AUTOMATIC VIDEO ZOOMING FOR SPORT TEAM VIDEO BROADCASTING ON SMART PHONES

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

Fan Chen
Tele laboratory, Universite catholique de Louvain, Belgium
fan.chen@uclouvain.be

Xavier Desurmont
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 helps to decide whether 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 centred 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 TV broadcasting, because of the small size of a phone screen. In this context, (Knoche et al., 2007) conducted 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 (ROI) where the zoom has to be applied. In a same way, (Ariki et al., 2006) built a system based on recognition rules, which chooses the best zoom value and locates it on the ROI of the scene. Their method is interesting, but the new camera point of view generated by their algorithm does not evolve seamlessly. In other words, the camera point of view changes to follow ball and players but does not follow action as a pan-tilt zoom camera could do.

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 remaining of the paper is organized as follows. Section 2 briefly presents an overview of the system and defines some notations used in the paper. Section 3 explains the clip segmentation and classification methods. Results of these processes are used to decide if it is necessary to apply the zooming algorithm on a given clip. Section 4 describes the ball detection process, implemented in order to locate the ROI of the scene. Section 5 focuses on estimating zooming frame parameters from which a new camera point of view is generated. Section 6 shows an additional process which detects the score banner and removes it using inpainting in order to improve the final video rendering. Section 7 shows some qualitative results while section 8 concludes and discusses on future perspectives.

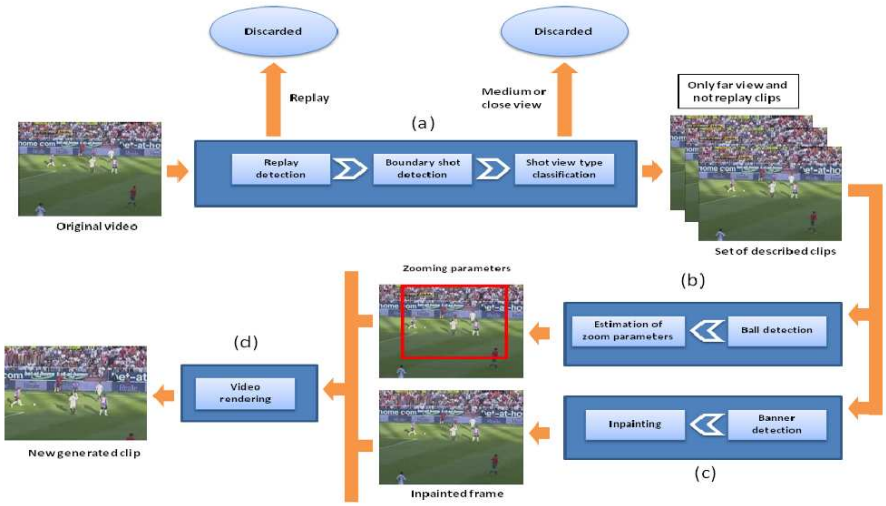


Figure 1: Overview of the system. Process (a) splits video into a set of clips. Each clip is described by its camera point of view (far, medium, close) and whether it is a replay or not. Processes (b) and (c) can be ran in parallel. Process (b) detects the ball and estimates parameters of zoom (position and factor). Process (c) detects the banner and applies an inpainting algorithm to remove it. The last process (d) generates a new video from inpainting video and zoomming 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, 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 $\hat{S}_i = (1 - Z_i).S_i$, $Z_i \in [Z_{min}, Z_{max}]$, $Z_{min}, Z_{max} \in [0,1[$. S_i is the size of the original frame and $\hat{S}_i$ is the size of the zooming frame. The zoom factor is initialized at a defined value Z_0. The second parameter is the zooming frame position $C_i = (C_{ix}, C_{iy})$. We detect the ball position $B_i = (B_{ix}, B_{iy})$ in the current frame i to locate the ROI of the scene. The ball position is used to estimate the new position of the zooming frame. As we will discuss in the section 5, we use also the frame rate FPS ($frame/s$) and the ball motion $V_i = B_{i-1} - B_i$ as parameters of the process. When zooming parameters are estimated, we detect the banner where score and time of game are displayed and we remove it using an inpainting method. At last, we use inpainted frame and zooming frame parameters to generate a new video adapted to a small screen.

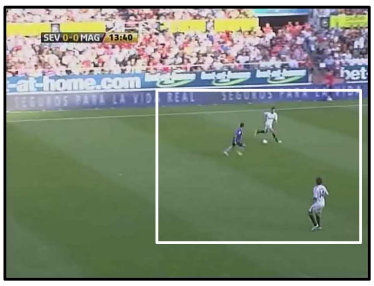


Figure 2: Inside the black border, the original frame. Inside the white border, the zooming frame.

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 audience 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.1.

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.3. 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.

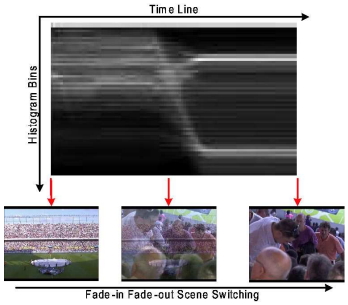


Figure 3: Varying Histogram during Smooth Shot Transition.

3) Detection of View-type. A simple but efficient method for view-type classification in soccer context 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. In order to locate grass regions we convert RGB images to HSL and only pick up pixels within the following color range (H: 40 - 120 (of 0 - 255) - S: 7 - 170 (of 0 - 255) - L: 25 - 180 (of 0 - 255), which covers green colors with different brightness, and is robust against illumination and shadow. Based on the results, we further use closing operator to connect neighbouring parts and then use labelling to filter out small blocks. 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 preclassify 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.4. 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.

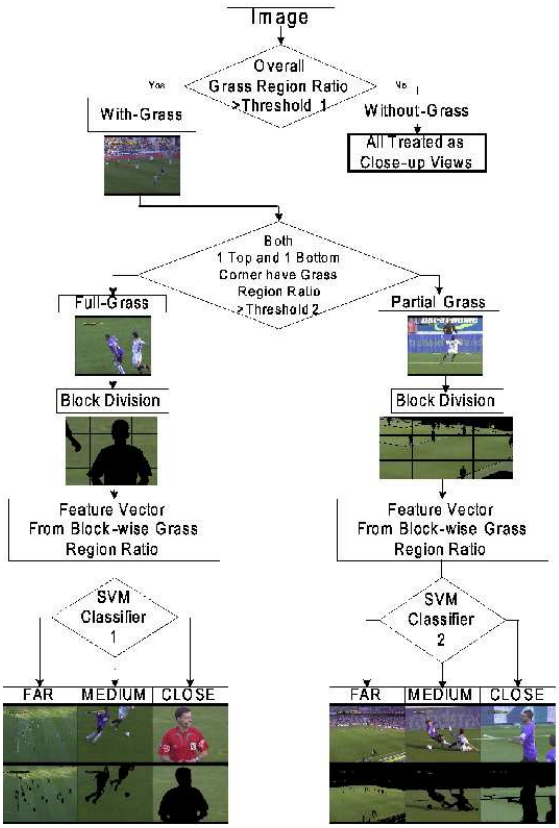


Figure 4: View Type Classification based on Grass Region Ratio.

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 in the concerned

shot. Moreover, if a shot-boundary is detected, the parameters of the zooming frame are reinitialized.

4 BALL DETECTION

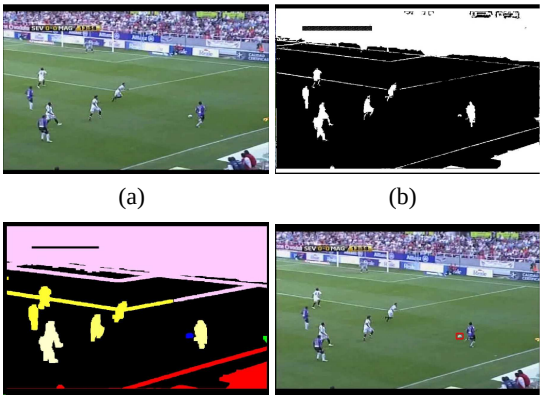
In a soccer game, the ball represents the central element of the scene. Indeed, players react according to the ball position. Consequently, a ball detection enables a ROI focusing. This problem is discussed in various studies as in (Yu et al., 2003) wherein the best ball candidate is detected with a modified version of the directional circle Hough transform. Then, the detection is 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 Gaussian filter to reduce the noise from original image.

Segmentation: Using color components, we select non-green pixels to separate the soccer field to other entities. Then a morphological operator of closure is applied to delete small artifacts and to merge sparse clusters of pixels belonging to the same object (players´ legs for example). We assume that the ball have a minimum size in order to not erase it during the morphological operation.

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

While this method is simple and efficient, it can reveal some problems when the ball is connected to players. A smoothing term in the zooming parameters estimation will help us to fix it (See Section 5).



(a) (b) (c) (d)

Figure 5: Ball detection process: (a) Input image. (b) Soccer field segmentation. (c) Label map after closure. (d) Ball bounding box in red.

5 ZOOMING FRAME

The final purpose of our system is to adapt a soccer game video to small screens highlighting ROIs. 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 watcher understanding what happens in the scene. Both aspects are controlled by parameters of position and zoom factor. Key elements to compute these parameters are followings:

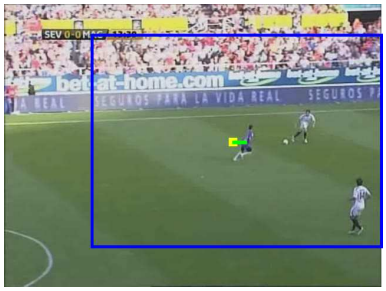


Figure 6: The zooming frame in blue. In the center of the rectangle, the zooming frame displacement.

Ball position: 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 motion: Ball motion is used to update the zoom factor. If the movement of the ball is small between two consecutive frames, zoom factor increases because the area of interest position evolves slowly. 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 action with a small zoom.

Frame rate: Higher is the frame rate, 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 input video and resolution of target screen to initialize minimal zoom value Z_{min}, maximal zoom value Z_{max} and initial zoom value Z_0.

Visual rendering: 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. Thus we added a smoothing term to control zooming frame trajectory and zoom variation. In addition, it is necessary to consider false alarms and non detections the system can generate. If no ball is detected in the current frame, the zoom value decreases to avoid zooming on a non interesting area. In this case, ball position is defined as the center of the original image. In the case of a false alarm, we noticed that ball position varies extensively between consecutive frames. Consequently this case is taken into account by the ball motion. A high motion decreases the zoom value.

Taking into account all these aspects, we compute the zoom factor and the zooming frame position with the following expressions:

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

$$C_i = C_{i-1} + \frac{1}{\beta.n} \sum_{k=i-n+1}^{i} (B_k - B_{k-1}) \qquad (2)$$

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

θ_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 α and β 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 zoom factor and zoom position values: $\forall i \in \mathbb{N}^*, Z_i \in [Z_{min}, Z_{max}]$ 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 graphic elements, such 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 can become inconsistents. To cope with such element, we add another process which detects and removes the score banner. 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.

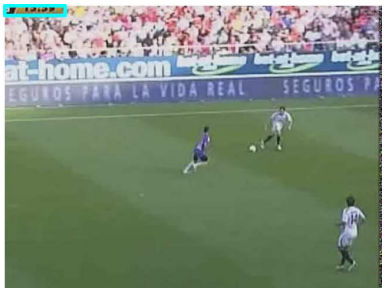
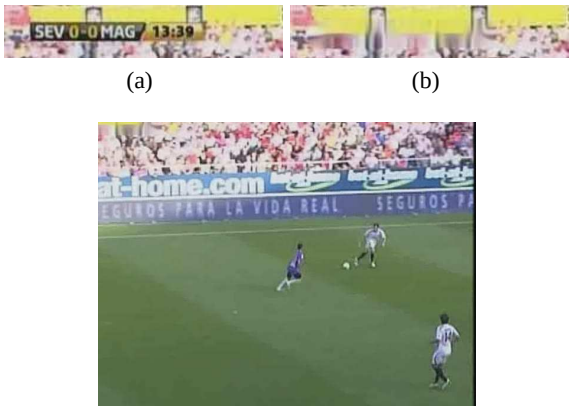


Figure 7: Incoherent resized image. In the top left corner, a piece of the banner can be seen.

The banner containing score and time of game is always displayed at the same position 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 the 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). The inpainting method is implemented as a two-step algorithm. First, image isophotes in missing data regions are reconstructed using multi-valued PDEs that perform anisotropic smoothing. Then, missing texture are synthetize therein using a smart bloc matching scheme.

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 360x640. We used the following configuration: $\beta = 15$, $\alpha = 0.3$, $\rho_1 = 0.03$,



(a) (b)

(c)

Figure 8: (a) Original score banner. (b) Banner is inpainted. (c) Rectified incoherent image.

$\rho_2 = -0.03$, $\rho_3 = -0.03$ and $n = 10$. These values have been choosen after few visual tests. Then each parameter have been hard-coded (fixed for all contexts). Some results are shown in Fig.9. We realized a visual subjective evaluation, comparing the video generated by our system and a video created by resizing the original video to the target screen resolution. The new video is more pleasant to watch and it is easier to understand what happens in the scene.

However, as we shown on Fig.10, 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. Based on this framework, we implemented a specific application for soccer game videos filmed by a multi-ptz camera network. The results are visually efficient and show the relevancy of our method. New ways have to be explored in order to improve our system. First of all, we could take into account 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 evaluate the performance of our system with a subjective assessment method proposed by International Telecommunication Union (ITU) 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 3DMEDIA projects.

REFERENCES

Ariki, Y., Kubota, S., and Kumano, M. (2006). Automatic production system of soccer sports video by digital camera work based on situation recognition. In *ISM '06: Proc.s of the Eighth IEEE International Symposium on Multimedia*, pages 851–860, Washington, DC, USA. IEEE Computer Society.

Bomcke, E. and Vleeschouwer, C. D. (2009). An interactive video streaming architecture fo h.264/avc compliant players. In *Multimedia and Expo, 2009. ICME 2009. IEEE International*.

Delannay, D., Roover, C. D., and Macq, B. (2003). Temporal alignment of video sequences for watermarking systems. pages 481–492, Santa Clara, USA. SPIE.

Ekin, A., Tekalp, A. M., and Mehrotra, R. (2003). Automatic soccer video analysis and summarization. *Image Processing, IEEE Trans. on*, 12(7):796–807.

ITU-R (2009). Recommandation bt.500-11 : Methodology for the subjective assessment of the quality of television pictures. Technical report.

Knoche, H., Papaleo, M., Sasse, M. A., and Vanelli-Coralli, A. (2007). The kindest cut: enhancing the user experience of mobile tv through adequate zooming. In *MULTIMEDIA '07: Proc.s of the 15th international conference on Multimedia*, pages 87–96, New York, NY, USA. ACM.

Pan, H., van Beek, P., and Sezan, M. I. (2001). Detection of slow-motion replay segments in sports video for highlights generation. In *ICASSP '01: Proc.s of the Acoustics, Speech, and Signal Processing, 2001. on IEEE International Conf.*, pages 1649–1652, Washington, DC, USA. IEEE Computer Society.

Takahashi, Y., Nitta, N., and Babaguchi, N. (2005). Video summarization for large sports video archives. *Mul-*

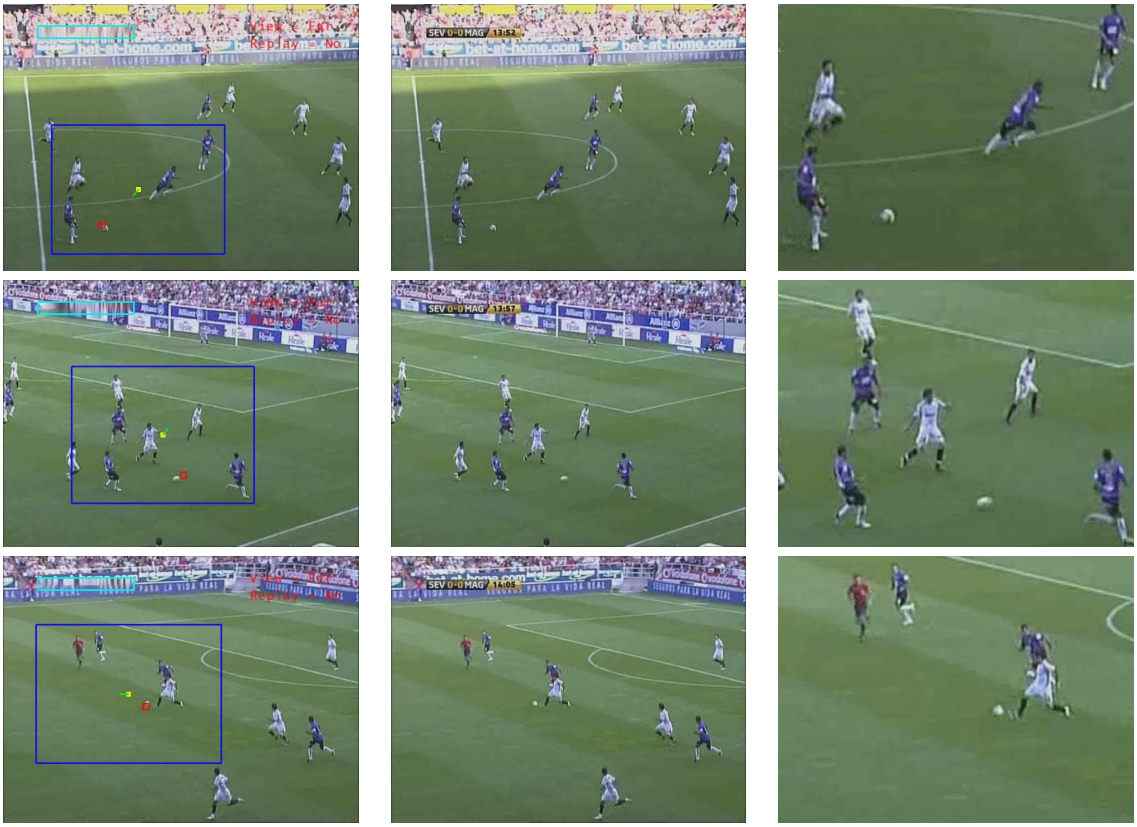


Figure 9: Left column shows processings (ball detection, banner inpainting and zooming box). Middle column shows original images resized to the target screen. Right column shows images resized to the target screen.

(a) (b)

Figure 10: Two problematic situations for our algorithm. (a) The ball is followed at the right of the original image but the ROI is located at left where a player has been hurt. (b) The ball is lost because of a sudden big shot.

timedia and Expo, IEEE International Conf. on, 0:1170–1173.

Tschumperlé, D. (2006). Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *Int. J. Comput. Vision*, 68(1):65–82.

Yu, X. and Farin, D. (2005). Current and emerging topics in sports video processing. In *IEEE International Conf. on Multimedia and Expo (ICME)*.

Yu, X., Jiang, N., Cheong, L.-F., Leong, H. W., and Yan, X. (2009). Automatic camera calibration of broadcast tennis video with applications to 3d virtual content insertion and ball detection and tracking. *Comput. Vis. Image Underst.*, 113(5):643–652.

Yu, X., Xu, C., Leong, H. W., Tian, Q., Tang, Q., and Wan, K. W. (2003). Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In *MULTIMEDIA '03: Proc.s of the eleventh ACM international conference on Multimedia*, pages 11–20, New York, NY, USA. ACM.