BACKGROUND SUBTRACTION FOR REALTIME TRACKING OF A TENNIS BALL

Jinzi Mao, David Mould and Sriram Subramanian
Department of Computer Science, University of Saskatchewan, Saskatoon, Canada

Keywords: Motion tracking, background subtraction, real-time vision, tennis ball detection, tennis ball tracking.

Abstract: In this paper we investigate real-time tracking of a tennis-ball using various image differencing techniques. First, we considered a simple background subtraction method with subsequent ball verification (BS). We then implemented two variants of our initial background subtraction method. The first is an image differencing technique that considers the difference in ball position between the current and previous frames along with a background model that uses a single Gaussian distribution for each pixel. The second is uses a mixture of Gaussians to accurately model the background image. Each of these three techniques constitutes a complete solution to the tennis ball tracking problem. In a detailed evaluation of the techniques in different lighting conditions we found that the mixture of Gaussians model produces the best quality tracking. Our contribution in this paper is the observation that simple background subtraction can outperform more sophisticated techniques on difficult problems, and we provide a detailed evaluation and comparison of the performance of our techniques, including a breakdown of the sources of error.

1 INTRODUCTION

While ball tracking systems have been successful in soccer (football) and baseball, ball tracking in tennis matches is less well explored. Applications including computer-assisted refereeing and computer-assisted coaching could benefit from real-time tracking of the tennis ball. However, ball tracking in a tennis match poses particular challenges owing to the ball's small size, high speed, and large variation in trajectories. Soccer balls are relatively large and move relatively slowly, while baseballs are slightly larger and (in a baseball pitch) have a much more highly constrained trajectory.

Neither object-based techniques nor non-object-based techniques are suitable for this application. Such techniques have limited processing speed and sometimes lack the ability to track objects which move significant distances between frames. In this paper, we present an investigation of image processing algorithms aimed at tennis ball tracking. We use background subtraction as the first step in the tracking process; background subtraction generates a number of regions representing changes in the image, all of which are possible ball locations, or ball candidates. We determine which ball candidates to report as tennis balls based on size and shape analysis of the candidate regions. First, we present a basic algorithm which uses an extremely simple, static background model; the results from this were encouraging, so we created two variant methods with different augmentations to the background model. The variants outperformed both the initial method and two existing alternative methods, even in a context where the background varied considerably and background subtraction might be thought unsuitable.

Tennis ball tracking systems were reported by Sudhir et al. (Sudhir et al., 1998) and by Pingali et al. (Pingali et al., 2000); neither group of authors made a systematic analysis of their systems' performance in a real-world environment. Also, they did not address ball occlusion or player-ball interaction. Our paper describes the results of our real-world deployment and gives detailed data on error rates and sources of error.

We present a framework for tennis ball tracking

Mao J., Mould D. and Subramanian S. (2007).
BACKGROUND SUBTRACTION FOR REALTIME TRACKING OF A TENNIS BALL.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 427-434

intended to satisfy two main requirements: speed and robustness. First, the system needs to execute quickly, in real time or close to it. Second, the system needs to be robust to changes in the environment, to deal with other moving objects, to track a fast-moving and erratically moving object, to deal with occlusion, and to track an object whose screen-space size varies significantly during a single test, owing to changes in distance to the camera (the ball ranged in apparent size from 5 pixels to 60 pixels during our tests). As we will see, we met some of these criteria, but not all; nonetheless, the techniques we developed have considerably better performance than existing techniques that we compared against.

The paper is organized as follows. Section 2 discusses previous work, including existing methods which we incorporate into our algorithm. Section 3 describes our algorithms. Section 4 describes our experiments, analysis of the data, and evaluation of the technique. Finally, we close in Section 5 with concluding remarks and suggestions for future work.

2 PREVIOUS WORK

A number of object detection and tracking techniques have been developed in the last two decades for tracking humans (Rano et al., 2004) and cars (Stauffer and Grimson, 1999). More recently, researchers have examined computer vision techniques for tracking sporting events (Han et al., 2002; Assfalg et al., 2002; Sudhir et al., 1998). Here we review the related literature on computer vision based object detection and tracking techniques.

Viola et al. (Viola and Jones, 2001) introduced classifier cascades for object recognition. They trained a set of weak classifiers on a set of very simple features, one classifier per feature; the classifiers are used in sequence to detect the presence of the target object, and since the weak classifiers are able to reject most non-target objects quickly, the majority of the computational effort is spent on difficult cases. Lienhart and Maydt (Lienhart and Maydt, 2002) extended this work by proposing a richer set of features (Haar-like features, including edge, line, and center-surround features) and showing a lower false positive rate than was achieved by the simple feature set of Viola et al (Viola and Jones, 2001).

Stauffer and Grimson (Stauffer and Grimson, 1999) proposed a background model in which each pixel is a mixture of Gaussian distributions; pixels which fit into some existing distribution are considered background, while pixels which lie outside all distributions are considered foreground. The method allows the distributions to adapt to new samples, so that only parts of the image which change faster than a set learning rate are still considered foreground, and portions which change more slowly are incorporated into the background.

Ren et al. (Ren et al., 2004) devised K-ZONE, a system for tracking baseball pitches. They used a mixture of Gaussians for background discrimination; their method uses trajectory information to reject some ball candidates. They report good results for their context, but the trajectory of the baseballs is considerably constrained compared to the variation we can expect in a tennis match.

D’Orazi et al. (D’Orazio et al., 2002) propose a system for tracking soccer balls using a modified Hough transform. They use the parametric representation of a circle to transform the image and determine points which are on the soccer ball. They show that the circular Hough transform is effective in detecting the soccer ball. However, their algorithm requires considerable processing to be viable as a real-time ball tracking technique.

In (Sudhir et al., 1998) the authors perform an automatic analysis of tennis video to facilitate content-based retrieval. They generate an image model for the tennis court-lines based on the knowledge of the dimensions and connectivity of a tennis court and typical geometry used when capturing a tennis video. They use this model to track the tennis players over a sequence of images.

In (Pingali et al., 2000) the authors use multiple cameras to track the 3D trajectory of the ball using stereo matching algorithms. A multi-thread approach is taken to track the ball using motion, intensity and shape. However, they do not give enough details of their implementation to compare their approach with ours.

Throughout this paper we use various image processing techniques, including median filtering and shape feature extraction (Shapiro and Stockman, 2001). The median filter is used to reduce noise in the image while shape features, including aspect ratio, compactness, and roughness, are used to check if a region’s properties resemble a ball or not.

3 ALGORITHMS AND INITIAL RESULTS

Complex algorithms, such as boosted classifiers based on Haar-like features, and circular Hough transforms have been brought to bear on the problem of tennis ball tracking. Many problems arise when they are applied to the tennis tracking system.

- The computation costs are too high. The processing speed for these referred techniques cannot guarantee high processing speed especially when high-resolution, high frame-rate cameras are used.
- The size of the tennis ball is quite small and the feature space in such a small region is limited, so can often be insufficient to describe the target object.

We believe a good strategy for the current problem would be to start with a simple algorithm and enhance it by combining the benefits of the techniques described above. A simple algorithm based on background modeling has several benefits, including high processing speed and low dependence on training dataset. We were interested in seeing how well a naive algorithm based on background modeling would perform.

3.1 Initial Approach

Our initial, naive algorithm works as follows. We compute a simple background model by averaging frames from a short video sequence of the vacant tennis court. Then, the algorithm determines ball candidates by finding regions whose intensity in the current frame is larger than in the background frame. Next, ball candidates in the vicinity of the player are discarded. Shape and dynamics characteristics for remaining candidates are computed, and those with shape too far from the ball or which did not move like the ball in previous frames are also discarded. Any remaining candidates are reported as being tennis balls.

We addressed the problem of ball-player interaction by explicitly identifying the largest foreground blob with the player. A player almost never appears as a single foreground blob because of the dynamics of the tennis game. Just before making a tennis shot players often move their forearms, upper bodies and feet, but rarely displace their hips. Thus a player often appears as 2 or 3 adjacent blobs (upper-body, legs, and forearm). We explicitly search for blobs that are comparable in size and in the vicinity of each other to identify them with the player (see Figure 1). We then extend our search to identify the player's racquet if it is visible. The main features used to connect the different blobs are blob size and adjacency. Once the player, the racquet, and the ball candidates in the vicinity of the player are discarded, the remaining candidates are checked based on size (5 to 60 pixels) shape (compactness and roughness close to 0.9) and dynamics characteristics (candidates whose distances to the predicted location exceeds a predefined threshold are removed) to discard false detections. Any remaining candidates are reported as being tennis balls. Note that this method allows us to correctly report when multiple actual tennis balls are present in the scene, as was sometimes the case (for example, a ball from an earlier rally had not been retrieved and was lying on the court while the players continued with a new ball).

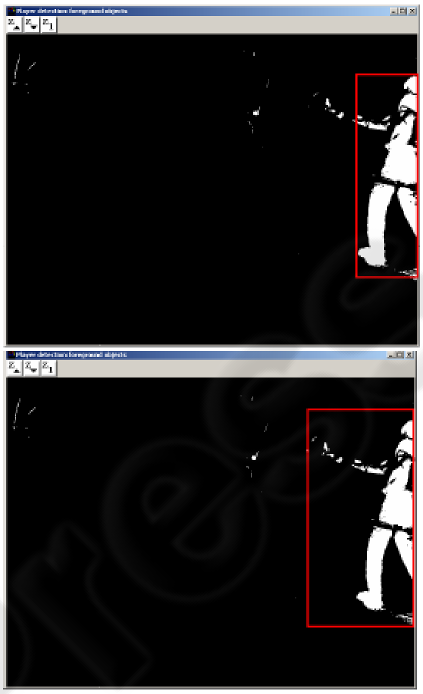

Figure 1: Algorithmic player removal. First, the largest region plus nearby large regions are detected (top). Second, the player region is expanded to include the racquet (bottom).

We compared our method against two existing methods: the technique of boosted classifiers (BC), trained with Haar-like features; and the Hough transform (HT) used to detect the tennis ball's circular shape.

The performance of our system is measured by three figures: the true positive rate; the false positive rate; and the processing rate. True positive rate is defined as the ratio of the number of correctly recognized features to the number of appearances of the feature. The false positive rate is the ratio of the number of incorrect recognitions to the number of instances where no feature was present. We measure processing rate by frame rate, that is, the number of images that can be processed in a second. A successful system has a true positive rate as high as possible, a false positive rate as low as possible, and a processing rate as high as possible.

We recorded raw data consisting of thirty brief videos of tennis games, each about 1 minute in length, in an indoor setting. We captured images using a monochrome camera at a resolution of 782 ×582 pixels at 25 fps. A sample frame is shown in Figure 2.

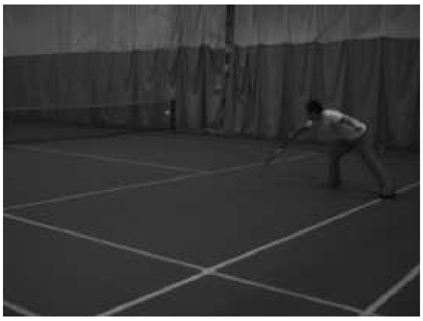

Figure 2: A sample frame from video taken on an indoor tennis court.

For each frame of each video, we manually marked the position of the tennis ball. This gave us a ground truth against which we could compare the output of the three algorithms. Then, we ran each video through each of the BS, BC, and HT algorithms. We computed true positive and false positive rates for each algorithm on the entire corpus of data; a summary of the results appears in Table 1.

Table 1: Summary of comparison between naive background subtraction and existing methods.

	BS	BC	HT
True positives	87.4%	30.1%	11.8%
False positives	1.35%	62.2%	73.4%
Processing rate	21.4	6.2	8.2

The data show that even with our naive approach we were able to do better than standard algorithms at the task of tracking the tennis ball. Our naive algorithm has a higher recognition rate, a lower error rate, and a faster frame rate than either of the comparison techniques. Also, it avoids a time-consuming training period; it took upwards of a week to train the boosted classifiers for the tennis context. The Hough transform, while not requiring much setup, performed extremely badly. The assumption that the tennis ball has a circular shape is not satisfied in frames of the video, owing to two main reasons. First, illumination can cause the tennis ball to have an apparently crescent shape rather than a round shape. Second, blurring because of rapid motion can cause the ball to appear elongated in the direction of motion. Other factors (such as noise, and deformation of the ball on impact) can also play a role.

We were pleased with our results in the indoor case, but wondered whether the good results were primarily due to the largely static background in the indoor environment. It might be that our naive approach would perform dramatically worse in cases of more dynamic backgrounds, while the performance of BC and HT would not suffer as much. Real tennis matches have somewhat dynamic backgrounds, since the crowd, the weather, and sometimes the advertisements change during the match.

To determine whether this was the case, we recorded another thirty videos in an outdoor environment. The weather on the day we did the recording was extremely active, with trees in the background swaying vigorously and clouds moving rapidly. The cloud movement caused issues for two reasons: first, the clouds were sometimes visible as moving objects in the scene; second, the clouds sometimes moved in front of the sun or out from in front of it, causing rapid changes in illumination. This highly dynamic background provides a good test case for automatic tennis ball tracking, since real tennis match videos are unlikely to have worse conditions. A sample frame from the outdoor setting is shown in Figure 3.

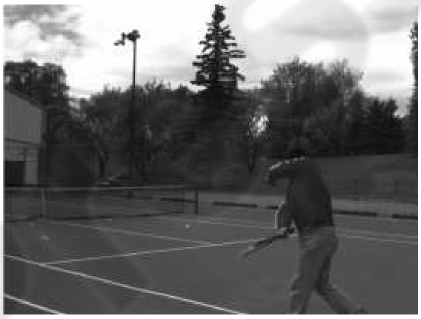

Figure 3: A frame from the outdoor tennis court. Notice the trees and clouds visible in the background.

Once the videos were recorded, we again performed manual ball detection to arrive at a ground truth and used each of the BS, BC, and HT algorithms to perform ball detection. The results are summarized in Table 2.

Table 2: Summary of comparison between naive background subtraction and existing methods in outdoor environment.

	BS	BC	HT
True positives	23.5%	27.2%	3.7%
False positives	56.4%	52.6%	77.5%
Processing rate	20.0	6.3	8.1

Not unexpectedly, the simple background subtraction algorithm did not cope well with the dynamic background; its true positive rate dropped significantly, and its false positive rate rose enormously. The method of boosted classifiers saw a slight reduction in true positive rate but now performs better than simple background subtraction. The Hough transform again performed poorly.

3.2 Improved Background Model

Having confirmed that background subtraction is a good technique when we can reliably determine the background, we decided to improve our background model. The naive approach uses a static background model obtained by averaging images of an empty tennis court. We decided to use a Gaussian model for each image pixel, inspired by the work of Stauffer and Grimson.

We devised two variants of the naive method – image differencing (ID) and mixture of Gaussians model (MG). We discuss both below.

3.2.1 Image Differencing

Image differencing (ID) is a technique inspired by the tennis ball tracking system of Pingali et al., who use the difference between the current and previous frame to determine ball candidates. The reasoning here is that the tennis ball is usually fast-moving, and will occupy an entirely different set of pixels in consecutive frames, while slower-moving objects will have significant overlap. However, Pingali et al.'s method uses a complex mechanism for estimating ball intensity levels, in order to cope with lighting, shadow, and distance variations. We wanted to see whether we could obtain good results combining image differencing with background subtraction.

In our ID technique, we model each background pixel with a single Gaussian. That is, we compute a mean μ and standard deviation σ for each pixel, using an initial set of background images; outliers (more than two standard deviations from the mean) will be considered foreground pixels.

The difference between the foreground of the previous frame and the foreground of the current frame is used to obtain one set of ball candidates, say A. The set of foreground pixels gives another set, say B. We obtain a final set of candidates C by taking the logical AND of A and B. The candidates in C are subjected to characteristics tests (compactness, aspect ratio, size) and those that pass are reported as being tennis balls. We report results of testing our ID technique in Section 3.2.3.

For pixels not considered part of the ball, parameters of the Gaussian distribution are updated as follows:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{1}$$

$$\sigma^2 = (1-\rho)\sigma^2_{t-1} + \rho(X_t - \mu_t)^2, \tag{2}$$

where ρ is the learning rate for the parameters, and X_t is the measured value for that pixel. This allows the technique to adapt to changes in the background.

3.2.2 Mixture of Gaussian Models

In the mixture of Gaussians technique (MG), each background pixel is modeled by a mixture of Gaussians; pixels that cannot be explained by any of the distributions are considered foreground objects (ball candidates). Each Gaussian is characterized by a weight w, a mean μ, and a standard deviation σ.

When creating the background model each pixel is verified against the corresponding gaussian distributions until a match is found or all distributions are checked. A match is found when the pixel value is within 2.5 standard deviation of the distribution.

If no match is found, a new Gaussian distribution is added or the least probable distribution is updated with the new pixel value as the mean, a high std., and a low weight. If a match is found, the weights of the existing distribution at time t are updated as follows:

$$w_{k,t} = (1-\alpha) * w_{k,t-1} + \alpha * M_{k,t} \tag{3}$$

where α is the learning rate for the weights, and $M_{k,t}$ is 1 for the distribution that matches and 0 otherwise. Following this reassignment the weights are normalized. The learning rate determines how fast the model responds to changes in the environment; a higher learning rate means changes to the environment will be adapted to the existing background model more quickly, while a low learning rate means that the initial background model will be slow to change.

The MG technique also updates the values for its parameters, using the process described in equations 1 and 2. Once the parameters have been updated, the Gaussians are ordered by the value of w/σ. The distribution with the higher weight w and smaller σ is the more probable representation of the current background in that pixel location.

3.2.3 Comparison of Techniques

Both variants use the player-removal system we designed for the naive algorithm.

We tested both the ID and MG approaches with the videos we had already obtained. We will make a comparison directly with the naive BS algorithm. Table 3 shows a summary of the results for the indoor case, and Table 4 summarizes the outdoor case.

Table 3: Summary of indoor results for our three methods.

	BS	ID	MG
True positives	87.4%	89.6%	90.7%
False positives	1.35%	1.07%	1.02%
Processing rate	21.4	19.0	14.7

We see only a slight improvement in quality of results in the indoor case; this is not surprising, since

Table 4: Summary of outdoor results for our three methods.

	BS	ID	MG
True positives	23.5%	35.6%	39.5%
False positives	56.4%	49.8%	26.5%
Processing rate	20.0	19.5	13.8

the nature of the indoor scene was such that the assumption of static background held up well. Nonetheless, some improvement is observed: both ID and MG have higher true positive rates than BS, and both have a lower false positive rate. Both algorithms are doing more sophisticated processing, and they have frame rates lower than did BS, though still higher than the BC and HT methods.

In the outdoor case, we see a larger difference. The true positive rate rose considerably, nearly doubling in the case of MG over BS. The false positive rate dropped modestly for ID over BS, but dropped off more dramatically for MG, falling to less than half what the naive background subtraction had had. Again, the frame rates are lower, representing the effect of the additional processing performed on each frame.

4 EVALUATION

The main factor affecting the success of background subtraction methods is the accuracy of the background model. Automated systems can make two types of errors: false negative (the object is present, but said to be missing) and false positive (the object is absent, but a spurious detection happens). If we can reduce the occurrences of false positives, we can be more aggressive in what we consider acceptable, and so reduce false negatives; alternatively, with a low rate of false positives, we can perform interpolation on the positives we obtain and have high confidence that we are not including spurious data points. We therefore investigated the sources of false positives.

We manually examined the three videos which exhibited the highest rates of false positive and tried to characterize the types of errors that were made. Possible sources of error included (i) confusion with similar objects; (ii) noise; and (iii) illumination changes, especially shadows. Table 5 shows the results. The overwhelming majority of errors arise from noise.

Table 5: Summary of sources of false positives.

Problem	Occurrence
Similar objects	12%
Noise	83%
Shadows	4%
Other	< 1%

Because noise is the main factor causing false positives, we are interested in seeing how many of the ball candidates after background subtraction are due to noise. We randomly selected two videos and performed background subtraction using each of our three methods; then, we looked at each frame after the background subtraction and manually counted how many regions were generated by noise. The results of this endeavor are reported in Table 6.

Table 6: Summary of number of noise regions.

Method	Noise regions per frame
BS	16.3
ID	15.8
MG	15.4

The number of noise regions is highest for BS, and lowest for MG: on average, MG had almost one candidate per frame less than BS. This has implications both for speed (fewer candidates means less processing, as fewer shape features need to be computed) and for accuracy (since there is no chance of a false positive if a candidate is not presented). However, the number of noise regions is still high. These results suggest that there is still room for improvement in the background subtraction phase.

We now report one final evaluation. We are able to achieve tennis ball tracking rates of around 90% with fast and relatively simple algorithms. The question remains: is 90% a good tracking rate, or can we reasonably expect to do better? To obtain data on which to frame a response, we manually marked the positions of the tennis ball on frames from 30 minutes of video of a Wimbledon tennis match, about 32400 frames in all. Our algorithms were not able to be applied to this video, both because there is no background footage available, and (more importantly) because we have assumed a static camera. The Wimbledon video is composed from multiple cameras that are controlled by expert cameramen. The editor carefully selects video sequences that provide high visibility for the ball to create the ideal viewing conditions for the viewer. We thus believe these videos give us best-case scenarios for ball visibility. Despite the careful crafting of the video, from the frames we looked at, the human eye could determine the tennis ball's location about 92% of the time. The remaining 8% of the time, the tennis ball is difficult to locate. A sample frame from the Wimbledon video appears in Figure 4; an example of a frame in which the ball was present with poor visibility is shown in Figure 5.

Of course, humans watching a tennis match can

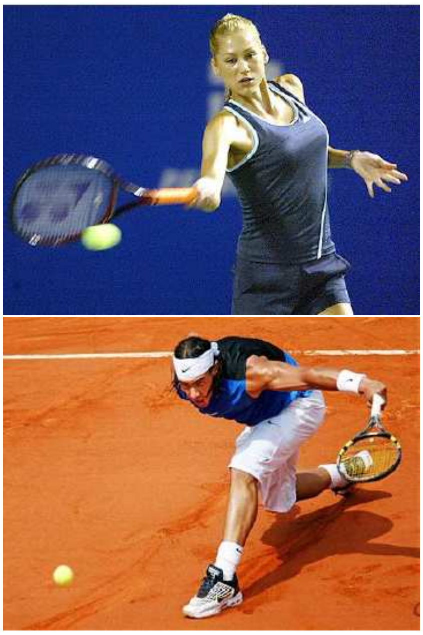

Figure 4: Sample frames from the Wimbledon video. In these frames, the ball is clearly visible.

achieve recognition rates higher than 90%.This is because they use additional cues to track the tennis ball. They use trajectory information to predict the path of the ball; this is likely the most important source of information not used by our methods. Also, human viewers can obtain secondary cues about the position of the ball based on the reactions of the professional tennis players, who are able to track the ball extremely well. It is unlikely that such information will be used by computer vision systems in the near future. Cues of this type often enabled us to locate the ball in the frames we labeled “poor visibility”, such as the first image in Figure 5, but it is not feasible even for the human eye to locate the ball with high confidence from the single frame alone, despite all the context information and object recognition ability that humans bring to the problem. We should comment also that color information enhances the ball visibility in the second image in Figure 5, but color information was not available to our algorithms, owing to our use of a high-speed monochrome camera.

Based on this experiment, we believe that our algorithms are performing quite creditably. We were able to achieve a recognition rate of 90% in the indoor case, and while the human eye remains demonstrably superior, we have gotten closer than previous methods.

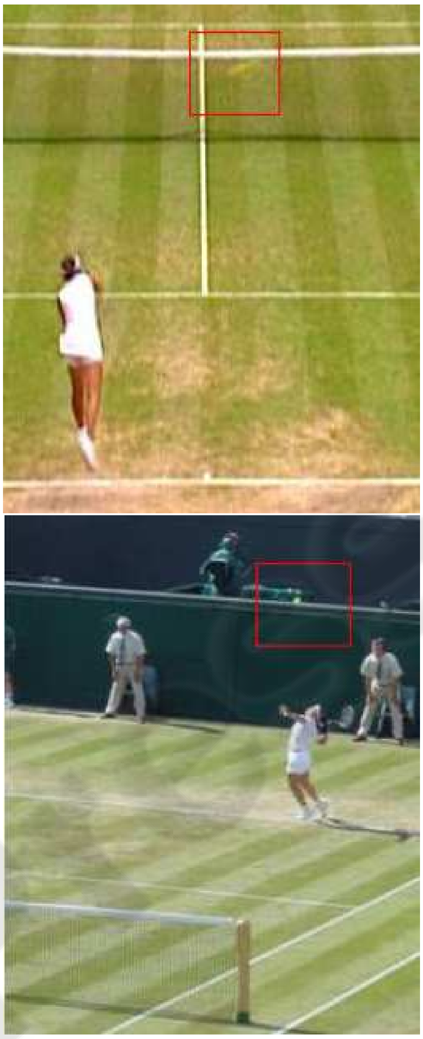

Figure 5: Examples of frames in which the ball is not clearly visible. The red squares indicate the ball’s location.

5 CONCLUSIONS AND FUTURE WORK

Our methods were able to achieve a true positive rate of about 90%, with a false positive rate of only about 1%, in the indoor environment. This compares with a true positive rate around 30% and a false positive rate around 60% to 70% for the technique of boosted classifiers and the Hough transform-based technique.

The outdoor case saw considerably more variation. Our techniques had between 20% and 40% true positive and 25% to 60% false positive. Boosted classifiers was able to achieve about 30% true positive and 50% false positive, while the Hough transform had only about a 4% true positive rate and more than 75% false positive rate. We attribute the poor performance of all techniques to the rapidly changing background caused by active weather.

One area for future work is to investigate deployment of the tracking system in conjunction with a system that depends on knowing the location of the tennis ball. Applications like robotic tennis-partner and player training can benefit from real-time tracking of the tennis ball. Our algorithm can be easily deployed for real-time tennis applications by using 4 to 6 cameras watching different parts of the court. Instead of running 4 (or 6 depending on the number of cameras) instances of the ball tracking algorithm we would have 2 cameras monitoring the ground at any given instance and when the ball leaves the field-of-view of one camera the next camera takes over the tracking. To further improve the processing speed, we can include various predictive tracking strategies like Kalman filtering.

We chose parameter settings (such as the aspect ratio and expected size of the ball) manually for each video. In the future, we would like to explore automatic parameter settings, perhaps from a few sample frames with the ball's position marked manually. In addition, we presently use static parameter settings for an entire video; this worked well for the short videos we tested with, but in future, we may want to allow dynamic parameters so that very long video sequences will work.

We have not used much trajectory information in discriminating among ball candidates, although trajectory is critical to how humans perform ball tracking. Future systems will need to incorporate trajectory information into their scene analysis.

Finally, both our image differencing technique and our mixture of Gaussians technique gave some improvement over simple background subtraction. We can consider combining MG and ID in future.

ACKNOWLEDGEMENTS

Thanks go to the IMG lab and the HCI lab at the University of Saskatchewan.

This work was supported in part by NSERC RGPIN 299070-04.

REFERENCES

Assfalg, J., Bertini, M., Colombo, C., and Del, B. (2002). Semantic annotation of sports video. *Proceedings of IEEE Transactions on Multi-Media*, 9(2):52–60.

D'Orazio, T., Ancona, N., Cicirelli, G., and Nitti, M. (2002). A ball detection algorithm for real soccer image sequences. In *ICPR '02, International Conference on Pattern Recognition*. IEEE Computer Society.

Han, M., Hua, W., Xu, W., and Gong, Y. H. (2002). An integrated baseball digest system using maximum entropy method. In *Proceedings of ACMM 2002*, pages 347–350.

Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *ICIP'02, International Conference on Image Processing*, pages 900 – 903.

Pingali, G., Opalach, A., and Jean, Y. (2000). Ball tracking and virtual replays for innovative tennis broadcasts. In *ICPR '00, International Conference on Pattern Recognition*. IEEE.

Rano, I., Raducanu, B., and Subramanian, S. (2004). Human presence detection and tracking for a concierge robot. In *Proceedings of IFAC Symposium on Intelligent Autonomous Vehicles 2004*.

Ren, J., Orwell, J., Graeme, A., and Xu, M. (2004). A general framework for 3d soccer ball estimation and tracking. In *ICIP '04, International Conference on Image Processing*, pages 1935–1938.

Shapiro, L. and Stockman, G. (2001). *Computer Vision*. Prentice-Hall, Upper Saddle River.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *CVPR' 99, Computer Vision Pattern Recognition*, pages 246–252.

Sudhir, G., Lee, J., and Jain, A. K. (1998). Automatic classification of tennis video for high-level content-based retrival. In *CAIVD'98, International. Workshop on Content-Based Access of Image and Video Databases*, pages 81–90.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR '01, International Conference on Computer Vision and Pattern Recognition*, pages 511–518.