A Mobile Augmented Reality System to Enhance Live Sporting Events

Samantha Bielli
Department of Computer Science
SUNY Oswego
Oswego, NY 13126
samantha.bielli@oswego.edu

Christopher G. Harris
Department of Computer Science
SUNY Oswego
Oswego, NY 13126
christopher.harris@oswego.edu

ABSTRACT
Sporting events broadcast on television or through the internet are often supplemented with statistics and background information on each player. This information is typically only available for sporting events followed by a large number of spectators. Here we describe an Android-based augmented reality (AR) tool built on the Tesseract API that can store and provide augmented information about each participant in nearly any sporting event. This AR tool provides for a more engaging spectator experience for viewing professional and amateur events alike. We also describe the preliminary field tests we have conducted, some identified limitations of our approach, and how we plan to address each in future work.

Author Keywords
Augmented reality; object recognition; computer vision; sporting event analytics

ACM Classification Keywords
H.5.1 Multimedia Information Systems: Artificial, augmented, and virtual realities; H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Worldwide, sports broadcasts via media such as internet and television are some of the most widely viewed events. Broadcast media offerings provide coverage with augmented statistics on the players and teams involved. Unfortunately, spectators in attendance at the event as unable to obtain this augmented information that can enhance their enjoyment of the game. Likewise, the number of spectators at amateur and casual league sporting events far outnumber those attending professional ones, yet there is little ability to augment the viewing experience with pertinent statistical information on amateur sports. Sports coaching and scouting is a $7 billion dollar industry, employing nearly 245,000 people in the United States alone. Job growth is likely to increase faster than average over the next 10 years. Nearly 98% of jobs involve the coaching and scouting of amateur athletes for which statistical information is not readily available. Having a way to assess and annotate each player's performance can provide more reliable information, improving the efficiency of identifying and training the best team athletes.

Last, the AR tool described here can make it easier for amateurs and professionals alike to monitor their performance improvement on the sports field. Players in amateur leagues can have sports card-like statistics recorded for them and displayed for any interested spectators to utilize. Likewise, outstanding amateur players can be recognized for their performance. A number of previous studies (e.g., [2, 4]) have demonstrated that the knowledge of the existence of information raises the level of performance of athletes who are being monitored. This observation effect can likely improve player performance, even at the amateur level.

Figure 1 Augmented information superimposed on the original image taken by the spectator

BACKGROUND
A number of researchers have proposed systems to obtain information from television, but were not designed for real-

AH '15, March 09 - 11, 2015, Singapore, Singapore
Copyright 2015 ACM 978-1-4503-3349-8/15/03...$15.00
http://dx.doi.org/10.1145/2735711.2735836

time observation of players by spectators. For example, Han et. al. developed a real-time AR system for court-based sports in [3]. Although it can provide a player's viewpoint in near-real-time, it is not designed to provide augmented information on players. In [6], Lu et. al. examine tracking and identifying players through movement, but it requires a priori training data, which may not be practical for most amateur sports events.

A number of other sports-related AR systems have been proposed, but must are tied to individual sports. Andrade et. al. illustrated how region adjacency graphs can be used to track soccer players across moving images, including detection of alphanumeric information in [1], but their methods have not been tested in the field. Lee et. al. proposed an AR system for baseball in [5], but this obtained player information based on the field position. Baseball's heavy dependency on a player's field location limits its applicability to other sports. In [7], Mahmood et. al. developed a player detection and identification application that can detect players by facial recognition; however, there are some limitations to their approach, such as the need to have players heads to be exposed and to appear frontal or near-frontal, which limit its capabilities in many sports. We know of no other AR system that can be used for on-site augmentation of sporting events on mobile devices across a variety of sports as the application presented here.

APPROACH

Our application involves recognizing both number characters and colors from an image taken by the spectator on a mobile device to produce an image similar to that in Figure 1. We use still images for two reasons. First, spectators are likely to have difficulties holding mobile devices steady without accidentally shaking the device while simultaneously interacting with the augmented contents obtained from video frames. Secondly, still images have higher resolution than image frames obtained from video sequences. Thus, still images provide a more accurate rendering of player information.

Our application was written in Java and XML and was created for use on Android devices and was tested on a Samsung Galaxy tablet. At a high level, the application prompts the spectator to take a picture using the device's camera, then progressively searches for characters and finds the most prominent color in the area of interest, and selects the resulting number-color combination from the available teams provided in a SQLite database.

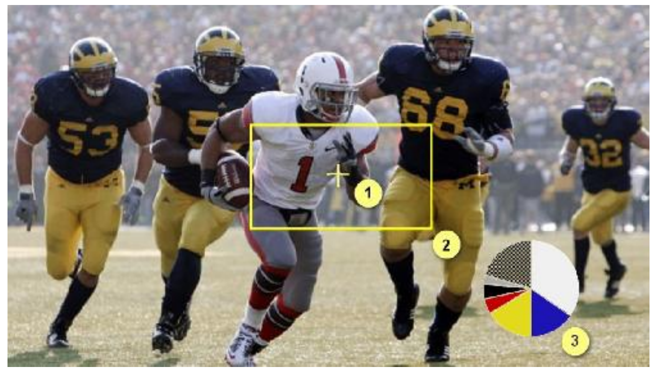

Figure 2. Steps taken by our algorithm. Starting from the image's center (1), we find the field of interest (2) and determine the candidate colors (3).

Application Flow Diagram

Figure 2 indicates the steps taken by our algorithm. Figure 3 describes the information flow used with our AR application. Before a match, an administrator would first select the jersey colors by capturing an image of each jersey. This provides information about jersey colors and associates it with team names. This information is then stored in a database that is either cloud-based or local (e.g., on a single device or network). Thus, statistics and information on each team and its players, can be made readily available. This data can be updated in real time by one or more people assigned to this task. This method allows for augmentation information to be captured for amateur sports where publicly available statistics are not available.

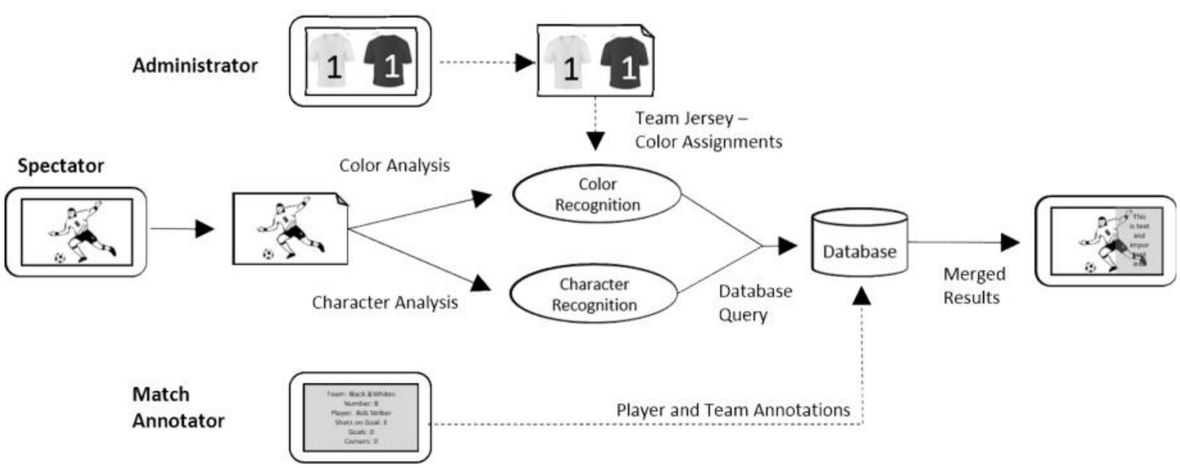

Figure 3. Flow diagram of the AR application

Once the teams are selected and jersey color is assigned to each team, spectators can obtain information through the built-in camera function of their tablet or smartphone. The application begins by prompting the spectator to take a photograph using the native camera application on the mobile device, as well as an illustration of how to do so. The spectator then utilizes the device's camera to capture and save the photograph so that it can be processed.

Once the spectator saves the image, the application processes it for color, and stores the name of the candidate jersey color. Then the image is processed for characters, and the candidate jersey number is stored as a text string. These results are then used to query the SQLite database; if an entry matches, the data is displayed, and if there are no matching entries, a message indicating so is displayed. In both cases, the original image is displayed to the spectator.

Character Recognition

Our application recognizes characters in images using the Tesseract Optical Character Recognition (OCR) API. It utilizes the Android Native Development Kit in order to provide a Java API for accessing natively compiled APIs Tesseract and Leptonica [9]. Since our application is searching specifically for numbers on sports player's jerseys, certain procedures are taken to make correct recognition more likely.

First, we find the center of the image; from our observation, spectators usually center their camera on the player for which they would like additional information. Second, using this center-weighted image, we progressively increase the size of the rectangle, which we call our *field of interest*, until a number is recognized in the image. We provide a buffer around the recognized number so that adjacent numerals on the same jersey are also included.

Pattern matching is used to limit the recognized text in the field of interest to the digits zero through nine. Non-numeric characters are replaced with spaces, and the text string is trimmed to get rid of these extraneous spaces. This ensures that in situations where numbers may have similar forms to letters, such as '4' and 'A' or '5' and 'S', that the API will not return the letter as the recognized character. Secondly, the Tesseract API is equipped with page segmentation modes, which allow for the programmer to select the type of text to be processed. In our application, the page segmentation mode is set so that the image is treated as a single word. This lets the API focus on the single number on the player's jersey without searching the image for multiple lines of text, making it faster to process.

The Tesseract OCR works in a step-by-step process, working from the center and progressively increasing the area of interest. Concurrently, the area of interest is broken down into smaller components until single characters have been recognized within the area of interest. The field of interest is examined for *components*, different colored areas which are assumed to be different shaped text areas. In the first step, the outlines of the components in the image are stored, and the outlines of nested components are also analyzed. By storing and inspecting nested levels of components instead of just the top level, Tesseract is able to detect other color combinations besides just black text on a white background. These "blobs" of text are divided into lines, with algorithms based on fitting quadratic splines to detect skewed text baselines [9].

While these algorithms are intended to allow for better recognition of text scanned from a book, they also work well to recognize numbers on jerseys where the fabric is curved over the player's body or at an unusual angle. After these lines are detected, the Tesseract OCR detects where there are larger spaces between characters to separate individual words. These words are then divided into characters, which are recognized by a classifier trained on data containing several samples of 94 different characters from eight fonts [8].

Color Recognition

Next, we determine the predominant colors in our field of interest. We calculate the percentage of colors that appear in our field of interest and compare them against the set of jersey colors stored earlier. Those with a percentage above a certain threshold are considered *candidate* colors.

We recognize color in images using the Palette class from the android.support.v7 API. This class has methods to retrieve the most vibrant color swatch from the image. Because the player's jerseys are usually the brightest colors on the field, this method works well to recognize candidate jersey colors. For situations in which players have dark or muted color jerseys, we utilize methods to recognize these subdued color swatches instead.

In our system, these candidate colors are returned as RGB integers. The integer is compared to a simple list of RGB values obtained in the initial setup for the event, using the mean squared error **(**MSE**)** of the red, green, and blue components of each candidate swatch. The colored swatch from the list with the lowest MSE is returned as the color of the player's jersey for the sporting event. These are processed in order from largest to smallest percentage of the field of interest.

Database

Once the jersey number and prominent color are extracted from the image, they are compared to a database containing statistics on sports players. The database is created with SQLite, and contains sports statistics stored in the context of different sports events, as well as each player's number, and the color of the jerseys their team has been assigned for the sporting event. The application finds the entry in the database for which the number and color both match and displays the player's name, team, and other information augments the image of the player taken by the spectator.

FIELD STUDY EXAMINATION

We conducted a field study examination of amateur sporting events including basketball, soccer, and ice hockey. We also used the application on television sporting events in order to find shortcomings with our application and discover its merits. With an accuracy of 90%, which we define as the percentage of images identifying both the team and the player's number in the database within 5 seconds of submitting the image. We found that the preliminary version of our application works in most situations but the accuracy could be improved by using a color space such as CIE-LAB or HSV, which is more effective than the use of the RGB color space in our study.

Although the application is remarkably robust at isolating colors from similarly-colored jerseys, as seen in Figure 4, this is more likely to be an issue with amateur teams, where jersey colors may not be distinct.

Misreads, where information on the wrong player was presented, were infrequent but could be improved. Most were due to blurred and partially obscured numbers; if the bottom right of an '8' or a '6' is obscured, it is hard to distinguish it from a '9' or a '5', respectively. Also, if the fold of the jersey obscured part of a number, the wrong information would be returned. Using additional textual information, such as the names on the back of the jersey, can help as an additional part of the query to reduce these errors, but this information is not always available Another issue is displayed in Figure 5, where the predominant color in the field of interest is misread and the wrong jersey color is chosen. We are looking at methods to reduce the size of the field of interest to become tighter around the number, which should reduce this type of error.

Figure 4. A rendering of how our system would focus on the center player. By focusing on the center and working outward, our system obtained reliability rates of 90%

CONCLUSION

We have described an Android-based application used by spectators to provide augmented information on players in sporting events. The application has three distinct stages – first it detects a number and color from a player's jersey; second it sends this number-color combination as a database query; last, it retrieves the data and superimposes the augmented information on the image. We have also briefly described a preliminary field examination where we examined the limitations of our application. Last, we have discussed how we will address these in future work.

Figure 5. An illustration of a limitation of our approach. The predominant color in our field of reference is blue, whereas our focus is on a player on the other team.

REFERENCES

1. Andrade, E. L., Khan, E., Woods, J. C., & Ghanbari, M. (2003). Player classification in interactive sport scenes using prior information region space analysis and number recognition. In *Proc. International Conference on Image Processing*, 2003. (Vol. 3, pp. III-129). IEEE.
2. Escarti, A., & Guzman, J. F. (1999). Effects of feedback on self-efficacy, performance, and choice in an athletic task. *Jour. of Appl. Sport Psych*, 11(1), 83-96.
3. Han, J., & Farin, D. (2007, September). A real-time augmented-reality system for sports broadcast video enhancement. In *Proc. ACM Multimedia* (pp. 337-340).
4. Iso-Ahola, S. E. (1995). Intrapersonal and interpersonal factors in athletic performance. *Scandinavian jour. of medicine & science in sports*, 5(4), 191-199.
5. Lee, S. O., Ahn, S. C., Hwang, J. I., & Kim, H. G. (2011). A vision-based mobile augmented reality system for baseball games. In *Virtual and Mixed Reality-New Trends* (pp. 61-68). Springer.
6. Lu, W. L., Ting, J. A., Little, J. J., & Murphy, K. P. (2013). Learning to track and identify players from broadcast sports videos. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7), 1704-1716.
7. Mahmood, Z., Ali, T., Khattak, S., Hasan, L., & Khan, S. U. (2014). Automatic player detection and identification for sports entertainment applications. *Pattern Analysis and Applications*, 1-12.
8. Smith, R. (2007). An Overview of the Tesseract OCR Engine. In ICDAR (Vol. 7, pp. 629-633).
9. Theis, R., "tess-two" Updated 2014. Retrieved from https://github.com/rmtheis/tess-two