ELEN E4830 Digital Image Processing Final Report

Players and Ball Tracking in Soccer Games

Qinglin Yang(qy2211), Tiantong Yu(ty2385)
Chenyao Liao (cl3757), Zhenduo Zhang (zz2551)

***Abstract*—The goal of this project is to process high-definition soccer footage (video clip from Youtube as the dataset for this project) in order to detect and track the locations of both the football and players, with the ultimate goal of automatically computing the number of passes which occur and the occupied rate of ball (the ratio of number of passes of one team to total number of passes for both teams) for each team. We are going to select some footage as our test footage which will include both stationary kicking and kicking that occurs when the intended receiver is in motion. Taking into consideration current advancements of digital image processing techniques, We are going to use MATLAB image processing methodology detailed below. Besides, we implement all the algorithms discussed in this project for detecting player and ball using OpenCV computer vision library, and the performance of all these algorithms in OpenCV platform is in general the same as that in MATLAB platform, which provides the ground truth for the validity of our algorithms. Finally, we conclude the whole project and present future work we are going to do.**

***Index Terms*—Morphological Processing, Color Segmentation, Background Subtraction, Template Matching, Trajectory, OpenCV**

I. INTRODUCTION

Digital Image Processing (DIP) is the process of digital images using various computer algorithms. This digital image processing has been employed in number of areas such as pattern recognition, remote sensing, image-sharpening, color and video processing and medical. In this project, we are going to use techniques such as morphological processing, color segmentation, background subtraction, template matching, and trajectory based ball detection. The literature review of these digital image processing techniques are briefly presented in this chapter.

Ever since the 1960s, some particular communities begun to use all sorts of schemes for nonlinear processing of images. Mathematical Morphology was developed 1964 by the collaborative work of Georges Matheron and Jean Serra. Morphology is one of several tools that form the foundation of techniques for extracting “meaning” from an image. Here, we are interested in morphological techniques for extracting the shape of moving soccer from the video.

In 1994, Koller et al. used background subtraction as an effective means of locating and tracking moving vehicles in freeway traffic. As time goes by, background subtraction is becoming a popular approach for detecting moving objects in video sequences. The basic idea consists in comparing each video frame with an adaptive background model (which can be reduced to a single image) free of moving objects. Pixels with a noticeable difference are assumed to belong to moving objects (they constitute the foreground) while others are classified as background.

Template matching is sometimes called as correlation-like methods or Area-based methods: Fonseca and Manjunath (1996) which is the combination of feature detection and feature matching. Template matching is a referential method which executes point-to-point comparison with an error-free reference data. It is widely used in the area of Optical Character Recognition, Computer Vision, Image Registration, Facial Recognition, etc. A numerous of template matching algorithms have been suggested over the years. In general, a template matching algorithm has three major parts: image space transformation, similarity measure and searching.

Besides the ball object detection, the interaction between the ball and players is also quite important. So detection of basic actions also is considered. An effective algorithm is proposed by Xinguo Yu, Changsheng Xu in 2003 which use trajectory - based ball detections which combined the objected tracking and trajectory and allow us to integrate trajectory algorithm into the image processing. By computing the velocity of ball, we could find the peaks and intervals which corresponding to the ball-player touching and passing. Also, since we separate the ball detection and player detection process, we could find out the ball passing belonged to which team by finding the corresponding location of the ball and players. More details will be discussed in methodology part.

II. Algorithms and Methodologies

A. *Color Segmentation*

1. Detection

Color segmentation can be used in the detection of the ball and the players. There are many types of color-based segmentation methods depending on different color spaces that we choose. We would like to use HSI color space because it contains direct information of the intensity that is important for color segmentation.

2. Motion Tracking

The method that we will use for tracking moving objects, in this case, the players, is mean shift tracking, which is one of the most generally used ways of color-based motion tracking. Other methods include camshift tracking, kernel-based object tracking, and ensemble tracking. Mean shift tracking is one of the first methods developed for motion tracking so that it has the most tested functionality. Another important reason we choose mean-shift tracking is that, the mean-shift algorithm is an efficient approach to tracking objects whose appearance is defined by histograms. It fits well for tracking player in this occasion.

Mean shift is a procedure for locating the maxima, or the mode, of the density function given the discrete data sampled from that function. First, we assign a random value x to a pixel. Then, we estimate the mean value based on a certain kernel function (typically Gaussian) which determines the weight of neighborhood pixels. The difference m(x) and x is called mean shift. We set the value of m(x) to x until m(x) converges.

Mean-shift algorithm is a nonparametric statistical method for seeking the nearest mode of a point sample distribution. Mean-shift has recently been developed into an efficient technique that can be used for appearance-based blob tracking. In this project, players tracking resembles blob tracking in a way that the players all wear vivid color shirts. In this tracking scenario, sample points are regularly distributed along the image pixel grid. A pixel's value m(x) is actually the sample weight of the point at that location. This sample weight is chosen such that pixels on the foreground blob have high weight, and pixels in the background have low weight. In another way, the probability of a color u in the target model. The mean-shift algorithm specifies how to combine this chosen sample weights m(x) in a local neighborhood with a set of kernel weights K(x) to produce an offset that tracks the central points of the blob in the image.

B. *Morphological Processing*

The local transformations in Morphology is defined as changing pixel values that are represented as sets. In the hit and miss transformation, an object represented by a set X is examined through a structural element represented by a set B. Different structuring elements are used to change the operations on the set X. The hit or miss transformation is actually a point operator.

$$X \otimes B = \left\{ x \mid B_x^1 \subset X \cap B_x^2 \subset X^c \right\}$$

In this equation, x represents one element of X, which is a pixel in an image. The symbol X^c denotes the complement of X and the structuring element B is represented by two parts, B^1 and B^2, that are applied to the set X or to its complement X^c . The structuring element is a shape and this is how mathematical morphology operations process images according to shape properties. The operation of B^1 on X is a "hit"; the operation of B^2 on X^c is a "miss." The subindex x in the structural element indicates that it is moved to the position of the element x. That is, in a manner similar to other group operators, B defines a window that is moved through the image.

The simplest form of morphological operators is defined when either B^1 or B^2 is empty. When B^1 is empty, the equation above defines an erosion, and when B^2 is empty, it defines a dilation. That is, an erosion operation is given by :

$$X \ominus B = \left\{ x \mid B_x^1 \subset X \right\}$$

and a dilation is given by

$$X \oplus B = \left\{ x \mid B_x^2 \subset X^c \right\}$$

In the erosion operator, the hit or miss transformation establishes that a pixel x belongs to the eroded set if each point of the element B^1 translated to x is on X. Since all the points in B^1 need to be in X, this operator removes the pixels at the borders of objects in the set X. Thus, it actually erodes or shrinks the set. One of the most common applications of this is to remove noise in thresholded images.

The dilation operator establishes that a point belongs to the dilated set when all the points in B^2 are in the complement. This operator erodes or shrinks the complement and when the complement is eroded, the set X is dilated.

C. Background Subtraction

By using the Background subtraction algorithm, depends upon the every frame features, background formation and foreground computations are calculated.

It has two steps in this algorithm, as shown in Fig.1. Fist, Initialization of background, i.e., the generation of first model of background. Second, Updating the model, i.e., depends upon the scene changes, the model information is updated. Fig.2 shows that the steps involved in the background subtraction algorithm.

```
Algorithm:
Initialization:
Calculate the frame difference at time t
 Thr –Threshold value
S1:Ĭt(x,y)=Intensity of pixel with coordinate (x,y) in the frame at time t.
Ĭt-1(x,y)=Intensity of pixel(x,y) in the frame at time t-1.
    While i =0:n;
      Ŧ(x,y)=| Ĭt(x,y)- Ĭt-1(x,y)|;
      If(Ŧ(x,y))= =o;
     Then Foreground binary mask= Ŧ(x,y);
      If (Ŧ(x,y)>Thr)
               Ŧ(x,y)=foreground pixel;
      Else
               Ŧ(x,y)=background pixel;
```

Fig.1 Background Subtraction Algorithm

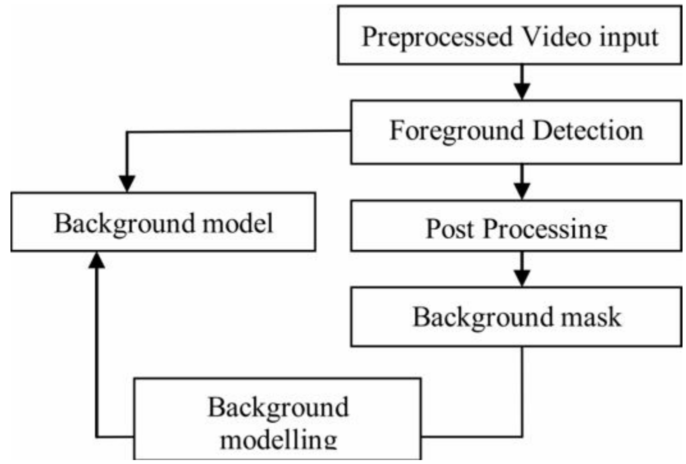


Fig.2 Procedure of Background Subtraction

D. Template Matching

The method of template matching is given as an algorithm, which is simple and easy to implement. The algorithm steps are as follows:

Step 1: The basic template matching algorithm consists of calculating at each position of the image under examination a distortion function that measures the degree of similarity between the template and image.

Step 2: The minimum distortion or maximum correlation position is then taken to locate the template into the examined image. In other words, template matching involves comparing a given template with windows of the same size in an image and identifying the window that is most similar to the template.

Step 3: During the detection and tracking of the soccer it must be noted that the soccer's attributes change a lot over the frames and using a single template to match all angles of the soccer may not be feasible.

E. Trajectory based ball detection

Traditional object tracking algorithm we discussed above is hard to locate the simple actions between the ball and players. Thus we reference a trajectory algorithms to compute the velocity of the balls in order to find the ball passing and kicking.

First step is to find the ball's candidate in each frame which has been completed before, and the second step is using trajectory information of ball candidate objects over a short sequence of frames to identify the ball trajectory.

For each frame *F*, we identify the set of objects in the frame, denote by *O(F)*, then we develop a set of sieves based on properties like size, color, shape, etc. The remaining objects are the ball-candidates *C(F)* and they satisfy all the properties we defined of ball. By setting up the threshold, we may get small set of objects, which may be ball. Since the ball should be the most active object in the soccer video, we should use trajectory information to find the ball.
The solid algorithm is shown:

Let CFI be each combination of ball candidates' numerical features in a sequence of frames.
Let Γ be the set of all candidate trajectories in a CFI
SET the ball trajectory set B to be empty
WHILE Γ is not empty DO
Move the trajectory T into B if it has the highest index in T and its index > given threshold
Remove the candidate trajectories that overlap with T in Γ.

Since we could compute the location of the ball in this way, it is easy to get the corresponding velocity. Every touch by player to the ball will alter the direction and velocity of the ball. Such motion change the ball will be called pivot points.

Plot the corresponding pivot-to-time graph. It should be easy to find the peak, which correspond to the touching point, and the interval of the peak should be the passing of the ball.

III. Datasets

We intend to use downloaded videos of soccer games from Youtube. The video we intend to use is taken from a game of 2018 FIFA Worldcup that is 20 second long 1080p mp4 video with about 20 frame/second refresh rate. The total number of images to be processed should be no smaller than 500. The type of images that we will use is JPG which is determined by the type that Matlab is able to extract from videos with size of 720x1280. We would like to process the images in the colorspace of HSI. When dealing with grayscale images in the process, 8 bit gray level is our choice.

IV. Results

A. *Test Videos Ball and Players Tracking*

1.1 Ball Tracking

1.1.1 Approach 1: Morphological Tracking

1) Color segmentation
In this case testing the test video with two players and an orange soccer ball, we chose color segmentation for ball tracking. Since the ball is orange, we use the red channel in colorspace for segmenting the ball. However, the bright areas in the players' shirt becomes a problem because they also have high levels in the red channel. Therefore, we filtered these bright parts in the blue channel (because the grass is green) by thresholding before segmenting the ball in the red channel. The filtered blue channel image, which is binary, resembles the function of a mask. Then the product of the mask and the red channel image gives us an image with filtered players, leaving only the soccer ball.

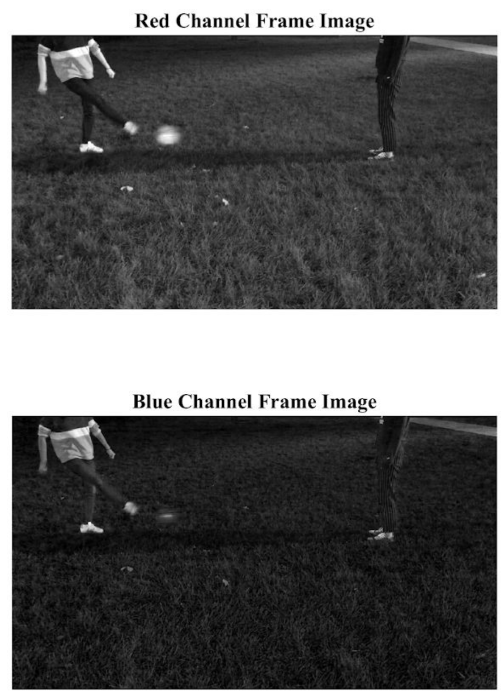


Fig.3 Preprocessing Using Color Segmentation

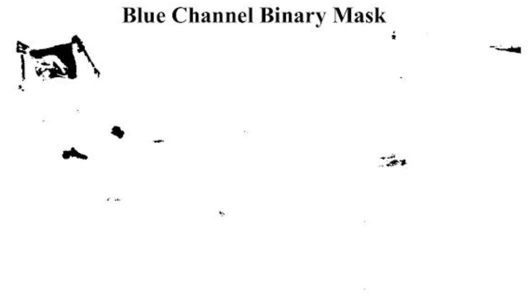


Fig.4 Final Frame Image After Pre-Processing

2) Morphological Processing
Then, after getting a binary image thresholding Fig.5, we let the image go through a series of morphological processes. First opening followed by closing to eliminate the players' arms and other unwanted areas. Then we use connected region process to find the connected region with largest area which is the ball area. Then, we apply region descriptor to find the diameter and center point of the area and use these parameters to draw a circle which is the edge of the ball.

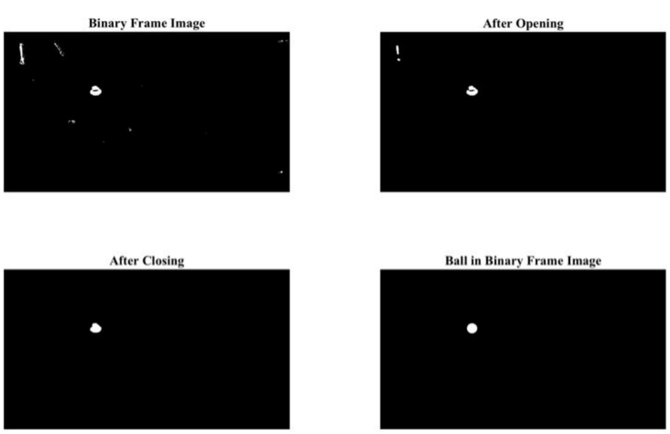


Fig.5 Morphological Processing

Fig.6 Final Ball Detection Result

1.1.2 Approach 2: Mean-shift Tracking

1) Motion deblurring
The ball of certain frame in test video might distort as an eclipse because of moving too fast, which would increase difficulty for tracking.

Fig.7 Motion Deblurring Using Wiener Filter

Here, we used three Motion Deblurring methods on these frames for converting the ball back into shape of sphere as well as deblurring . As shown below, deblurring images applying the Lucy-Richardson Algorithm worked best. In the method adopting

Wiener filter, shades were still remained around the ball. Motion Deblurring results by Wiener Filter, Blind Deconvolution and Lucy-Richardson Algorithm are displayed below.

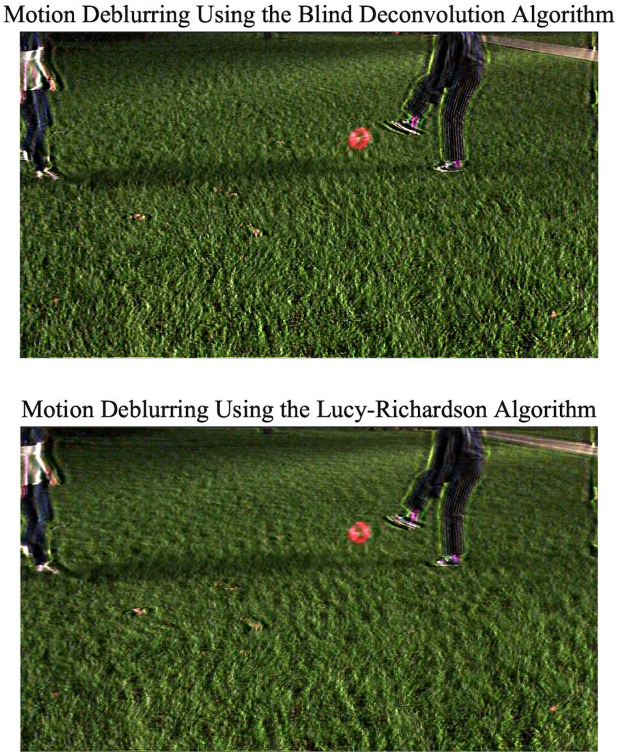


Fig.8 Motion Deblurring Using Blind Deconvolution and Lucy-Richardson Algorithm

2) Mean-shift Ball and Tracking
Decide the weight of pixels in the interested area. Use Gaussian kernel to decide the weight of each pixel based on its distance to the center point. Then apply mean shift to visual tracking by creating a confidence map in the new frame based on the weighted color histogram of the object in the previous frame.

Fig.9 Mean-shift Tracking Result

Then by using mean shift we can find the peak of a confidence map near the object's old position. The confidence map is a probability density function on the new image, assigning each pixel of the new image a probability, which is the probability of the pixel color occurring in the object in the previous image.

1.2 Players Tracking

1.2.1 Color Segmentation
As for the selection of colorspace we would use for player segmentation, we have tested for HSV Colorspace, XYZ Colorspace, Lab Colorspace and YCbCr Colorspace, as listed below. According to the results, Lab Colorspace worked worst for separating players from background. While in HSV Colorspace, the clothes and shoes of both players as well as part of the ball could be easily extracted, and hence was used for further processing.

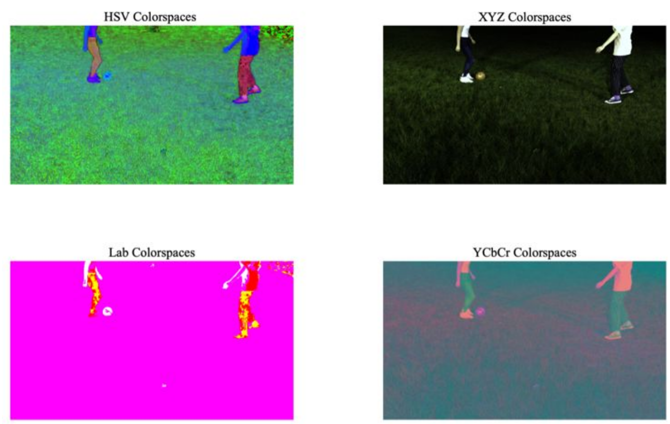


Fig.10 Frame in Different Colorspaces

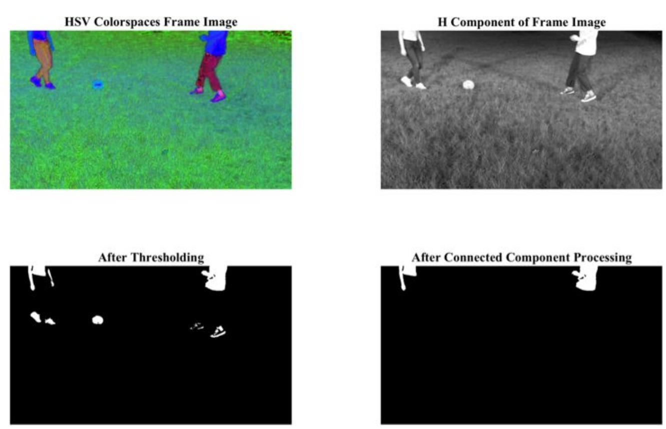


Fig.11 Color Segmentation in HSV Colorspace

1.2.2 Morphological Processing

After we decide the final colorspace that we would like to use, we choose an exact channel that we can operate most efficiently in, which is the H Component Channel and carry out thresholding to get a binary image. Use connected component processing to find the two largest areas and apply region labeling to track them separately.

Result in Binary Frame Image

Final Result for Player Tracking

Fig.12 Final Players Detection Result

1.3 Speed Tracking and Kick Counting

Using the results that we obtain from tracking, the practical application of the algorithm can be 1) calculate the real-time speed of the soccer ball, and 2) counting the kicks of each play in a game for judging the player's performance.

1.4 Algorithm Comparisons

Based on the methods described above, we could do comparison on mean shift and morphological processing. We compare several aspects based on the performance of this 2 algorithms test on the self-recorded test video. More details will be shown in the chart:

	Mean Shift	Morphological
Speed (on same PC testing same video)	28.310	28.751
Accuracy	Stable	Higher when no large areas with similar color are in the frame
Immunity to Blurring	Low immunity	High immunity
Generalization	Not well generalized, parameters should be adjusted when applied to different videos	Not well generalized, depends a lot on the color of the T-shirts and ball
Advantages	Possible improvements with more complex algorithms	Less complicated algorithm
Disadvantages	It is not scalable. Accumulated error from previous frame. Dependence on first frame and interest area choosing. Not 100% accuracy.	•Vulnerable to environment changes

B. Real FIFA World Cup Video

2.1 Player Tracking

2.1.1 Morphological Color-based Segmentation

As we described in section 1.1, color segmentation could also be used in distinguishing players from different teams. This method will be really efficient to teams whose colors are in high contrast. In our project, we use a classic 2018 FIFA World Cup video cut, Japan VS Belgium, as our test video.

By choosing the most highlight region in red and blue dimension, and do dilation afterwards, we are able to distinguish the players from 2 teams.

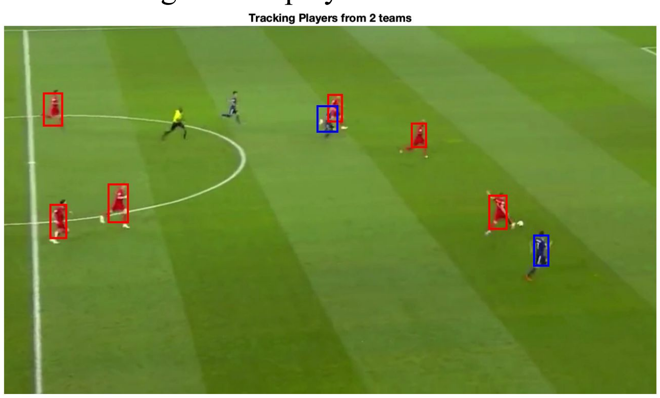
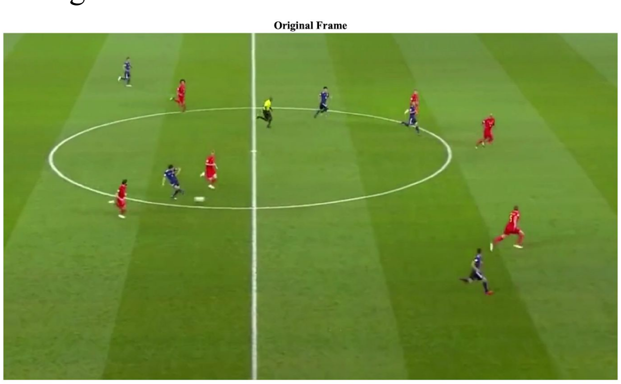


Fig.13 Tracking Performance of Color-based Segmentation

As shown in Fig.13 Players from different teams have been labelled in rectangle with corresponding color.

2.1.2 K-means Based Segmentation

Despite the Color-based Segmentation doing well in high contrast team color, we need to consider the competition that 2 teams may have similar colors and hard to be divided in colorspace. To solve this problem, we consider K-means Clustering to distinguish colors.



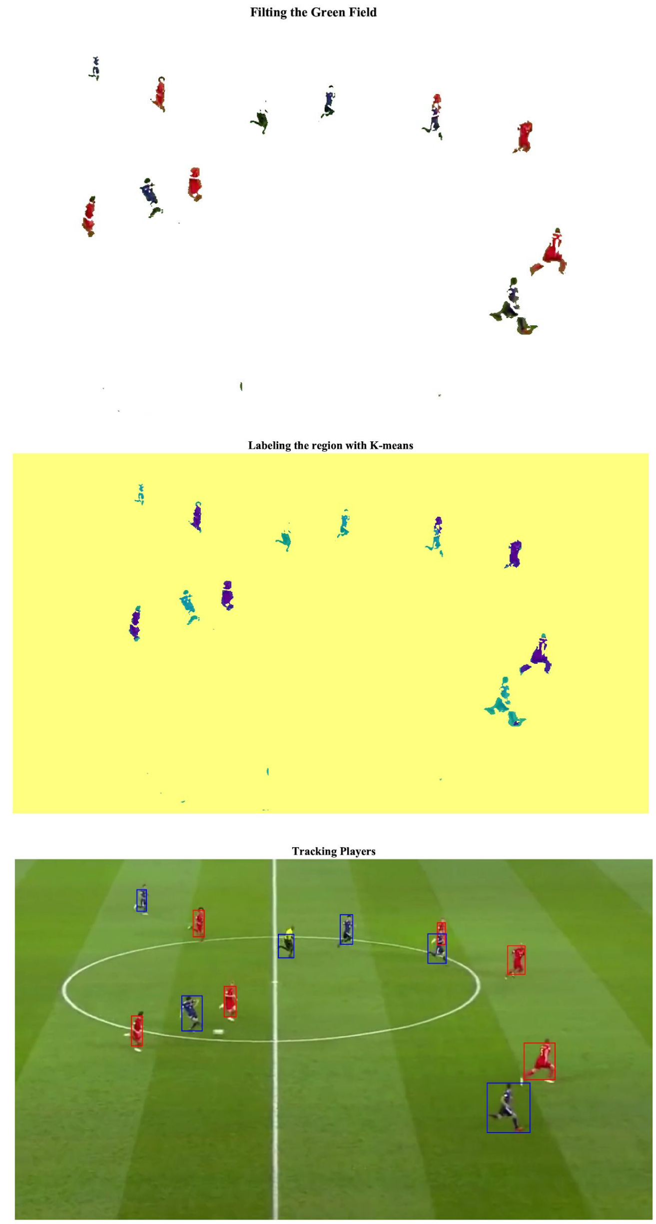


Fig.14 Background Subtraction, Region Labeling and Final Player Tracking Results

As shown in Fig.14, we first use rgb color levels to subtract green field, set them to white will decrease the noise from the field line. Then we use K-means Clustering to do region labelling, players from different team will be assigned to their most similar color region, labelled as 1,2 and 3, correspond to background, red team and blue team. Based on every frame the label varies, but based on the centers, we could tell the difference of 3 labelled region. We could see the labelled region in 3rd photo of Fig.14, the yellow part is the background, purple region is read team and green region belong to blue team.

The details of K-means Algorithm are shown below:

Given: data $\vec{x}_1, \vec{x}_2, \dots \vec{x}_n \in \mathbf{R}^d$, and intended number of groupings *k*
Alternating optimization algorithm:
- Initialize cluster centers (say randomly)
- Repeat till no more changes occur
  1. Assign data to its closest center $\vec{c}_1, \vec{c}_2, \dots \vec{c}_k$ (create a partition)
  2. Find the optimal centers (assuming the data partition is fixed.

The result of K-means algorithms are shown in last image of Fig.14.

2.1.3 Comparison of morphological and K-means Clustering

	Morphological	K-means Clustering
Speed (for 1 frame)	1.794 s	4.193 s
Accuracy	stable	Depend on the color noise.
Generalization	Low	Higher, the algorithm automatically assign players into 2 clusters

2.2 Ball Tracking

2.2.1 Template Matching
So far we have use k-means and morphological to distinguish players. Then we use template matching to locate the soccer in the game. Since the snickers of players have similar color with ball, color-based segmentation may be inefficient in this situation.

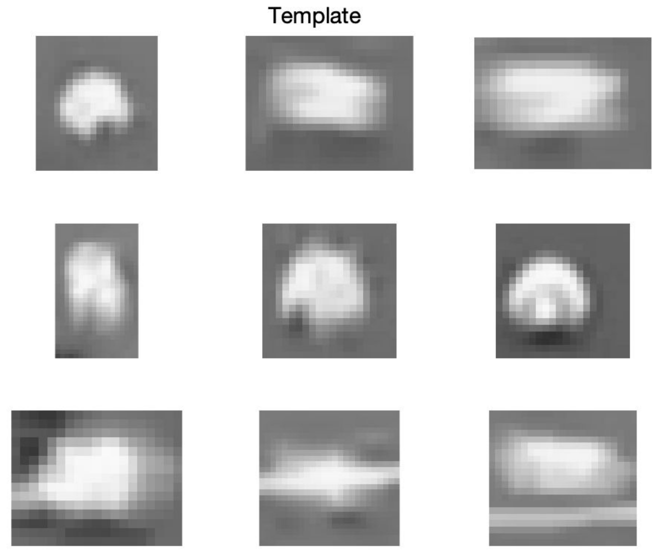


Fig.15 Templates for Template Matching

As shown in Fig.15, Based on the template matching process, I have choose 9 templates for test video.

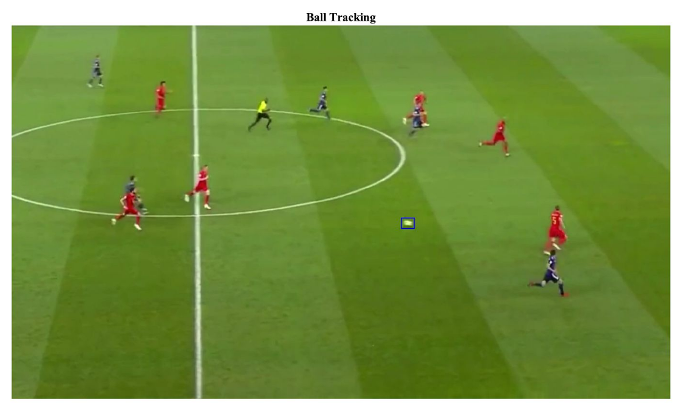


Fig.16 Results of Ball Tracking

We do convolution to find the correlation between the template and target frame. Find the maximum correlation coefficient and corresponding target ball location. We choose threshold as 0.8, which means we only use the candidates with coefficient over 80%. Also, we compare every 9 template candidate and find the most possible candidate, assign it as the ball location we choose.

As shown in Fig.16, the ball is viewed in a blue bounding box.

2.2.2 Speed calculation and Accuracy Computation

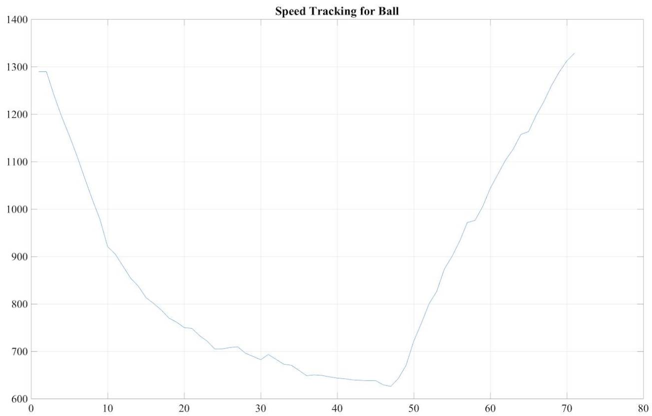


Fig.17 Speed of ball in self-recorded test video

The speed of the ball could be computed, we could see the peaks in the Fig.18 and Fig.17, which correspond to the kicking actions in the video. Every time a player kick or the ball bounced on the field, the speed will appear a peak. Also, if the speed gradually slower with the time, it should correspond to a pass or a shooting.

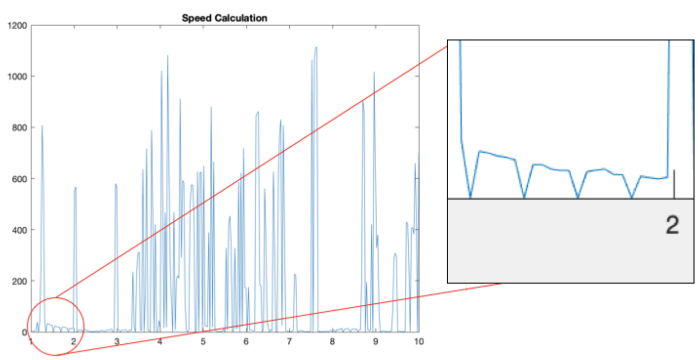


Fig.18 Speed of ball in real FIFA test video

As shown in Fig.18, the ball speed shows some 0 values, which corresponding to the detection loss of the template matching. It is hard to calculate the accuracy by eyes, thus we use it for evaluation. Every unnormal valley could be counted as a loss. As for 313 frames, we have 47 zero values, which is 84.98% accuracy.

2.3 OpenCV realization:

2.3.1 Player tracking

In order to demonstrate the validity of algorithms implemented in this project, we use the same algorithms to process the soccer game clip to better explain what we've done in this project is actually effective through implementing these algorithms in other platforms like OpenCV computer vision library, which was originally developed by Intel but subsequently released under the BSD license. The original image is one of frames extracted from the soccer game clip shown in Fig.19 and the tracking performance using OpenCV computer library is shown in Fig.20.

Fig.19 The frame extracted from the soccer game clip

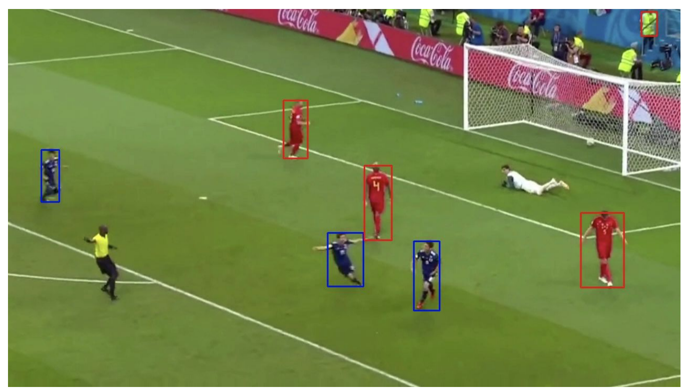


Fig.20 Tracking Performance after using Color-based Segmentation

We first read the soccer game frame by frame and then convert all the frames into HSV format based images. By using HSV image can we detect any specified color background ranging from specific codes. In this project, we first need to detect the green field using HSV technique, and the other parts will in general become black. After that, it is much easier to detect players by jersey colors.

As shown in Fig.21, we define a mask of a given green color range to detect the field. The players and audiences are shown in black, the green field is preserved within our given green color range.

Fig.21 Detecting the field after using a mask within the green color range

After applying morphological closing operations to these frames, the noise present in the crowd will be filtered out. Thus the accuracy of detection can be guaranteed.

Fig.22 The frame after doing morphological operations

Now we just need to find all contours on every frame, shown in Fig.22. We consider contours whose height is greater than width as detected player. For recognition of jersey color, we can do the same. Namely, we do masking operation on the detected players to distinguish different jersey colors. For simplicity, we just put a rectangle with blue or red colors representing blue or red teams to each person.

2.3.2 Ball detection

For detecting soccer ball, we detect white color using template matching with a suitable mask in this case. The performance for this soccer detection, shown in Fig.23, remains to be improved in the future as listed in the future work.

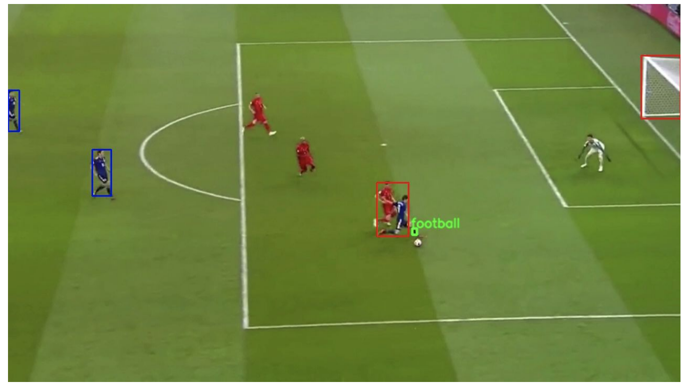


Fig.23 Ball detection using template matching

V. Conclusion

We implemented algorithms such as color based morphological operation, mean shift tracking, kmeans tracking, template matching to track players and ball in both the real soccer game and our test games. After using these algorithms, the overall performance for tracking player and ball is good. Some conclusions can be drew below:

1.We did algorithm comparison between Morphological and Mean-Shift. Mean-Shift algorithm has comparable higher immunity to blurring compared to Morphological. Also, we compared algorithm comparison for Morphological and K-means Clustering algorithms, and Morphological algorithm owns the higher speed and stability, while K-means owns better generalization characteristic.

2. Speed calculation and accuracy computation for the ball, and the final result shows achieved 84.98% accuracy.

3. Moreover, we implement all the algorithms discussed in this project for detecting player and ball using OpenCV, and the performance of all these algorithms in OpenCV platform is in general the same as that in MATLAB platform, which provides the ground truth for the validity of our algorithms.

VI. Future Work

A. Mean-Shift Ball Tracking

We could make an attempt to upgrade to a scalable mean shift tracking. One of the biggest drawbacks that affects the accuracy of detection and tracking is the motion of the ball moving closer or further to the camera. Therefore, the interested area should change in different frames instead of remaining the same. The generalization of the algorithm could also be further improved by this upgrade.

B. Test Videos Player Tracking

Other objects in the field may also influence the accuracy of detection and tracking which we should consider. It can be even further combined with face detection in the future with higher resolution videos.

C. Real FIFA Video Tracking

Use K-means to track the field line, goal and judge. Increase the accuracy of players detection, solve the miss condition due to the distance between players and camera.

D. Trajectory

Use non-parameter regression to find the smoothing line of ball's speed, remove the noise because of the incorrect detection. Locate the touching point, calculate kicking, passing, running time of different teams. Integrate the trajectory selection, find out a passing is successful or not, calculate ball-holding time.

VII. Credits

1 *Color-based morphological* test video ball and player tracking: Chenyao Liao, Tiantong Yu
2 *Mean shift* test video tracking: Tiantong Yu
3 *Motion deblurring* algorithm: Chenyao Liao
4 *Color-based morphological* game video player tracking: Qinglin Yang
5 *K-means* game video player tracking: Qinglin Yang
6 *Template matching* game video ball tracking: Qinglin Yang
7 *OpenCV realization*: Zhenduo Zhang

References

[1] Liang, Dawei Liu, Yang Huang, Qingming Gao, Wen. (2005). A Scheme for Ball Detection and Tracking in Broadcast Soccer Video.

[2] *T. DOrazio, M. Leo, A review of vision-based systems for soccer video analysis Pattern Recognition (2010),doi:10.1016/j.patcog.2010.03.009*

[3] *Mark S. Nixon, Alberto S. Aguado.(2012).Feature extraction & image processing for computer vision.*

[4] *S. Haifeng and X. Chao, "Moving Object Detection Based on Background Subtraction of Block Updates," 2013 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Shenyang, 2013, pp. 51-54.*

[5] *S. V. Mashak, B. Hosseini, M. Mokji and S. A. R. Abu-Bakar, "Background subtraction for object detection under varying environments," 2010*

[6] *International Conference of Soft Computing and Pattern Recognition, Paris,2010,pp.123-126.doi:10.1109/SOCPAR.2010.5685960*

[7] *Chantara, Wisarut Mun, Ji-Hun Shin, Dong-Won Ho, Yo- Sung. (2015). Object Tracking using Adaptive Template Matching. IEIE Transactions on Smart Processing and Computing. 4. 1-9. 10.5573 / IEIESPC.2015.4.1.001.Yu, Xinguo Xu, Changsheng Leong, Hon Tian, Qi Tang, Qing Wan, Kongwah.(2003).Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. ACM Multimedia. 3. 11-20. 10.1145/957013.957018.*

[8] *Yu, Xinguo Xu, Changsheng Leong, Hon Tian, Qi Tang, Qing Wan, Kongwah. (2003). Trajectory based ball detection and tracking with applications to semantic analysis of broadcast soccer video. ACM Multimedia. 3. 11-20. 10.1145/957013.957018.*

[9] *https://www.codeproject.com/Articles/336915/Connected-Component-Labeling-Algorithm*

[10] *Fukunaga, Keinosuke; Larry D. Hostetler (January 1975). "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition". IEEE Transactions on Information Theory. 21 (1): 32–40. doi:10.1109/TIT.1975.1055330.*

[11] *R. T. Collins. Mean-shift Blob Tracking through Scale Space. InCVPR, 200*