Analysis and Retargeting of Ball Sports Video

Stephan Kopf*, Benjamin Guthier
University of Mannheim
68131 Mannheim, Germany
{kopf|guthier}@informatik.uni-mannheim.de

Dirk Farin, Jungong Han
University of Technology Eindhoven
5600MB Eindhoven, The Netherlands
dirk.farin@gmail.com, jg.han@tue.nl

Abstract

The quality achieved by simply scaling a sports video to the limited display resolution of a mobile device is often insufficient. As a consequence, small details like the ball or lines on the playing field become unrecognizable. In this paper, we present a novel approach to analyzing court-based ball sports videos. We have developed new techniques to distinguish actual playing frames, to detect players, and to track the ball. This information is used for advanced video retargeting, which emphasizes essential content in the adapted videos. We evaluate the precision and recall achieved in the analysis and measure the computational time taken. In addition, we compare our new approach to other video retargeting techniques based on scaling, cropping, and seam carving.

1. Introduction

Nowadays, the playback of sport events is highly relevant not only for television but also for small handheld devices or the Web. The quality of video retargeting is especially important for these events due to fast motion and small objects. For example, in case of tennis videos, the court lines and the ball may no longer be perceptible in low resolution videos. We use tennis videos as an example to describe our new algorithms. However, the techniques are also applicable to other sports and are utilized to adapt badminton, soccer, and volleyball.

A general approach to video retargeting is to scale or crop the frames. *Saliency maps*, *optical flow*, or relevant objects, *e.g.*, faces or moving objects, may be used to identify *regions of interest* [1]. In our mobile cinema project [4], we have presented a video adaptation technique where we put the focus on the preservation of foreground objects like general objects, faces and superimposed text. The goal is the preservation of the aspect ratio, but also to maximize the visible information of these regions in the adapted video. Image regions are subject to constraints to preserve a minimum perceptible size as well as a maximum reasonable size. Solving an optimization problem (maximize the visible content) results in the parameters for scaling and cropping [6]. Algorithms based on scaling and cropping perform poorly if relevant objects are located at different image borders. In the case of sports videos, the motion of the players makes cropping of large borders impractical.

A new video retargeting technique called *seam carving* was proposed by Rubinstein *et al*. [8]. The idea is to detect and remove connected lines of pixels (seam manifolds) of low energy. The graph cuts algorithm is used to detect the next optimal 2D seam manifold. The computational complexity and the memory requirements depend on the number of pixels in a shot, which makes the technique only applicable to low resolution videos. We proposed an enhanced seam carving algorithm that overcomes these computational limitations [5]. By analyzing and compensating camera motion in each shot, we reduce the complexity of detecting seams from a 3D minimization problem into a 2D problem. Another major disadvantage of seam carving is the fact that straight lines become curved and cause clearly noticeable errors. We have proposed a technique to reduce these effects by modifying the energy in the local neighborhood of the intersection point of a seam and a straight line to prevent other seams from removing adjacent line pixels [3]. Although the noticeable errors in buildings or streets drop significantly, this technique does not provide acceptable results for tennis videos due to the large number of court lines.

A general problem of all techniques that change the selection of pixels to be removed or scaled over time (like seam carving or warping) is the fact that straight lines become curved or that jitter is added to the video. We present a novel approach which avoids these problems. The basic idea of our approach is to identify and improve the most critical regions in a sports video. It is obvious that players, court lines, and the ball are most relevant in many sports.

Several approaches have been proposed to analyze ball sports video like tennis [2, 7] or soccer [9]. Most approaches focus on the detection of court lines and players,

*A major part of this work was performed at Eindhoven University.

and use this information to derive important semantic events like goal, foul, service, or rally. In previous work, we also put our focus on the detection of court lines and players [2]. Automatic ball tracking is used in some approaches to enable virtual replays and to derive important game events. Seo *et al*. presented a technique to detect and track a soccer ball [9]. This technique is not easily applicable to other sports like badminton or tennis due to the differences in size and speed of the ball. In case of tennis ball tracking, most approaches use high quality cameras or even tracking from multiple cameras [7]. These approaches are not applicable to broadcast videos due to noise and compression artifacts. We do not consider semi-automatic approaches that require a manual initialization whenever the ball is lost.

In this paper, we present novel techniques to analyze and adapt court-based sports video. Compared to other approaches, our framework is applicable to different kinds of ball sports. The distinct features of our approach are:

1. We propose new algorithms to analyze court-based ball sports video: a fast algorithm for court scene detection, a novel technique to segment players and objects, and a robust algorithm to detect and track the ball (without manual initialization).
2. We present our novel video retargeting application for sports video. The idea is to enhance the visibility of court lines and the ball, to identify borders of low importance, and to use a combination of cropping and scaling to reduce the spatial resolution of the video.

The paper is structured as follows: The following section focuses on algorithms to analyze sports video. Section 3 describes our adaptation system. In Section 4, we evaluate the algorithms, and we conclude the paper in Section 5.

2. Analysis of sports video

Our sports video analysis and adaptation system is based on four modules which analyze a video and one additional module for video retargeting (see Figure 1). In a first step, the system distinguishes playing frames from other frames. Additional modules detect court lines, players, objects, and the ball. For each frame, the components are run one after another. Only if a frame is processed successfully in the current step, it is forwarded to the next module. So, for example, if the court field is not detected in a frame, algorithms to locate court lines or players are skipped. Previously computed data is re-used wherever possible.

The adaptation of a frame depends on the results of the analysis. A frame is scaled if no semantic information could be derived. Otherwise, the advanced video retargeting module uses all available information to adapt a frame. Figure 1 gives an overview over the components of our system; Figure 2 shows exemplary intermediate results of the processing steps.

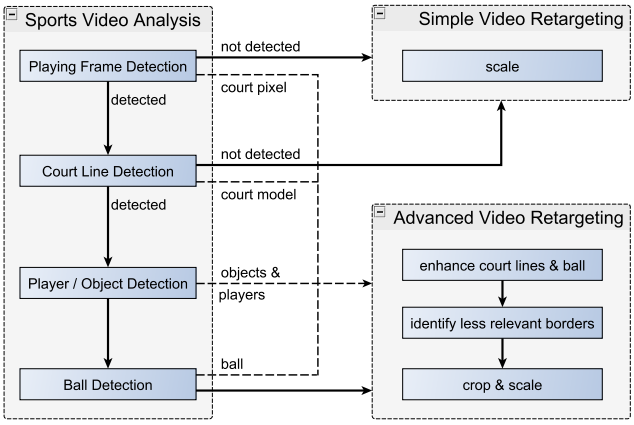

Figure 1. System overview

2.1. Playing Frame Detection

A sports video consists of two types of shots: Those in which the actual game takes place (playing shots) and other shots showing audience, display panels, close-up shots of faces, or advertisement. Only playing shots should be processed by our advanced video retargeting algorithm; all other frames are simply scaled.

We assume that a frame containing large areas of similarly colored pixels depicts a court and is thus considered to be part of a playing shot. Our algorithm distinguishes between an initialization phase to detect the dominant court colors and a playing shot detection phase. In the initialization phase, a histogram H_i based on the 4 most significant bits of each color channel in the *RGB* color space is calculated (4096 bins). A small image border is ignored to reduce the computational effort and to remove highly textured regions of playing frames (e.g., the audience). The values of the histogram bins are sorted in descending order and stored in H_i^*. The minimum number of bins N is calculated, so that the first N bins contain T_H percent of all pixels:

$$\arg_N \left(\sum_{i=1}^{N} H_i^* > T_H \wedge \sum_{i=1}^{N-1} H_i^* \leq T_H \right) \qquad (1)$$

A frame is then labeled *playing frame* if N is less or equal than T_N percent of the bins. If the current frame is not a playing frame or no valid court model is detected (see Section 2.2), the initialization phase is repeated for the following frame. Otherwise, the N largest bins in the histogram are marked. Part of the computational effort is then saved in the detection phase. Subsampled frames are used to calculate a histogram for each subsequent frame. If the sum of pixels in the marked bins exceeds T_L percent of all pixels, the frame is labeled *playing frame*. The threshold values were derived empirically. $T_H = 85$, $T_L = 80$, and $T_N = 3$ provide reliable results for sports videos like tennis, soccer, or badminton.

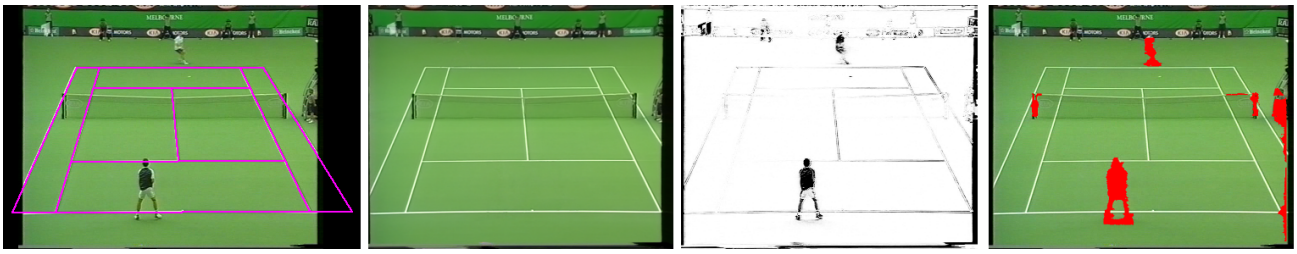

Figure 2. Results of the analysis steps: court model, background image, difference image, player and object segmentation

2.2. Court Line Detection

Court lines are highly relevant to understanding a sports video. Therefore, they should be clearly visible in the final video and a precise localization is essential for retargeting. We perform a full court line search consisting of three steps on the first frame and then a computationally cheaper court model tracking on the subsequent frames.

The first step is the detection of *court line pixels*. We do this by detecting white pixels that fulfill certain constraints regarding brightness, local structure, and texture. From the court line pixels, we then estimate a *parametric description of the court lines* in the next step. In the third step, a *geometric transformation* between the parametrized court lines in a frame and a court model is calculated. Court models for tennis, badminton, volleyball, and soccer are modeled in our system.

In all subsequent frames, the transformation matrices calculated so far are used to perform *court model tracking*, which is considerably faster than the matching. This court line detection and tracking method was previously published in [2]. Please refer to the original publication for more details.

2.3. Player Detection

Knowing which image regions belong to moving objects (e.g., players or billboards) helps to make the detection of the ball more robust. We aggregate nine previous frames into one *background image*. The frames are chosen from one shot, so that the temporal distance between the frames is maximized. The camera motion is compensated by transforming each frame by its respective transformation matrix. Then, foreground objects are removed from the background image by applying a median filter to all pixels at one image position. A *difference image* is then computed by subtracting the background image from the current motion compensated frame.

In the next step, high pixel values of the difference image are clustered into *regions*. First, the pixels in the difference image belonging to court lines are set to zero and holes are filled by a morphological closing operation. Then, a vertical projection profile is calculated that sums up the differences in each column. The maximum peak of this projection profile is defined as start position. All columns to the left and right of the start position which exceed 10 percent of the maximum value are selected. These selected contiguous columns approximate the horizontal position of a player or an object in the frame very well. This is repeated with a horizontal projection profile to estimate the vertical position.

If the bounding box of the selected region is larger than a set minimum size, it is classified as player/object region. The algorithm continues to detect regions as long as the maximum value of the vertical projection profile exceeds a threshold. This threshold reflects the minimum height of an object. Figure 2 shows sample results of the individual steps.

2.4. Ball Detection

The fast motion of a small tennis ball causes blurring, interlacing artifacts, and disconnected ball pixels that differ only slightly from background pixels (see Figure 3). Simple features like color, shape, or contrast are thus unsuitable to detect the ball. We present a ball tracking algorithm based on a particle filter which does not assume that the ball can be detected in all frames. It consists of the following steps:

1. Create probability map for current frame.
2. Initialize particles (only once).
3. Update position and diffuse position of particles.
4. Create new particles.
5. Estimate position of ball and mark ball pixels.

The algorithm starts by calculating a probability map that describes each pixel's similarity to the ball. For this purpose, the last four frames are aligned to the current frame and a difference image $D(x, y)$ is calculated. For each pixel position (x, y), it contains the maximum absolute difference of the pixels of the current frame $I_0(x, y)$ and the aligned frames $I_k(x, y), k = 1 \dots 4$:

$$D(x,y) = \max_{\forall k=0\dots4}\{I_k(x,y)\} - \min_{\forall k=0\dots4}\{I_k(x,y)\}.$$

All $D(x, y)$ for which at least one of the following is true are set to zero:

- $D(x, y) < T_D = 30$,

Figure 3. Typical visual quality of the ball in the test videos. Top row: ball in slow motion. Bottom row: fast motion.

- $I_0(x, y)$ is darker than the court color,
- (x, y) coincides with a court line, is inside or adjacent to an object region, or is part of a large region of high difference values.

The probability map PM is then set to the normalized difference image ($\sum_{\forall x,y} PM(x, y) = 1$).

A particle filter (or *sequential Monte Carlo method*) estimates the position of the ball for each frame t based on observed data, which in our case is the probability map $PM_t(x, y)$. We use a total of 300 particles. Each particle S_i is a triple $S_i = (x_i, y_i, p_i)$, consisting of a pixel position (x_i, y_i) and a probability p_i.

Initialization of the particles is only done once. For each particle, a pixel position is randomly chosen according to the probability distribution PM_0. It follows that a pixel with zero probability is never chosen as particle; a pixel with a high probability may be selected several times.

Step 3 modifies the position of all particles. If a ball was detected in the last two frames, the direction and speed of the ball are estimated and the positions of all particles are updated accordingly. Because the real position of the ball may vary due to changes in speed and direction, the particles are additionally scattered by a Gaussian distribution.

In step 4, the probabilities of the particles are updated by the probability map PM_t of the current frame. Each particle is substituted by a new particle chosen randomly according to the probability distribution defined by the current particles.

In the last step, the position of the ball is estimated from the current particles. They are divided into 10 clusters by K-Means using Euclidean distance. The probability of a cluster is defined as the sum of probabilities of its particles. We define a *relevance score* for each cluster, which is the sum of cluster probability, the inverse cluster size (the ball is very small in most sports), and the distance to the estimated position of the ball from previous frames (if available). The position of the ball is the center of the cluster with the highest score. Bright pixels in the local neighborhood of the cluster center are marked as ball pixels.

3. Adaptation of Sports Videos

If no playing frame was detected, our application uses simple bi-linear interpolation to scale the frame to the designated size. Otherwise, the following advanced adaptation technique is used: First, the court lines are emphasized to make them easier for users to recognize. This is done by applying gamma correction to these pixels. Even if the ball is occluded, the ball detection algorithm selects the most probable ball region. To avoid false ball detection, the motion of the ball in the last three frames is analyzed. If the speed and direction are valid, the ball pixels are dilated to close gaps caused by interlacing and gamma correction is applied on the dilated pixels.

After emphasizing ball and lines, borders of low relevance are identified in a frame. The relevance of a border depends on the location of the ball and the players. Parameters to crop a frame are calculated, so that a certain minimum distance of players and ball to each border is kept. The maximum width of the bounding boxes of the players defines the minimum distance. To avoid camera jitter, the width of each cropped border may only change by 1 pixel from frame to frame. In the last adaptation step, we use bi-linear interpolation to scale the frame to its final size. Figure 4 shows some results of our algorithm and compares them to simple scaling.

4. Evaluation

In the following, we present three evaluations: The first examines the computational effort and the quality of the analysis steps for tennis. The second evaluation shows the applicability of the algorithms to other court-based sports, whereas the third compares the quality of our new video retargeting approach to scaling, cropping, and seam carving.

In our first evaluation, we focus on the detection of playing frames, object segmentation, and ball detection. The quality of *line detection* was already discussed in previous work [2]. We selected six test sequences with a total length of 38 minutes (five in PAL and one in CIF resolution). The videos were chosen to cover a broad variety of tennis scenarios, i.e., singles and doubles, different court surfaces, and lighting. Figure 4 (left) depicts one of these videos.

Analysis steps	Precision	Recall
playing frame detection	96.1 %	99.8 %
player detection	90.2 %	91.9 %
ball detection	85.4 %	33.6 %

Table 1. Quality of the analysis steps

Based on the number of correct (C), missed (M), and false (F) detections, we compute precision $P = \frac{C}{C+F}$ and recall $R = \frac{C}{C+M}$ of the individual algorithms (see Table 1). For this purpose, all six videos were used. For player detec-

Figure 4. Examples of tennis, badminton, and soccer videos adapted to a resolution of 240x192 pixels. Top row: scaled. Bottom row: adapted with our advanced video retargeting approach ($\gamma = 3.0$). Court lines in the background are much easier to see.

tion and ball detection, 200 frames were randomly chosen from each video and the results were manually inspected.

We consider the *detection of playing frames* first: False hits are not as critical as a low recall, because missed frames will not be processed in the following steps. The missed frames are typically located at the beginning of new shots, e.g., in cases of short dissolves. The overall fraction of playing frames is 49.6 %.

Ball detection works much better when all regions with moving objects (including players) are automatically marked beforehand. According to our definition, the *object segmentation* is correct if at least all player regions are marked and if the ball is not part of the selection. Most false hits occur because the position of the tennis ball is adjacent to the player, so that ball and player regions merge. Regions are typically missed if the player is stationary for a long time, e.g., during a serve.

We define a miss in the *ball detection* phase as having obtained a ball position that is inconsistent with the previous frames and thus having discarded it. Based on this definition, our ball detection has a recall of only 33.6 %. This is mainly due to the ball being occluded or being too similar to background pixels like court lines. As a consequence of our implementation, ball detection then also fails during the next three frames. Oftentimes, this kind of miss is uncritical, because the ball cannot be recognized in the original video either. A false hit in ball detection occurs when the ball is detected in a consistently wrong place for several frames in a row.

To measure the *average computing time* of the individual steps, we used a standard PC (AMD 2.4 GHz, 4 GB RAM) and only considered the PAL resolution videos (see Table 2). The frame alignment and calculation of the difference image is the most expensive step and takes 559.9 ms, followed by 195.1 ms for the calculation of the difference image for ball detection.

Analysis step	Computational effort [in ms]
playing frame detection	
- initialization	0.9
- detection	0.2
player detection	
- frame alignment / difference image	559.9
- selection of pixels	17.3
ball detection	
- difference image	195.1
- particle filter	14.9

Table 2. Computational effort

In the *second evaluation*, we tested the applicability of the proposed algorithms for other ball sports. Whenever possible, the parameters and thresholds were not modified. No modification of parameters was required for playing frame detection, and the reliability is similar for tennis, badminton, and soccer. In case of volleyball, the recall drops slightly due to the larger number of players.

Two parameters of the court line detection algorithm were adjusted: the court model and the color of the court lines. Precision and recall of court line detection depend

on the total number of visible lines in a frame. The court modeling usually fails when players occlude large parts of volleyball lines or when insufficient soccer lines are shown. Curved lines such as the one next to the penalty spot in Figure 4 (right) are ignored by our algorithm and will not be highlighted.

Compared to tennis, ball detection is more reliable for badminton and soccer. Due to the small size and the speed of the tennis ball, the amount of blurring and interlacing artifacts is usually very high. Especially in case of volleyball, we observed that the particle filter fails when shirt and ball color are similar. If this is the case, the adaptation of the ball should be disabled. This is not critical for soccer or volleyball due to the size of the ball.

We carried out a *user study* to evaluate the quality of the adapted sports videos. Several videos were adapted to a screen resolution of 240x208 pixels using scaling, cropping, seam carving, and our new approach. After an initial analysis, it became obvious that seam carving is completely unsuitable for ball sports videos. A video demo comparing the three applicable techniques is available on our web page[1]. The user study thus focuses on scaling and our new approach. The effect of the parameter γ on court lines and the ball was analyzed in particular. Nine videos with different parameters were generated. Ten users graded the quality of the adapted videos on a scale from 5.0 (excellent) to 1.0 (insufficient). Additional questions regarding the general quality of the videos were asked. Table 3 summarizes the results.

Parameter γ	Court lines	Ball
1.0 (no modification)	2.7	1.2
1.5	3.3	3.1
3.0	3.4	3.7

Table 3. User evaluation: visibility of court lines and ball

A modification of the court lines does only slightly improve the perceptibility. The enhancement of the ball on the other hand is of higher importance. Even a small modification makes it much easier to recognize its location. Users commented that the ball is lost in some frames and rated the quality of these videos very low. We also evaluated the effect of cropping in three categories (none, small, large). The average values for all categories are very similar, even though the standard deviation of the grades is very high. Some users preferred centered views on ball games; others did not like the additional motion caused by cropping. Still, some users rated the cropped videos much better. We conclude that users need to be able to adjust this value to their personal preferences.

[1] http://pi4.informatik.uni-mannheim.de/pi4.data/content/projects/moca/

5. Conclusions and Outlook

The precision of court line, player, and ball detection is high enough to achieve good adaptation quality of sports videos. Our approach has advantages over previous semi-automatic techniques which require user interaction to determine the position of the ball in some frames. The advanced video retargeting offers significant advantages compared to other approaches: Court lines and especially the ball are discernible more easily without compromising the temporal stability of the image.

6. Acknowledgments

A major part of this work was performed during a research stay of Stephan Kopf at Eindhoven University. We would like to acknowledge and thank the Video Coding and Architectures Research Group at Technical University Eindhoven for the financial support. The authors acknowledge the financial support granted by the Deutsche Forschungsgemeinschaft (DFG).

References

[1] T. Deselaers, P. Dreuw, and H. Ney. Pan, zoom, scan – Time-coherent, trained automatic video cropping. In *Comp. Vision and Pattern Recognition*, pages 1–8, 2008.

[2] J. Han, D. Farin, and P. H. N. de With. Broadcast court-net sports video analysis using fast 3-D camera modeling. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1628–1638, 2008.

[3] J. Kiess, S. Kopf, B. Guthier, and W. Effelsberg. Seam carving with improved edge preservation. In *Proc. of IS&T/SPIE Conference on Multimedia on Mobile Devices*, volume 7542, pages 75420G–75430G, 2010.

[4] S. Kopf and W. Effelsberg. Mobile cinema: Canonical processes for video adaptation. In *Multimedia Systems*, volume 14, pages 369–375. Springer Berlin / Heidelberg, Dec. 2008.

[5] S. Kopf, J. Kiess, H. Lemelson, and W. Effelsberg. FSCAV: Fast seam carving for size adaptation of videos. In *Proc. of ACM Intl. Conf. on Multimedia*, pages 321–330, 2009.

[6] S. Kopf, F. Lampi, T. King, and W. Effelsberg. Automatic scaling and cropping of videos for devices with limited screen resolution. In *Proc. of ACM Intl. Conf. on Multimedia (video program session)*, pages 957–958, Oct. 2006.

[7] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *Intl. Conf. on Pattern Recognition*, volume 4, pages 152 –156, 2000.

[8] M. Rubinstein, S. Avidan, and A. Shamir. Improved seam carving for video retargeting. *ACM Trans. on Graphics, SIGGRAPH*, 27(3), 2008.

[9] K. Seo, J. Ko, I. Ahn, and C. Kim. An intelligent display scheme of soccer video on mobile devices. *IEEE Trans. on Circuits and Systems for Video Technology*, 17(10):1395 – 1401, Oct. 2007.