AUTOMATIC VIDEO ZOOMING FOR SPORT TEAM VIDEO BROADCASTING ON SMART PHONE

Fabien Lavigne
TCTS laboratory, University of Mons, Belgium
Fabien.Lavigne@fpms.ac.be

Fan Chen Xavier Desurmont
Tele laboratory, Universite catholique de Louvain, Belgium
fan.chen@uclouvain.be
Image department, Multitel, Mons, Belgium
desurmont@multitel.be

Keywords: sport team video broadcasting, clip segmentation, view type detection, ball detection, region of interest focus

Abstract: This paper presents a general framework to adapt the size of a sport team video extracted from TV to a small device screen. We use a soccer game context to describe the four main steps of our video processing framework: (1) A view type detector specifies if the current frame of the video has to be resized or not. (2) If the camera point of view is far, a ball detector localizes the interesting area of the scene. (3) Then, the current frame is resized and centered on the ball, taking into account some parameters, such as the ball position and its speed. (4) At the end of the process, the score banner is detected and removed by an inpainting method.

1 INTRODUCTION

Last years, the number of applications developed on smart phones increased dramatically and more and more multimedia content is proposed on these devices. At the same time, a lot of topics emerged in video processing, especially in sport events broadcasting context, as reviewed by (Yu and Farin, 2005). For example, (Yu et al., 2009) presented a camera calibration method in order to insert 3D virtual content in a tennis game scene and (Takahashi et al., 2005) proposed a method to create summaries of soccer videos.

However, before adding virtual information or summarizing a sport game for a mobile phone, it is necessary to adapt the content, usually extracted from the TV broadcasting, because of the small size of a screen phone. In this context (Knoche et al., 2007), realized two studies on 84 participants in order to evaluate the effect of zoom on human perception. They extracted from their statistics a function computing the optimal zoom for an extra large shot, depending on the target display size. While it provides the optimal size of the zoom, it does not locate the region of interest where the zoom has to be optimized. In same way, (Ariki et al., 2006) built a system based on recognition rules, which chooses the best zoom value and locates it on the region of interest of the scene. Their method is interesting, but the new camera point of view generated by their algorithm does not evolve seamlessly.

In this paper, we propose a general framework in order to adapt a sport team TV video to a small device screen. A complete implementation of each block of the method is detailed in the context of a soccer game filmed by a multi pan-tilt-zoom camera system. The first section presents briefly an overview of the system and defines some notations used in this paper. The second section explains a clip segmentation and classification method. The results of this process are used to decide if it is necessary to apply our zooming algorithm on a given clip. The third section describes the ball detection process, implemented in order to locate the region of interest of the scene. The fourth section focuses on estimating the zooming frame parameters which generate new camera point of view. The fifth section shows an additional process which detects the score banner and removes it using inpainting to improve the rendering of the final video. The sixth section shows some results and the last one concludes and discusses on the future perspectives.
Figure 1: Overview of the system. The process (a) splits the video into a set of clips. Each clip is described by the camera point of view (far, medium, close) and whether it is a replay or not. The two processes (b) and (c) can be ran in parallel. The process (b) detects the ball and estimates parameters of the zoom (position and factor). The process (c) detects the banner and applies an inpainting algorithm to remove it. The last process (d) generates a new video from the inpainting video and the zoom parameters.

2 PRINCIPLES AND NOTATIONS

This part presents an overview of the system (see Fig.1) and defines some notations used in the paper. The system is divided into four main processes. The first one extracts a set of clips from the whole video. It also detects whether a clip is a replay or not and classifies the camera point of view into three categories: far, medium, and close. If the camera point of view is far and the clip is not a replay we applied the two following processes in parallel.

In a second process, we compute, for each frame of the original video, the parameters of a zooming frame (see Fig.2). The first parameter is the zoom factor $Z_i$. The size of the zooming frame is computed with the following expression $S_i = (1 - Z_i) \cdot S$, $Z_i \in [Z_{min}, Z_{max}], Z_{min}, Z_{max} \in [0, 1]$. $S$ is the size of the original frame and $S_i$ is the size of the zooming frame. The zoom factor is initialized at a defined value $Z_0$. The second parameter is the position of the zooming frame $C_i = (C_{ix}, C_{iy})$. We detect the ball position $B_i = (B_{ix}, B_{iy})$ in the current frame $i$ to locate the region of interest of the scene. The ball position is used to estimate the new position of the zooming frame. As we will discuss in the fourth part, we use also the frame rate $FPS$ ($frame/s$) and the speed of the ball $V_i = B_{i+1} - B_i$ as parameters of the process.

In the third process, we detect the banner where the score and time are displayed and we remove it with an inpainting method. At the end, we use the inpainted frame and the parameters of the zooming frame to generate the new video adapted to small screens.

Figure 2: Inside the black border, the original frame. Inside the white border, the zooming frame.
3 CLIP SEGMENTATION AND VIEW TYPE CLASSIFICATION

Compared to general videos, sport team videos usually have well-organized structures of shots, based on several elemental view types of cameraworks. For each shot, the cameraman can either give a far view for describing the complexity of the team sports, show more details of the action in a local area with a medium view, or zoom into a close-up view for enhancing the emotional involvement of the audience. Furthermore, sudden view switching during the evolving of a tight game action is suppressed in order to avoid the distraction of audiences attention from the current game. Hence, by first dividing the video into a sequence of clip shots and dealing with each clip with different strategies according to its view type, we obtain a semantically reasonable and computationally efficient base for further processing. Especially, the preprocessing for clip segmentation and view-type classification has three major subtasks, as shown in the workflow in Fig.3.

1) Detection of Replay. The two major methods for detecting replays are detection of replay-logos, and detection of slow-motions (Pan et al., 2001). Although replay-logos are producer-specific, we follow this approach, because we think that it is easier and more accurate to detect replay logos than to detect slow-motions, due to the fact that the view angle in the replay might change a lot from the normal play.

2) Detection of Shot-boundary. A difficult problem in shot boundary detection is to deal with special effects supporting smooth transition between two scenes, e.g., fade-in fade-out. Using histogram as features, we notice that histogram is gradually varying along with this smooth scene switching, as shown in Fig.4. Hence, shot-boundary detectors based on difference of histograms between two successive frames, as proposed in (Delannay et al., 2003), are not efficient in this case. Therefore, we propose an improved shot-boundary detector based on the difference between the average histogram of its left and right neighborhoods.

3) Detection of View-type. A simple but efficient method for view-type classification has been proposed by (Ekin et al., 2003). For scenes having a large portion of grass area, the non-grass blobs within the grass area reflects objects in the soccer field. The basic idea in (Ekin et al., 2003) is to evaluate the ratio of grass area to non grass area in each subdivision of the scene to identify the view type. Scenes with few or even without grass region could be either a public view or a game view. All public views will be omitted in our work. A game view without grass area usually gives a quite close view of the scene, even though it is a medium view, e.g., a scene focusing on the foot actions of players. Therefore, it is safe to treat all these scenes without grass area as close-up views. Based on the method in (Ekin et al., 2003), we further pre-classify the scene type before classification according to the percentage of overall grass ratio, and use support vector machine to replace the linear classifier for better classification performance, as shown in Fig.5. Extra robustness is achieved by running the view-type classification over all frames within the shot and making the final decision by taking a majority vote.

The results of this process are then used to decide if the following zooming process is applied. Indeed, we applied the zooming algorithm only if the camera view type is far and there is no replay. Moreover, if a shot-boundary is detected, the parameters of the zooming frame are reinitialized.
4 BALL DETECTION

In a soccer game, the ball represents the central element of the scene. Indeed, the players react according to the ball position. Consequently, detecting the ball in the video allows us to focus on the region of interest of the scene. This problematic is discussed in various studies as in (Yu et al., 2003) wherein the best ball candidate is detected with a modified version of the directional circle Hough transform and validated with a neural network classifier. In our case, we rely on a generic image processing chain. This chain is composed of five blocks including filtering, segmentation, objects labelization, objects description and objects recognition.

We use a structural representation of the scene to specify how to implement and choose input parameters of each operator of the process chain. A soccer game is represented by a set of entities (ball, players, bleachers ...) and each entity is described by a set of features (shape, color...). The existing spatial relations between these entities are also described (the ball is on the soccer field, the bleachers border the soccer field...). This problem formulation allows us to elaborate the following ball detection chain:

**Preprocessing:**
We apply a classic Gaussian filter to smooth the original image.

**Segmentation:**
Using the color components, we select the non-green pixels to separate the soccer field to other entities. Then we apply a morphological operator of closure to delete small artifacts and to merge sparse clusters of pixels belonging to the same object (the players’ legs for example).

**Object description:**
We build a label map from the segmented image and extract features of each object necessary in the recognition process (size, shape, bounding box ...).

**Object recognition:**
At this step, we have to select the ball from a set of objects. Bleachers and players are removed using their size and white lines using their shape. We delete artifacts (non-existing entities) using the ratio between width and height of their bounding box and their color.

This method is simple and efficient but problematic when the ball is connected to the players.
5 ZOOMING FRAME COMPUTATION

The aim of our system is to adapt a video of a soccer game to a small screen highlighting the interesting part of the scene. Therefore, we compute a new point of view when the camera is far from the ground. We define a zooming frame from which the new point of view is generated. This frame has to be centered on the ball and close enough to the ground to allow the user to understand what happens in the scene. Both aspects are controlled by parameters of position and zoom factor of the zooming frame. Key elements to compute these zooming parameters are the followings:

**Ball position:**
The ball position indicates where the action takes place. This value, given by the algorithm detailed in section 4, is used to compute the position of the zooming frame.

**Ball speed:**
The ball speed is used to update the zoom factor. If the movement of the ball is small between two consecutive frames, the zoom factor can be high because of the slow evolution of the area of interest position. It happens when the ball and camera do not move but also when the camera follows the ball trajectory at the same speed. If the ball moves fast, the zoom has to be small because it is easier for the zooming frame to follow the action with a small zoom.

**Frame rate:**
The higher is the frame rate, the slower zooming parameters have to evolve between two consecutive frames. Taking into account the frame rate allows us to process videos with different frame rates.

**Resolution of the target screen:** We use the ratio between resolution of the input video and resolution of the target screen to initialize the minimal zoom value $Z_{\text{min}}$, the maximal zoom value $Z_{\text{max}}$ and the initial zoom value $Z_0$.

**Visual effect:**
Our system has to provide a pleasant video to watch. Therefore, we have to avoid visual artifacts such as jumping of the zooming frame position and zooming and dezooming effects. We added into the formula terms to smooth the zooming frame trajectory and the zoom variation.

**System errors:**
It is necessary to consider errors the system can generate. If no ball is detected at the current frame, it can result from a non detection or the ball might not be present in the scene. In this case, the zoom value decreases to avoid zooming on a non interesting area and ball position is defined as the center of the original image. It is not possible to know whether the system sends a false alarm or gives a wrong ball position. But in this case, we noticed that ball position varies extensively between consecutive frames. Consequently this case is taken into account according to the speed of the ball. A high speed decreases the zoom value.

Taking into account all these parameters, we com-
pute the zoom factor and the zooming frame position with the following expressions:

\[ Z_i = Z_{i-1} + \frac{(1 - \alpha)f(i) + \alpha(Z_{i-1} - Z_{i-2})}{FPS} \]  

\[ C_i = C_{i-1} + \frac{1}{\beta n} \sum_{k=i-n+1}^{i} (B_k - B_{k-1}) \]  

\[ f(i) = \begin{cases} 
\rho_1, & \text{if } \theta_i \text{ and } V_i < s, \ (\rho_1 \in [0, 0.1]) \\
\rho_2, & \text{if } \theta_i \text{ and } V_i \geq s, \ (\rho_2 \in [-0.1, 0]) \\
\rho_3, & \text{otherwise, } \ (\rho_3 \in [-0.1, 0]) 
\end{cases} \]

\( \theta_i \) is true if a ball has been detected by the system. \( s \) is a threshold for the speed of the ball \( V_i \). The terms \( \alpha \) and \( \beta \) are smoothing factors, the higher these factors are, the smoother the evolution of the zooming frame is. The ball position \( B_i \) is defined as the center of the original video if no ball is detected. Some defined thresholds limit the values of the zoom factor and the zoom position: \( \forall i \in \mathbb{N}^*, Z_i \in [Z_{\min}, Z_{\max}] \) and \( \forall i \in \mathbb{N}^*, |C_i - C_{i-1}| < D_s \). It is also obvious that the zooming frame cannot be out of the original frame.

### 6 ADDITIONNAL PROCESSES

In a sportive video, other graphics elements, as a banner with score and time of the game, are incrusted in the initial video. If we apply directly our algorithm of zooming, the position and the size of such element become inconsistent. That is why we add another process which detects and removes the score banner. It is also allows new functionalities for the system. After deleting the banner, we create a small banner video and the user can decide when he wants to display it on the screen. He can also choose its position and its size on the screen.

The banner containing score and time of the game is always displayed at the same position on the screen and its size does not vary. This banner can be divided in two parts: the one where the graphical elements can change (time and score) and the one with no change (border). To detect the presence of the banner, we create a mask with part of the banner which never changes. Using this mask, we detect the banner using a simple frame differencing approach. When the banner is detected, we remove it using an inpainting algorithm developed in the GREYCstoration library and published in (Tschumperlé, 2006).

![Figure 10: Incoherent resized image. In the top left corner, a piece of the banner can be seen.](image10.png)

![Figure 11: (a) The original score banner. (b) After detection the banner is removed by inpainting.](image11.png)

![Figure 12: The rectified incoherent image.](image12.png)

### 7 RESULTS

Our proposed method has been tested on a 2 hours soccer video with a 25fps frame rate. The resolution of the original video is 800x600 and the resolution of the target screen is 480x360. We used the following configuration: \( \beta = 15, \alpha = 0.3, \rho_1 = 0.03, \rho_2 = -0.03, \rho_3 = -0.03 \) and \( n = 10 \). Some results of this sequence are shown in Fig.13. We realized a visual subjective evaluation comparing the video generated by our system and a video created by resizing the original to the resolution of the target screen. The new video is more pleasant and it is easier to understand what happens in the scene.

However, as we shown on Fig.14, some improvements are needed. Indeed, some situations are not optimal for our algorithm. The first one is when the ball is alone and no player follows it (the ball goes in touch
for example). In this case, following the players could be more interesting than following the ball. But this situation happens rarely and does not last a long time. The second one is more problematic, it happens when the ball moves slowly during few times and then a player makes suddenly a big shot. It is difficult with a smooth motion to follow the ball when its speed changes suddenly. In this case, several frames are necessary to refocus the zooming frame on the ball.

8 CONCLUSION AND PERSPECTIVES

This paper presented a general framework to adapt the size of a sport team video extracted from the TV to a small device screen. This framework is divided into four main processes:

- The division of the original video into clips where each clip is described by its camera point of view and whether it is a replay or not.
- The estimation of the zooming frame parameters using a ball detection algorithm.
- The banner detection following by an inpainting process.
- The rendering of the new video using the inpainted frame and the zooming frame parameters.

Based on this framework, we implemented a specific application for soccer game videos filmed by a multi-ptz camera network. The results are visually interesting and show the efficiency of our method. New ways have to be explored in order to improve our system. First of all, we could detect the players and use the results to make a compromise between zooming on the ball and including most of the players in the zooming frame. It also could be very interesting in a near future to make a subjective evaluation of our system with a method similar to the one the International Telecommunication Union (ITU) proposed in the report (ITU-R, 2009). Such evaluation will allow us to measure the impact of our system on human perception but also to select optimal input parameters. To conclude, we are about to integrate this system in a real-time streaming server/client architecture based on the one proposed by (Bomcke and Vleeschouwer, 2009).

ACKNOWLEDGEMENTS

This work has been funded by the Wallon region in the framework of WALCOMO and 3DME-DIA projects.

REFERENCES


Figure 13: Left column contains images from the original video. Middle column contains the results of the ball detection and the zooming frame. Right column contains the resulting images.

Figure 14: Two problematic situations for our algorithm. (a) The ball is followed but the region of interest is not centered on the ball. (b) The ball is lost because of a sudden big shot.