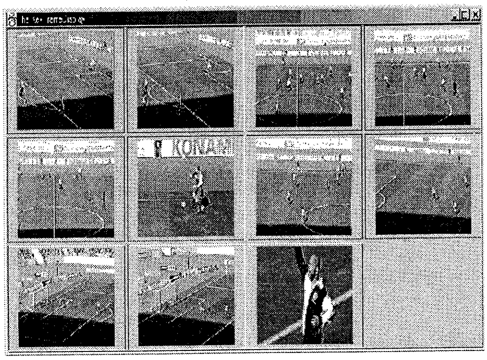
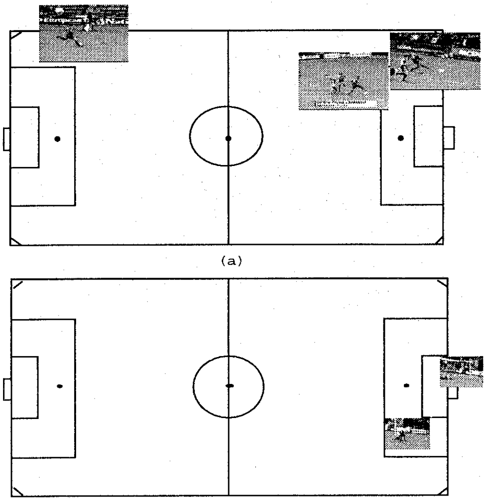


Figure 7: Integrating keyframe extraction using positional information and conventional scene change detection methods

the ball is not present within a certain portion of the captured image. This usually occurs when there is a close up scene of a player or referee. In this case, since the video object (the soccer ball) is not visible it is impossible to annotate any positional information. Therefore no keyframes are extracted during this time. We therefore suggest that the algorithm we developed in the Section 4.2 be used in combination with conventional scene change detection methods. This will cause the keyframes in Figure 4 to be integrated into Figure 6. Figure 7 shows the final set of keyframes that represent the entire play. By combining these two methods, we're able to increase a user's understanding of the semantic content of the video file.

5. BROWSING KEYFRAMES

In the browsing stage, representations that are good high level overviews of the content of the video data should be displayed. A user by looking at such representations should be able to quickly understand the video content and browse through many videos in a short period of time. Moreover, the user should have the capability to view the video data in a nonlinear fashion and be able to get an overview of the entire video image. The development of visual summaries representing video depends heavily on the category of the video [7]. Different categories of video will require different forms of visual summaries. In this section we look into the problems of browsing keyframes that have been extracted using the algorithm in Section 4.2. Finally, we introduce a solution to overcome these problems by using an animation sequence to represent the positional changes between keyframes.



(a)

(b)

Figure 8: Graphical index

5.1 Problems While Browsing Key Frames

The conventional method to browse a set of keyframes is by viewing a tiled arrangement as shown in Figure 7. Here, the keyframes are arranged in a left to right, top to bottom order. This allows the user to grasp the sequence at which each keyframe was extracted. Key frames that have been displayed in this way are suitable for video data like sticoms, news and movies that have a story sturcture. Programs of sporting events however, do not have a story structure. This makes it difficult for a user to grasp the contents of a sporting event by browsing keyframes that have been displayed in a conventional manner.

We used positional information to effectively extract keyframes from a soccer video stream. Therefore, to help visualize the positional variations of each keyframe we developed a graphical index (as shown in Figure 8). However, as shown in Figure 8(a), since the size of each key frame is too big, it is difficult to determine the exact position of the ball in reference to the soccer field. In order to solve this problem, we reduced the size of each keyframe as shown in Figure 8(b), but this makes it difficult to view the contents of each keyframe.

5.2 Integrating 3D Graphics While Browsing

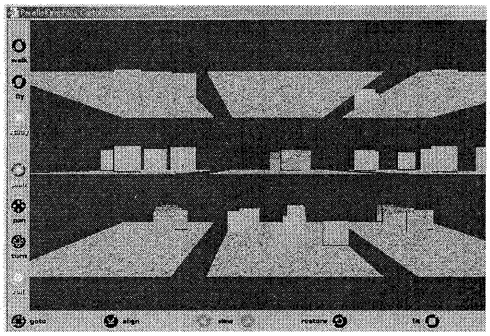


Figure 9: Initial 3D-graphical index

5.2.1 *Browsing keyframes*

The positional attributes of the soccer ball within a video frame are also compounded with other attributes like height, camera angle and camera motion. While browsing, we simulate these extra attributes by integrating three dimensional graphics. Figure 9 shows the initial attempt at implementing a three dimensional interface to browse keyframes using VRML. Here, we only considered the ball's positional information (x and y axes) when placing the keyframes onto the field. Each segment represents a single play as defined in Definition 1

As shown in Figure 10, we enhanced the prototype in Figure 9 by positioning and rotating the three dimensional soccer ground while maintaining the keyframe in a static position. By doing this, we are able to give users a perspective view of each keyframe in reference to the entire soccer field.

5.2.2 *Visualizing Positional Changes Between Key Frames*

After an initial keyframe has been selected, another keyframe is selected using a pointing device. As shown in Figure 11 an animation sequence that shows the positional changes between the first and second keyframes that have been selected is displayed. The keyframes are displayed together with appropriately sampled non keyframe images using the three dimensional index in Figure 10 In our case study (involving a 24 second soccer video file) we were able to obtain satisfying results by displaying 5 temporally sampled non keyframes between two keyframe images. If there are less than 5 frames between two keyframes, the three dimensional indices for those keyframes are displayed cosecutively.

When displaying the animation sequence, each frame image is displayed consecutively in the same static position. We rotate and position the three dimensional representation of the soccer ground to give a perspective view of where each frame is located in reference to the entire field.

We can use Definition 1 for a single unit of play as a natural segment for a team ball sport. It is then possible to use the animation sequence in Figure 11 to browse a video file of an entire soccer game. Instead of having to choose two separate keyframes to display an animation sequence, the system automatically displays the animation sequence between the first and final frames of each play. This can be considered as a possible alternative to browse soccer video images over the internet.

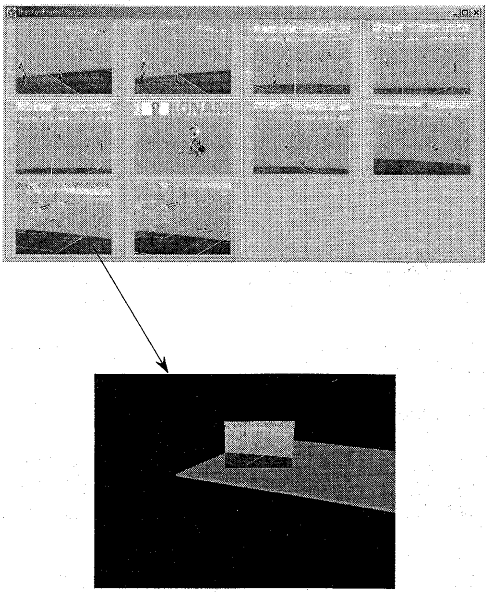


Figure 10: Example of intergrating 3D-graphics while browsing keyframes

6. RELATED WORKS

It would be ideal to be able to automatically track the soccer ball using image processing techniques. However, according to studies conducted by Intile and Bobick [6], sudden rapid changes in ball movement violate the smooth motion assumption of tracking algorithms. Additionally, accurate motion estimates are difficult to obtain because they are compounded with camera motion. In order to overcome this when tracking football players in a closed world, contextual knowledge is applied in the "football domain". Here, a closed world is defined as a space-time region of an image sequence in which the complete taxonomy of objects is known and in which each pixel should be explained as belonging to one of those objects. Given contextual object information, context specific features can be dynamically selected as the basis for tracking. A context specific feature is one that has been chosen based upon the context to maximize the chance of successful tracking between frames.

Choi et. al [4] attempt automatic soccer game analysis as follows:

1. The ground is extracted in order to track players, find lines and make mosaic images for model transforma-

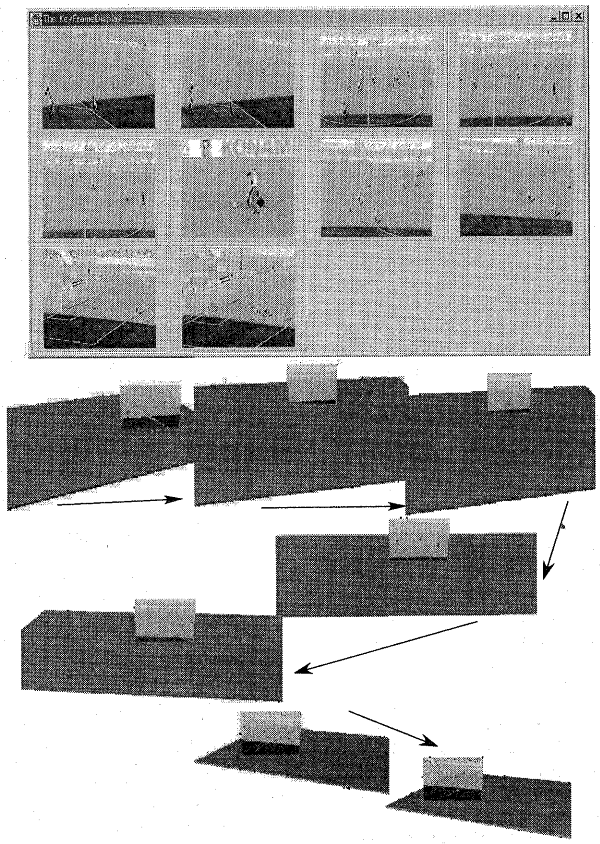


Figure 11: Example of animation sequence to visualize positional changes between selected keyframes

tion. A color histogram information under the assumption that the region of the ground is green and occupies most areas in an image used.

2. Each player and the ball is identified and tracked in the image sequence. Players move in a non-rigid manner, frequently collide into each other and are occluded by others. Template matching and Kalman filtering is applied for player tracking.

3. The absolute position of the players is computed. However, since there are small number of features in a soccer sequence, it is not easy to find the absolute location of players on the field. A field model is constructed and image to model transformations are computed to attack the problem. When the center circle of the field model is found in an image, the transformation is computed by the locations of four feature points from the image of the center circle. Otherwise, an image mosaicking technique is used to find out the image to model transformation. Then the trajectory of each player is computed.

Yow et. al [5] argue that a majority of the "exciting" shots are captured in relatively short durations compared to the length of the game. It is important to be able to analyze the contents and detect highlights of a soccer video. Therefore techniques to automatically detect and extract the essence of the games by content analysis of the video and to effectively present the field and motion paths of the players and ball via panoramic reconstruction of selected highlights are presented.

7. CONCLUSION AND FUTURE WORKS

In this paper, we first showed that video images of soccer games have very little significant inter-frame difference in terms of their visual contents. For this reason, using conventional keyframe extraction methods cause very few keyframes to to be extracted. We therefore developed an algorithm to extract keyframes by using an object's positional information and by detecting changes in pixel level intensity. This increases the number of frames that are extracted and provides a better summary of the entire video file. We conclude that the keyframes that were extracted in this manner provide a better summary of the contents of a soccer game as opposed to present conventional methods.

We also introduced a method to browse keyframes using an animation sequence. Here, each keyframe is shown consecutively on a static position on the monitor. A three dimensional image of the keyframe's surroundings is moved accordingly to show the positional changes that occur to each keyframe that is being displayed. By using such an animation sequence, the user is able to grasp both the sequence and position of each keyframe in relation to the entire soccer field. We are now investigating the feasibility of our system in other sports like baseball and volleyball.

In most sports footage "exciting" shots can be identified by analyzing the video's sound intensity. In these sort of video images sound intensity levels will rise when an interesting play occurs. We are therefore investigating the posibility of including sound intensity levels into consideration when extracting keyframes.

The algorithm in Section 4.2 is meant to dynamically extract keyframes from a video stream that is meant for television broadcast. This form of broadcast is limited to a single video and audio stream. A webcast however, may be comprised of multiple video and audio streams along with associated hypertext documents. The number of streams may also be dynamic. For example during a televised webcast of a soccer game, one video stream may be initially used when no significant plays are occurring. However, another stream showing a different portion of the pitch may be added to enhance the understanding of a players position in relation to the ball. The present system should be extended in order to be able to extract keyframes from multiple video streams.

In video, the data objects developed have to accommodate the various information modalities such as text, audio and visual content. Text-based search could be a first step when searching a video. Many multimedia data objects are modeled with the use of text to describe the attributes of various information modalities. These text descriptions should also describe the video content sufficiently to help users to locate segments of interest. However due to a limitation in the vocabulary, it is difficult to express the contents of a soccer video file by only using text descriptions. For example, a user who wants to view a pass scene may not be able to find any satisfactory results because of the variations that exists when defining a pass. We therefore suggest that the algorithm in Section 4.2 be used together with text based searches as a a filter to help increase search accuracy of a video file.

Acknowledgments

This research was partly supported by the Ministry of Education, Culture, Sports, Science and Technology's Scientific Research Fund Foundation(B)(2) under the project "Accessing Personalized Broadcast Content and Services Using Personal Digital Recorders" (project no: 14380177)

8. REFERENCES

[1] M.Petkovic and W.Jonker "A Framework for Video Modelling", Center for Telematics and Information Technology, University of Twente.

[2] G. Ahanger and T.D.C Little. "A Survey of Technologies for Parsing and Indexing Digital Video", Journal of Visual Communication and Image Representation, 7(1):28-43, March 1996.

[3] Ahmed K. Elmagarmid, Haitao Jiang, Abdelsalam A. Helal, Anupam Joshi and Magdy Ahmed. "Video Database Systems, Kluwer Academic Publishers.

[4] Sunghoon Choi, Yongduek Seo, Hyunwoo Kim and Ki-Sang Hong. "Where are the ball and players?: Soccer Game Analysis with Color-based Tracking and Image Mosaick", Dept of EE, Pohang University of Science and Technology.

[5] Dennis Yow, Boon-Lock Yeo, Minerva Yeung and Bede Liu. "Analysis And Presentation Of Soccer Highlights From Digital Video" Proceedings of the Second Asian Conference on Computer Vision, Singapore, December 1995, IEEE.

[6] S. Intille, A. Bobick. "Visual Tracking Using Closed-Worlds", M.I.T Media Laboratory Perceptual Computing Section Technical Report, No. 294, November 1994.

[7] R.M.Bolle, B.-L. Yeo and M.M. Yeung. "Video query: Research Directions", IBM Journal of Research and Development.

[8] Luara Lemay and Rogers Cadenhead. "Teach Yourself Java 2 in 21 days", Sams: A division of Macmillan Computer Publishing.