Video Scene Extraction Tool for Soccer Goalkeeper Performance Data Analysis

Yasushi Akiyama
Saint Mary's University
Halifax, Nova Scotia
Yasushi.Akiyama@smu.ca

Rodolfo Garcia
Saint Mary's University
Halifax, Nova Scotia
Rodolfo.Garcia.Barrantes@smu.ca

Tyson Hynes
Kilo Communications
Halifax, Nova Scotia
tyson@gkstopper.com

ABSTRACT

We will present a new approach for the scene extraction of sport videos by incorporating the user-interactions to specify certain parameters during the extraction process, instead of relying on fully automated processes. It employs a scene search algorithm and a supporting user interface (UI). This UI allows the users to visually investigate the scene search results and specify key parameters, such as the reference frames and sensitivity threshold values to be used for the template matching algorithms, in order to find relevant frames for the scene extraction. We will show the results of this approach using two videos of youth soccer games. Our main focus in these case studies was to extract segments of these videos, in which the goalkeepers interacted with the balls. The resulting videos can then be exported for further player performance analyses enabled by Stopper, an app that tracks keeper performance and provides analytical data visualizations.

KEYWORDS

UI tool for spatial and temporal data analyses, video analysis, video segmentation algorithms, interactive data processing, image processing, template-matching

ACM Reference Format:
Yasushi Akiyama, Rodolfo Garcia, and Tyson Hynes. 2019. Video Scene Extraction Tool for Soccer Goalkeeper Performance Data Analysis. In *Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019* , 9 pages.

1 INTRODUCTION

In this paper, we will present a new approach to extracting segments of sport videos by incorporating the user-interactions to specify certain parameters during the extraction process, instead of relying on fully automatic approaches. When coaches and players review videos of their own games or those of competing teams for analysis purposes, they typically fast forward through game footage until they find the segments that show important plays within these games. For example, Stopper [31] is a mobile app that tracks soccer goalkeeper performance and provides analytical data visualizations. The users of this app can record the data while watching live games or retrospectively watching the recorded videos. While a single soccer game is typically 90 minute long, the amount of time a goalkeeper is involved in plays is significantly less than the full duration of a game. Thus, it would be ideal if a previously edited shorter version of the video that only shows the relevant plays (i.e., video highlights) is provided for the users of Stopper so that they will not need to skip irrelevant parts within a game.

While some video segmentation and summary generation algorithms exist and work in certain domains [9, 15, 26], to our knowledge, there is no approach that can directly be applied to our problem domain. Our system provides an intuitive UI that allows the users to specify certain areas of a video frame to be used for the template-matching algorithms. The system will then find all the relevant frames based on the template matching results. The tool also allows the users to select the sensitivity of template-matching so as to control how many false-positive frames are to be included in, or false-negative frames excluded from, the resulting video highlights.

The rest of the paper is organized as follows. We will first briefly describe Stopper in Section 2 to give more context of the current research. In Section 3, we will discuss the overview of the past research and approaches to addressing similar problems. Section 4 will describe the proposed approach, together with the UI that is designed to provide certain user interactions to select several parameters. Section 5 will show the results of the case-studies to test our approaches in different settings. Our main focus was to extract segments of these videos, specifically when the goalkeepers interacted with the balls. These videos can then be exported for further player performance analyses enabled by Stopper. We will finally provide conclusions and discussions on the implications for the future work in Section 6.

IUI Workshops'19, March 20, 2019, Los Angeles, USA

2 STOPPER

Stopper (shown in Fig. 1) is a mobile app developed to record and visualize soccer goalkeeper game performance data. Users can track the data in five key performance areas: (1) Saves (a shot directed towards the goal that is intercepted by the goalkeeper), (2) Goals Against (a shot that passes over the goal line), (3) Crosses (a ball played into the centre of the field), (4) Distribution (a pass by the goalkeeper using either their hands or feet), and (5) Communication (how the goalkeeper verbally and in gestures supports and organizes their team), which collectively provide a framework for analysing goalkeeper strengths and weaknesses.

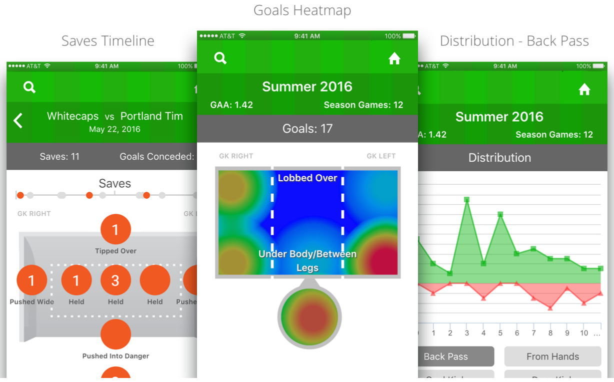


Figure 1: Stopper is a mobile app developed to record and visualize soccer goalkeeper game performance data.

Commonly used metrics such as Goals Against Average (GAA), Save Percentage (Sv%), and Expected Goals (xG) provide limited correlation to goalkeeper ability [7, 30]. As a result, analysing individual goalkeeper performance separately from the overall team performance carries an inescapable degree of subjectivity [6, 23]. The resulting data based on Stopper's five key components can establish a more comparative benchmark for individual player performance, and it is less likely influenced by the quality of the defensive play of their own team or the attacking capability of opposing teams, compared to the traditional performance measurements.

While Stopper's analytical data visualizations in these performance areas can help understand the overall keeper performance, corresponding videos showing the tracked actions will provide a crucial component for more detailed analyses as a training and coaching tool. Currently, the users first log the goalkeeper performance using Stopper while watching the game. Once the data is recorded, Stopper uses timestamps of the goalkeeper actions logged during a game to generate video snippets for individual goalkeeper actions. Our focus in the current research is somewhat the reverse of this process. That is, we will first extract video segments that only contain the goalkeeper interactions and provide the users with the extracted videos. In this way, they will not need to watch the entire 90 minute of the soccer game in order to log the performance data.

3 RELATED WORK

Automatic Video Segmentation

There have been studies in the related problem domains. One group of research focused on video segmentation approaches, specifically for videos of sports. Oyama and Nakao [25] proposed an approach to identifying different types of plays (i.e., scrum, lineout, maul, lack, place-kick) in a rugby video based on the image analysis of player interactions. Li and Sezan [17] also proposed an approach to classifying different plays in sport videos, using broadcast videos of baseball and football. Ekin et al. [5] used the low-level analysis for cinematic feature extraction for scene boundary detection and scene classifications in soccer videos. A slightly different approach was proposed by Baillie and Jose [1], using an audio signal analysis to detect certain scenes by incorporating Hidden Markov model classifiers in their algorithm. All these studies utilize broadcast videos, often of professional sports, that were shot from multiple cameras positioned at different locations in stadiums. Thus, switching between scenes, or cuts, often gave sufficient cues for these approaches to detect different plays in these games. Since our current work is focused on the analyses of videos of youth players, the videos are usually recorded by a single camera, positioned to align with the centre line of a field. Different plays are recorded by panning the camera horizontally so there are no "cuts" to be detected in the recording.

Video segmentation and scene detection approaches outside of the sport video domain have also been investigated [21, 28]. These approaches detect scenes/segments based on cuts that typically produce abrupt changes in video boundaries or on video transitions that exhibit certain characteristics in visual parameters such as colour and brightness changes. However, these approaches suffer from the same issue as the above ones that capitalize on cuts and switching between cameras. Further, some approaches can work well in certain sports (e.g., detecting scrums in rugby that have distinct player interactions/formations) but are not straightforwardly applicable to other sports. For example, soccer games typically have variety of plays that do not necessarily exhibit visual patterns in player interactions, perhaps with only very few exceptions such as corner kicks or penalty kicks. In the case of audio analysis [1], the approach relies on a large number of spectators to generate sufficiently salient audio features. Most youth games may not even have any spectators or audience (e.g., practice games and scrimmages) to generate audible cues to detect certain plays. Thus, none of the above approaches will work well in our specific problem space.

Object Detection and Template Matching Algorithms

Approaches to the object detection in images and video can broadly be devided into four categories [29], feature-based, motion-based, classifier-based [11, 18], and template-based. Feature-based object detection utilizes object features such as shapes [20] and colours [16]. These approaches, however, did not work well in our preliminary investigation when we tried to keep track of players (e.g., keepers) or a soccer ball, due to several potential factors such as objects (e.g., humans) often changing shapes during the play, colours of uniforms being too similar to the background colours (e.g., green jerseys on green grass), and frequent occlusions of target objects. Motion-based detection approaches often use static background reference frame(s) and detect changes in the foreground by eliminating the background images [12, 14, 32, 33]. These approaches typically require static background images, but in our case, since the camera follows soccer balls, the background images keep changing, making it difficult for us to straightforwardly apply them to detect objects in videos. We also investigated the possibility of integrating some classifier-based approaches into our framework; however, we could not find suitable solutions that could detect frames with target objects, especially when there are not sufficient samples for the model training. Therefore, we used a simple template matching algorithm using the normalized cross-correlation [27] in our framework. As will be discussed in this paper, the object detection approach itself in our framework can be switched to another, potentially better, solution later. The focus of the current paper is to propose a generic framework of video segmentation and show the early results of this proposed approach.

To this end, we have observed that generic automatic video segmentation approaches can benefit from certain domain knowledge. For instance, the work such as done by Kim et al. [13] and Oude-Elberink and Kemoi [24] integrate the user-interaction for their object detection and tracking algorithms in the video. Our proposed approach also utilizes the user input in order to complement and improve the automatic video segmentation algorithms. We will now describe our approach in the next section.

4 PROPOSED APPROACH

Our proposed approach will work in five basic steps, interactively with the users' input. This section describes each of these steps in detail, together with the corresponding UI modules, a prototype of which is developed as a web-based application.

(1) The user uploads an original video and specify a reference frame

Users will first select a video, from which they want to create video highlights, using the provided interface (shown in Figure 2). They will then specify what we call a *reference frame*. Reference frames are the frames, in which they specify areas to be used to find relevant frames that contain certain objects or backgrounds. The UI tool allows the users to skip back and forth to find a frame that shows objects that most likely appear when the target actions occur. For example, if we are to find the segments that show the goalkeeper who is on the right-hand side of the pitch in action, then the user should select a frame when the camera pans to the right so that it includes the entire goal area (shown in Figure 3).

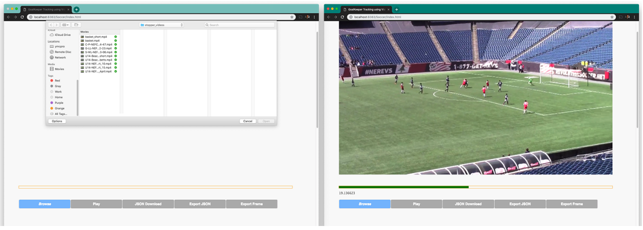


Figure 2: The UI tool allows users to select and preview the target video.

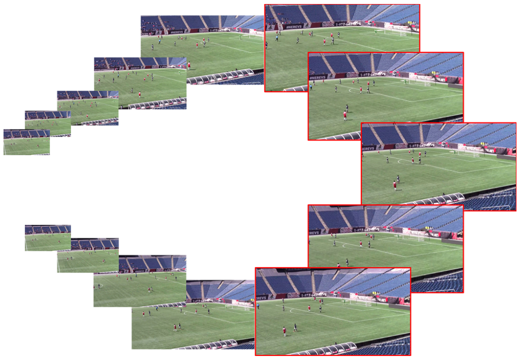
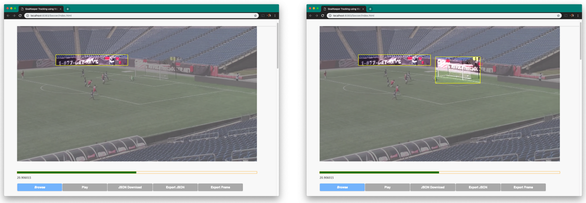


Figure 3: The film strip shows video frames, some of which are candidate reference frames (indicated by the red outline) to be used in the next step.

(2) The user selects reference areas to be used for the frame search

The next step is to specify areas of the reference frame that the users want to use for the relevant frame search. We call these areas *reference areas.* This step is necessary for reducing the chances of the algorithm detecting irrelevant frames due to the overall similarity of the video frames. For example, soccer videos often contain many frames that are considered similar by most similarity metrics due to the fact that certain background images such as bleachers and grass on the pitch appear in almost every single frame in the video. However, we do not want to include these background areas because they are too generic and are not great references in terms of finding relevant frames. Instead, we need to only include portions of the reference frame that display salient objects or features that can be used to identify relevant frames. For example, if the users are to find video segments that contain the goalkeeper's interactions, then they may choose reference areas that show the entire goal area and/or the goal itself. This step of reference area selection is depicted in Figure 4.



(a) After the user selected one reference area.

(b) After the user selected another reference area. The user continues to add as many reference areas as they wish to.

Figure 4: The selection of reference areas to be used for the relevant frame search algorithm.

(3) The system rates each frame in the original video with the relevance metric

For each frame in the original video, the algorithm will calculate its likelihood of containing each of the reference areas by using a template matching algorithm. It will repeat the process for all the reference areas and calculate the overall likelihood of the reference areas appearing in that frame, as an average likelihood for all the reference areas. The template matching algorithm that was employed in our case studies was provided in an open source library for computer vision and machine learning software and image processing, called OpenCV (Open Source Computer Vision Library) [2]. The function *matchtemplate* in this library calculates the cross-correlation [27] between a reference area and the target frame. Conceptually, it scans the target frame by sliding a reference area (i.e., the template) over the target frame pixel by pixel, while calculating the correlation of the two images at each location: the reference area and the portion of the frame underneath it. This process is depicted in Fig.5.

Let $C(x, y)$ be the cross-correlation of the two images at a pixel (x, y), $T(x, y)$ the pixel value of the target frame at (x, y), and $R(x, y)$ the pixel value of the reference area at (x, y), the metric is calculated by the following formula.

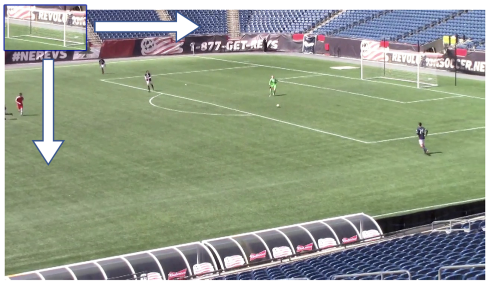


Figure 5: Each reference area is compared against an area in the target frame, by shifting it pixel by pixel, while calculating the correlation of the reference area and an area of the same size at each location within the target frame.

$$C(x,y) = \sum_{x',y'} \left\{ T(x',y') \cdot R(x+x', y+y') \right\} \tag{1}$$

Further, based on the general observation that a frame in the video may likely have different lighting/intensity based on factors such as camera angles and exposures, we will use the normalized cross-correlation to mitigate the lighting effects:

$$C(x,y) = \frac{\sum_{x',y'} \left\{ T(x',y') \cdot R(x+x', y+y') \right\}}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} R(x+x', y+y')^2}} \tag{2}$$

We calculate $C(x, y)$ for all the pixels given by Eq. 2, and then use the maximum value in $C(x, y)$ (i.e., the highest likelihood of the reference area matched in the target frame) as the relevance metric for this frame. We repeat this process for each reference area and calculate the overall likelihood of the reference areas appearing in the frame. The pseudocode of this entire step of calculating the frame relevance metric is described in Algorithm 1.

Algorithm 1 Relevant frame search algorithm

```
for each frame $f_i$ in original video do
    $sum \leftarrow 0$
    $n \leftarrow 0$
    for each reference area $a_j$ do
        $p_{ij} \leftarrow$ Likelihood of $a_j$ appearing in $f_i$
        $sum \leftarrow sum + p_{ij}$
        $n \leftarrow n + 1$
    end for
    $ave_i \leftarrow \frac{sum}{n} (n > 0)$
    if $ave_i > threshold$ then
        $relevantFrames.add(f_i)$
    end if
end for
```

We have experimented with other metrics commonly used for the template matching such as the sum of squared difference [10], but the normalized cross-correlation yielded the best results overall. While the current paper presents our entire framework of the video segmentation processes that can easily be facilitated by novice users, the template matching algorithm itself in our framework can also be replaced by others (e.g., [4, 19, 22]) and by incorporating certain machine learning algorithms such as deep learning models [3] that may potentially improve the accuracy of the template matching.

(4) The user selects a threshold value

The tool will now display the resulting relevance metrics from the previous step, and using this visualized data, the user can select an ideal threshold to be used for the next step. The UI allows the user to move the threshold line on the visualized relevance metrics data, so that they can control which section(s) of the original video to be included. The green dots in Fig. 6 indicate the frames to be included in the final extracted video highlights, while the frames indicated by the red dots will be excluded from them. Naturally, lowering the threshold may include false positive frames (i.e., irrelevant frames), while raising it may result in false negative frames (i.e., missed relevant frames).

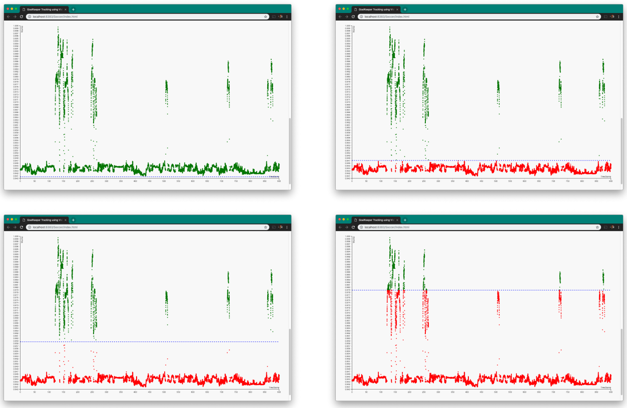


Figure 6: The user moves the line on the visualized relevance metrics to control the threshold. The frames indicated by the green dots are to be included in the final extracted video segments. In this particular example, the threshold used in the bottom-left plot effectively separates the two groups of data points.

This visualization tool is also synchronized with the video viewer. That is, as the user clicks on the data points in the data plot, the video viewer's time cursor is also moved to that particular point in time, allowing the user to visually inspect the corresponding plays in the original video. This interaction is depicted in Fig.7. Therefore, with this visual aid, the user will need to spend less time to scan the original video as it allows them to directly get to the frames that will likely include the keeper interactions with the ball. It is this interactive visual investigation of the video data that allows the users to minimize the time spent on searching for the relevant frames.

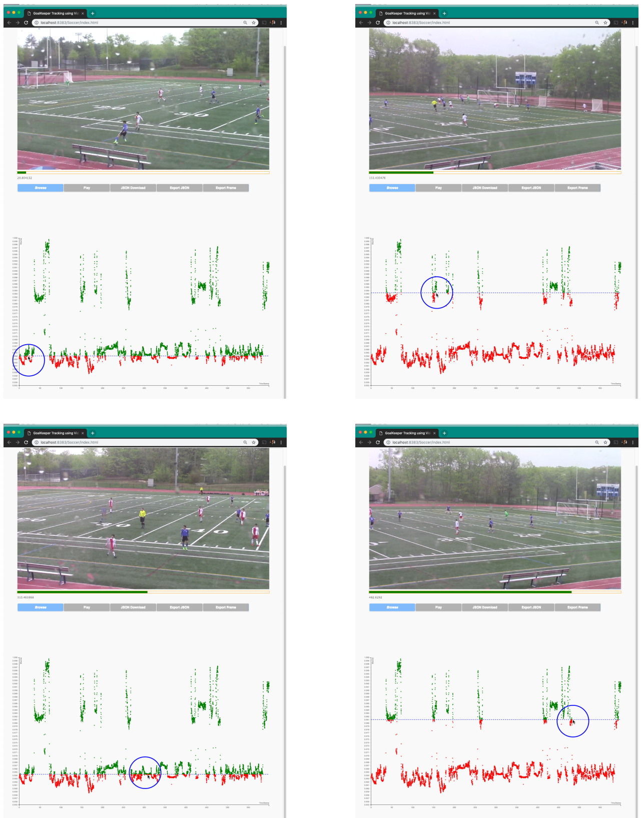


Figure 7: The corresponding video frames will be shown by clicking the data points in the visualization tool, allowing the users to interactively inspect the video based on the relevance metrics calculated by the tool. This interactive visual investigation of the video data presumably reduces the time spent to search for important plays in the video.

(5) The system extracts video segments with the frames with the relevance rating higher than a threshold value

The final step is to extract video segments that contain the relevant frames. Our current approach is to take all the frames with relevance metric values above the specified threshold (i.e., all the green segments shown in Fig. 6). The video segments will then be created by sequencing all these relevant frames.

5 CASE STUDIES

In the case studies, we used video footage from two US Soccer Development Academy league games. Both the videos were in the MPEG-4 (AAC, H.264 codec) format and had the dimensions of 1280 by 720 with the frame rate of 29.97 frames per second (fps).

(1) Video #1: Contains 15 minutes of a soccer game, with a relatively clear weather and fair lighting.
(2) Video #2: Contains 10 minutes of a soccer game, under the rainy condition with darker lighting.

Both the cameras were set a few metres above the ground so they were slightly looking down the pitch. They were secured on tripods and the panning mechanism was used to keep track of the ball, thus the videos only show a portion of the pitch at one time, and never the entire field at any given time. All the cases were run on a MacBook Pro (13-inch, 2018) with 2.3GHz Intel Core i5 CPU and 8GB 2133MHz LPDDR3 memory.

Case 1: Detecting the keeper interactions on the right-hand side of the pitch on Video #1, with the reference area containing the goal

In order for the template-matching to work, we will first need to choose a reference area(s) that will be unique and static in shapes and colours for the most part of the video. For example, choosing goalkeepers themselves as references do not typically produce ideal results as they move around while the shape of the object (i.e., a human) changes significantly. Also, in the videos that we used in these case studies, the keepers wore shirts with neon yellow and green colours, which often blended in with the green colour of the grass, potentially confusing the template-matching algorithm. Therefore, we chose a reference frame that shows the entire goal on the right-hand side of the pitch (shown in Fig. 8) and selected this goal as a reference area (shown in Fig. 9). After the visual inspection of the data, we used the correlation metric of 0.98 as the threshold to create the resulting videos.

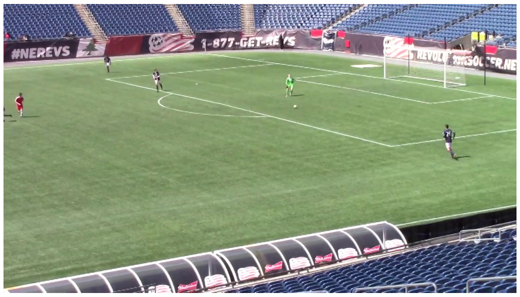


Figure 8: The reference frame that shows the entire goal on the right-hand side of the pitch.

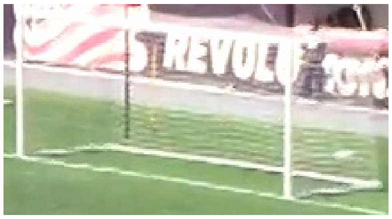


Figure 9: The reference area that shows the goal on the right-hand side of the pitch.

The results are shown in Fig. 10. Most of the anticipated frames were detected as relevant, with the highest relevance metric was indeed detected at the reference frame around at the 131st second. However, the algorithm missed some frames that should have been considered as relevant in terms of the plays in which the keeper was involved. For example, if you look at the frame at the bottom left in Fig. 10 that shows the play at 129 seconds into the video. This play was right before the reference frame and the keeper is actually holding the ball. However, this and some of the other frames leading up to the reference frame were omitted from the relevance frames. This omission was in fact inevitable as the reference area clearly shows the entire goal while this frame at 129th second is missing the right side of the goal. One solution to include these frames is to lower the threshold, but it will also include irrelevant frames that appear earlier in the video. Therefore, while the template matching algorithm itself seems to have worked properly, we probably did not choose the most ideal reference frame/area(s).

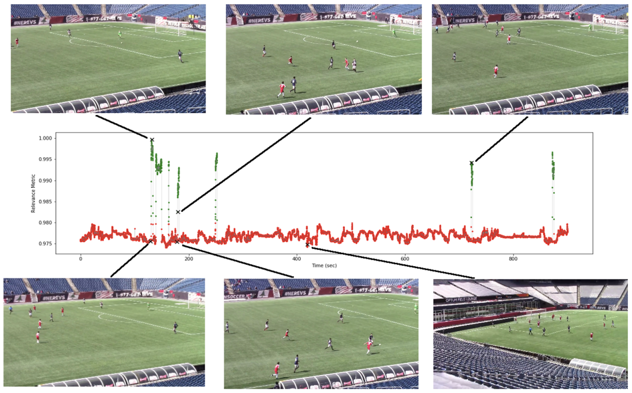


Figure 10: The results of Case 1.

Case 2: Detecting the keeper interactions on the right-hand side of the pitch on Video #1, with the reference areas containing both the goal and the unique background area

Given the above results, in addition to the goal, we also experimented with an additional reference area, which shows a unique background area shown in Fig. 11, thus we used both the goal as well as this unique background from the same reference frame to perform the template-matching.

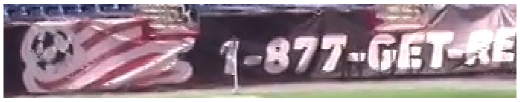


Figure 11: The reference area that shows a unique background area.

As shown in Fig. 12, this additional reference area improved the performance in that it included those frames (e.g., the top left frame shown in Fig. 12) that did not have the entire goal but were parts of the play, in which the keeper interacted with the ball. This result illustrates the importance of integrating the user input into these processes instead of relying on entirely automated approaches.

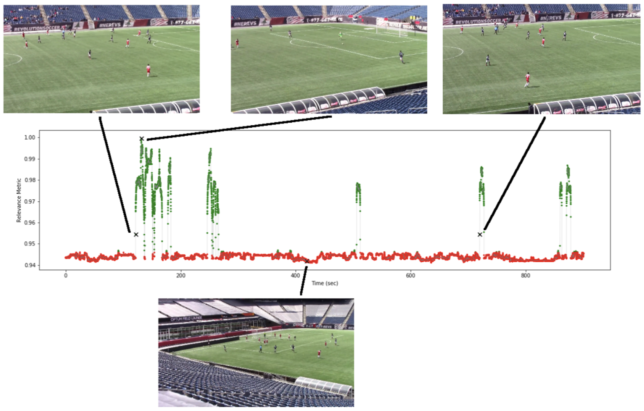


Figure 12: The results of Case 2 (using the threshold value of 0.946). The resulting video included frames even when the goal is not shown but they were parts of the relevant play.

Case 3: Detecting the keeper interactions on the right-hand side of the pitch on Video #2

We also tested with the video with some visible noise caused by the rain. The reference area used in this test is shown in Fig. 13.

As shown in Fig. 14, the expected relevant frames were still appropriately detected, despite that the visibility condition was not as ideal as the first two cases.

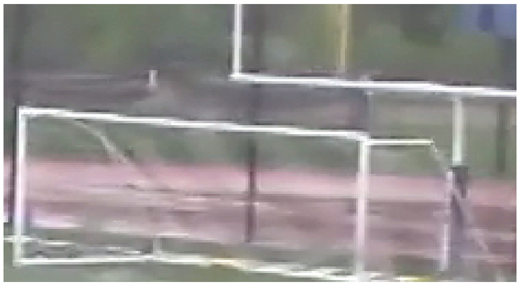


Figure 13: The reference area used to identify the relevant frames for the keeper on the right-hand side of the pitch.

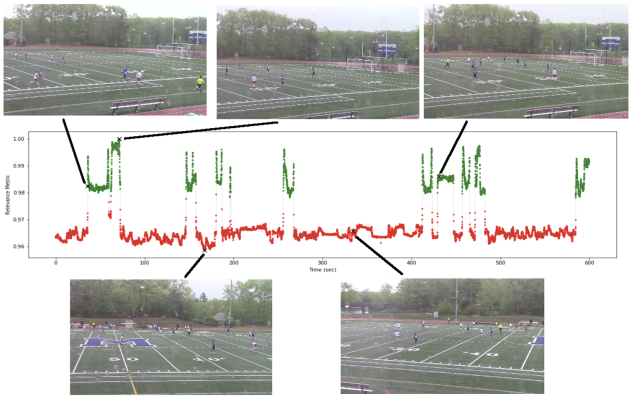
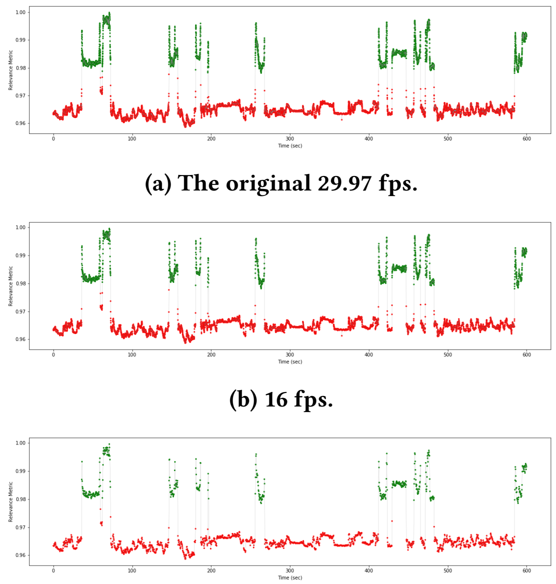


Figure 14: The results of Case 3 (using the threshold value of 0.978).

Case 4: Detecting the keeper interactions on the right-hand side of the pitch on Down-sampled Video #2

For all the above cases, we used the original frame rate of 29.97 fps and ran the relevant frame search algorithm on all the frames. However, typical plays of soccer do not require such a high frame-rate for our purposes, thus we experimented to first down-sample the original video to lower frame rates of 16 fps and 4 fps, in order to increase the efficiency of our approach. Once we identified the relevant frames, we used the original video to extrapolate the corresponding segments of the video. As seen in Fig. 15 that shows the results of comparing three different frame rates, it produced almost identical curves of the relevant metrics.

The results showed that this down-sampling significantly accelerated the process without affecting the overall results. To give the general idea of the processing time for the relevant metrics calculations, Table 1 shows the calculation time for Video #2 which was ten minute long. Based on this observation, the tool was able to calculate the relevant frames in about 1/16 of the length of the original video. Note that the calculation time itself of course can also be improved



(a) The original 29.97 fps.

(b) 16 fps.

(c) 4 fps.

Figure 15: The relevance metrics of the same video, using (a) the original 29.97 fps frame rate, (b) 16 fps, and (c) 4 fps. They all produced the similar curves as well as the extracted videos.

Table 1: The comparisons of the relevance metric calculation time based on the different frame rates.

Frame rate (fps)	Time (seconds)
Full	207.88
16	106.68
4	38.09

further in a few ways, for example, by calculating the frame relevance metrics in parallel, as the relevance metric for each frame does not depend on the other frames' results, by downsampling the video resolution, or skipping a number of pixels during the template matching instead of checking against every single pixel.

6 CONCLUSIONS AND FUTURE WORK

We proposed a new framework of semi-automatic video segmentation of sport videos, and the UI tool that implements the proposed approach. Instead of relying on a fully automatic method, our approach consists of five fundamental steps that integrate the user input and knowledge to help reduce potential errors. The provided UI tool allows the users to easily select a reference frame and reference areas that are used to detect relevant video frames that contain target player actions, and visualizes the relevance metrics to determine the optimal threshold value for the video extraction. The users can interactively investigate the corresponding video segments capitalizing on this visualization tool, thus likely spending less time to search for important plays in the videos. The case-studies showed that our approach worked well with certain videos, but there are several factors that affected the performance of the approach and we are currently working to improve it in multiple aspects.

One such aspect is the further investigation to compare the template-matching and object detection algorithms. As discussed throughout the paper, there are some algorithms that may potentially improve the accuracy of the tool. Some algorithms may be more suitable for certain conditions such as videos with specific image backgrounds or lighting conditions. Some of the predictive models such as those utilizing the deep learning algorithms may potentially be an option once we obtain enough video data in order to train the models. In this case, a potential approach is to first run a clustering algorithm on videos based on certain parameters such as background types and lighting conditions, and then create separate models for each of those types.

As well, the threshold is currently determined by the users before the system renders the video, but this may potentially be estimated, for example, by integrating known threshold estimation methods [8]. Finally, the framework itself can potentially be applied to other similar types of sports such as basketball, rugby, and field hokey. Our approach will of course need to be modified to accommodate differences in games. For example, one experiment that we conducted with a basketball video revealed that, while the tool did work relatively well detecting any plays near the hoop, since the game moves much faster than soccer, there should be some sort of mechanism to include frames leading up to, and after those plays to be included to show more complete sequence of actions. Solutions to these new challenges posed by other types of sports will likely lead to further improvement of the tool in general.

7 ACKNOWLEDGEMENT

This research was supported by Natural Research Council (NRC) Canada Industrial Research Assistance Program and Nova Scotia Business Inc. Productivity and Innovation Voucher Program. A special thanks to Jyothi Sethi, an M.Sc. student at Saint Mary's University, who contributed her skills to implement the UI tool.

REFERENCES

[1] M. Baillie and J. M. Jose. 2004. An Audio-Based Sports Video Segmentation and Event Detection Algorithm. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*. 110–110.

[2] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[3] Davit Buniatyan, Thomas Macrina, Dodam Ih, Jonathan Zung, and H. Sebastian Seung. 2017. Deep Learning Improves Template Matching by Normalized Cross Correlation. *CoRR* abs/1705.08593 (2017).

[4] Luigi Di Stefano, Stefano Mattoccia, and Federico Tombari. 2005. ZNCC-based Template Matching Using Bounded Partial Correlation. *Pattern Recogn. Lett.* 26, 14 (Oct. 2005), 2129–2134.

[5] A. Ekin, A. M. Tekalp, and R. Mehrotra. 2003. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing* 12, 7 (July 2003), 796–807.

[6] Garry Gelade. 2014. Evaluating the ability of goalkeepers in English Premier League football. *Journal of Quantitative Analysis in Sports* 10, 2 (2014), 279–286.

[7] Sam Gregory. 2015. GoalkeepersâĂŹ save percentage an unreliable stat. http://www.sportsnet.ca/soccer. Accessed: February, 2019.

[8] Bruce E. Hansen. 2003. Sample Splitting and Threshold Estimation. *Econometrica* 68, 3 (2003), 575–603.

[9] T. Hashimoto, Y. Shirota, A. Iizawa, and H. Kitagawa. 2001. Digest making method based on turning point analysis. In *Proceedings of the Second International Conference on Web Information Systems Engineering*, Vol. 1. 83–91 vol.1.

[10] M. B. Hisham, S. N. Yaakob, R. A. A. Raof, A. B. A. Nazren, and N. M. Wafi. 2015. Template Matching using Sum of Squared Difference and Normalized Cross Correlation. In *2015 IEEE Student Conference on Research and Development (SCOReD)*. 100–104.

[11] Matroid Inc. 2019. Matroid. https://www.matroid.com/. Accessed: February 2019.

[12] S. Johnsen and Ashley Tews. 2009. Real-Time Object Tracking and Classification Using a Static Camera. In *IEEE International Conference on Robotics and Automation - Workshop on People Detection and Tracking*.

[13] Munchurl Kim, J.G. Jeon, J.S. Kwak, M.H. Lee, and C. Ahn. 2001. Moving object segmentation in video sequences by user interaction and automatic object tracking. *Image and Vision Computing* 19, 5 (2001), 245 – 260.

[14] Rajshree Lande and R. M. Mulajkar. 2018. Moving Object Detection using Foreground Detection for Video Surveillance System. *International Research Journal of Engineering and Technology* 5, 6 (June 2018), 517–519.

[15] H. H. Le, T. Lertrusdachakul, T. Watanabe, and H. Yokota. 2008. Automatic Digest Generation by Extracting Important Scenes from the Content of Presentations. In *2008 19th International Workshop on Database and Expert Systems Applications*. 590–594.

[16] S. Lefevre, E. Bouton, T. Brouard, and N. Vincent. 2003. A new way to use hidden Markov models for object tracking in video sequences. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, Vol. 3. III–117.

[17] B. Li and M. Ibrahim Sezan. 2001. Event detection and summarization in sports video. In *Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL 2001)*. 132–138.

[18] Yi Liu and Yuan F. Zheng. 2005. Video object segmentation and tracking using ψ-learning classification. *Circuits and Systems for Video Technology, IEEE Transactions on* 15 (2005), 885 – 899.

[19] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (01 Nov 2004), 91–110.

[20] Wei-Lwun Lu and J. J. Little. 2006. Simultaneous Tracking and Action Recognition using the PCA-HOG Descriptor. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. 6–6.

[21] Zheng Lu and Kristen Grauman. 2013. Story-Driven Summarization for Egocentric Video. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*. IEEE Computer Society, Washington, DC, USA, 2714–2721.

[22] Abdullah M. Moussa, M I. Habib, and Rawya Rizk. 2015. FRoTeMa: Fast and Robust Template Matching. *International Journal of Advanced Computer Science and Applications* 6 (October 2015), 195–200.

[23] Joel Oberstone. 2009. Differentiating the Top English Premier League Football Clubs from the Rest of the Pack: Identifying the Keys to Success. *Journal of Quantitative Analysis in Sports* 5, 3 (2009), 1–29.

[24] Sander Oude Elberink and B Kemboi. 2014. User-assisted Object Detection by Segment Based Similarity Measures in Mobile Laser Scanner Data. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XL-3. 239–246.

[25] T. Oyama and D. Nakao. 2015. Automatic extraction of specific scene from sports video. In *2015 10th Asian Control Conference (ASCC)*. 1–4.

[26] Vyacheslav Parshin and Liming Chen. 2004. Video Summarization Based on User-defined Constraints and Preferences. In *Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval (RIAO '04)*. Paris, France, France, 18–24.

[27] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala. 2009. Image Registration by Template Matching Using Normalized Cross-Correlation. In *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. 819–822.

[28] Adel A. Sewisy, Khaled F. Hussain, and Amjad D. Suleiman. 2016. Speedup Video Segmentation via Dual Shot Boundary Detection (SDSBD). *International Research Journal of Engineering and Technology (IRJET)* 3, 12 (December 2016), 11–14.

[29] Sanjivani Shantaiya, Keshri Verma, and Kamal Mehta. 2013. Article: A Survey on Approaches of Object Detection. *International Journal of Computer Applications* 65, 18 (March 2013), 14–20.

[30] Colin Trainor. 2014. Goalkeepers: How repeatable are shot saving performances? http://www.statsbomb.com/2014/10/goalkeepers-how-repeatable-are-shot-saving-performances. Accessed: February, 2019.

[31] GKStopper. 2019. PROFESSIONAL GOALKEEPER SOFTWARE: The app that tracks keeper performance. http://gkstopper.com/. Accessed: February 2019.

[32] L Vibha, Chetana Hegde, P Shenoy, Venugopal K R, and Lalit Patnaik. 2008. Dynamic Object Detection, Tracking and Counting in Video Streams for Multimedia Mining. *IAENG International Journal of Computer Science* (01 2008).

[33] Q. Zhang and K. N. Ngan. 2011. Segmentation and Tracking Multiple Objects Under Occlusion From Multiview Video. *IEEE Transactions on Image Processing* 20, 11 (Nov 2011), 3308–3313.