LONGSHOT FOOTBALL DETECTION AND TRACKING USING MEAN SHIFT ALGORITHM

A Thesis Submitted In Partial Fulfillment Of The Requirements For The Degree Of

Master of Technology

in

Signal & Image Processing

by

Chitturi Vinod Kumar

213EC6263

Department of Electronics and Communication in Engineering

National Institute of Technology Rourkela

Odisha,India-769008

May,2015

LONGSHOT FOOTBALL DETECTION AND TRACKING USING MEAN SHIFT ALGORITHM

A Thesis Submitted In Partial Fulfillment Of The Requirements For The Degree Of

Master of Technology

in

Signal & Image Processing

by

Chitturi Vinod Kumar

213EC6263

Under the super vision of

Prof. Sukadev Meher

Department of Electronics and Communication in Engineering

National Institute of Technology Rourkela

Odisha,India-769008

May,2015

Dedicated to My Family...

SIGNAL AND IMAGE PROCESSING
ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ODISHA, INDIA-769008

CERTIFICATE

This is to certify that the Thesis Report titled, **"LONG SHOT FOOTBALL DETECTION AND TRACKING USING MEANSHIFT ALGORITHM"**, submitted by **Mr. VINOD KUMAR CHITTURI** bearing**Roll No. 213EC6263** in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Electronics and Communication Engineering** with specialization in**"Signal & Image Processing"** during session 2014 - 2015 at National Institute of Technology Rourkela is an authentic work carried out by him under my supervision and guidance.

Place: Rourkela

Date: 01Jun2015 **Prof. Sukadev Meher**

SIGNAL AND IMAGE PROCESSING
ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ODISHA, INDIA-769008

DECLARATION

I certify that

1. The work contained in the thesis is original and has been done by myself under the supervision of my supervisor.

2. The work has not been submitted to any other Institute for any degree or diploma.

3. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Chitturi Vinod Kumar

Acknowledgment

I would like to express my gratitude to my thesis guide **Prof. Sukadev Meher** for his guidance, advice and support throughout my thesis work. I am especially indebted to him for teaching me both research and writing skills, which have been proven beneficial for my current research and future career. Without his endless efforts, knowledge, patience, and answers to my numerous questions, this research would have never been possible. The experimental methods and results presented in this thesis have been influenced by him in one way or the other. It has been a great honour and pleasure for me to do research under supervision of Prof. Sukadev Meher. Working with him has been a great experience. I would like to thank him for being my advisor here at National Institute of Technology, Rourkela.

Next, I want to express my respects to **Prof. K.K Mahapatra, Prof. S. K. Patra, Prof. Samit Ari, Prof .A.K. Swain, Prof. D.P. Acharya, Prof. U. K. Sahoo, Prof. A. K. Sahoo, Prof. L.P.Roy, Prof. Manish Okade** for teaching me and also helping me how to learn. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank to all my faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela, for their generous help for the completion of this thesis.

I would like to thank all my friends and especially my classmates for thoughtful and mind stimulating discussions we had, which prompted to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

I am especially indebted to my parents for their love, sacrifice, and support. My parents are my first teachers, after I came to this world and I have set of great examples for me about how to live, study and work. I am grateful to them for guiding my steps on the path of achievements since my infant hood.

Chitturi Vinod Kumar

Contents

LONGSHOT FOOTBALL DETECTION AND TRACKING USING MEAN SHIFT

ABSTRACT

The soccer video analysis has a lot of importance these days commercially. There are many challenges in making the soccer video automatically analyzed. The automatic analysis is highly useful in analyzing player performance, computer assisted referee and tactics inference. In this thesis, we have proposed distinct methods to capture the ball movements thereby finding which team has the control on the ball for most of the time. Distinct methods mean different methods for detection, tracking and to find the control.

In the detection to identify the location of the ball, we have proposed a 9 step method. We have to execute these steps one by one. These steps include play field detection, edge detection, moving object detection, conjunction, morphology, shape strainer, size sieve, dominant color extraction, neighborhood detection. These steps are mainly based on attributes of the ball. These attributes include color, shape, and size. The moving object detection step work based on the optical flow of the ball.

In the tracking phase, we will select a region of interest in which we will find the most probable location of the ball by maximizing the likelihood of multiplication of color histogram of location and ball. After finding the most probable location, we localize the ball in the region by mean shift tracking method. The mean shift tracking algorithm is the one which works base on weights derived from Bhattacharyya coefficient to delineate the ball or target. Bhattacharyya coefficient is naturally used to find the similarity between two statistical samples by considering their probability distributions.

In the control phase, we divide the frame into different clusters using k-means algorithm. In k-means, we start the iterative process by selecting random starting points. The image pixels which has color values close to the different centers will be clustered into different regions. We will find the centroids or centers of these different regions and run the process again. We have applied this algorithm to our soccer frames got the results.

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction: Ball Tracking

The automatic analysis of soccer ball in video sequences has been an interesting application in computer vision. Because of its commercial importance sports video analysis got A lot of interest from researchers. It is a difficult challenge to analyze the huge Volume of available sports data. The important areas of sports video analysis revolve around the following events[3]:

- Object tracking,
- Tactics inference,
- Shot classification,
- Video indexing and summarization,

– Ccontent insertion,

– Computer assisted referring.

1.1.1 Object Tracking

Object tracking is an important issue in the field of sports. Object in the sense players and ball. Because of its usefulness, for further understanding of the video, detection and Tracking the path of the ball is has got the lot of importance.It is also useful in

– Player team identification,

– Player Position Localization in the actual playfield

Liu, Sophie Xiaofan et ali in his ”Video-based soccer ball tracking” proposed a combination of Background subtraction and morphological operation to detect players and Ball, Kalman filter to track the ball.Chakraborty Bodhisattwa proposed a trajectory-based ball detection and tracking algorithm with a robust background subtraction model.

1.1.2 Tactics Inference

The tactics analysis mainly used by the coaches and players. By using this, they can evaluate the things like

– Teamwork in a soccer game by using the movement of the players in the game,

– Performance evaluation and enhancement in sport.Coaches and analysts currently spend a lot of time in front of a PC screen going through films and analyzing the individual players play during football matches. Before the players get to see the analyses for themselves, it takes a long time. This process can be fully automated

Figure 1.1: A camera looks down over the Blundell Park pitch from the gantry

by using our tools. This automation allows us to save time and reduce the amount of data involved.

1.1.3 Shot Classification

Shot classification is one of the primary issues that are present in the sports video analysis. By using this, we can analyze things like

- Timing of the shot,
- What shot player is playing in the situation.

Chakraborty Bodhisattwa proposed to find ”serve” by position of the player, ”bump” by path of the ball, ”setter” by position of the player near the net,”spike” by position of the ball in a basketball sequence.

1.1.4 Video Indexing And Summarization

Video indexing and summarization is useful in extracting important events in the game by marking the video at different spots. These important events referred as Highlights and this marking is called as Indexing. Use of this indexing includes

- Highlights, shorter representation of the game to the viewers,
- Managers can send part of the sports video or practice session video to the players and they can discuss it.

1.1.5 Content Insertion

Content insertion is a purely commercial purpose. It is a recognized known fact that sports have significant importance in the advertising area. It is necessary to modify the sports video to inserting and changing the ad holdings according to the local area. The complexity of this task lies in inserting the additional information at the correct place in right time without disturbing the original content of the sports video.

1.1.6 Computer Assisted Referring

Sometimes it is difficult to take some decisions in the field by the referee on the spot. This criticality in making the decisions can be made simple by this computer assisted referring which is fully automated. For example, Hawk-eye system highly useful in tennis and cricket. In the below figure 1.2 we have shown the system that uses seven high-speed cameras per goal, it will the ball in the air through triangulation.

Figure 1.2: Hawk-Eye by 7 High Speed Cameras

1.2 Basic Object Tracking

Object tracking is one of the topics that gained the interest of most of the researchers in the area of computer vision. Because of high power systems, inexpensive video cameras and the need for automation in video analysis increased the interest in object tracking algorithms. The use of object tracking is important in the tasks of[20]:

- Motion-based recognition, that is, gait based human identification, automatic object detection, etc.;
- Automated surveillance for increasing safety in several applications such as national security, home safety, bank safety, etc.;
- Video indexing to get a set of video documents sharing similar intentions;
- Human-computer interaction, that is, gesture recognition;
- Traffic monitoring to control traffic in high traffic cities;
- Computer assisted game study for automatic analysis of series of significant events, happen in the game, such as the interaction between players and ball in football, etc.;

– Vehicle navigation based on video planning the path.

Object Tracking can be defined as estimating the path of the object in an image plane as it revolves around the scene.

1.2.1 Complexities in Tracking

one can found the tracking is a challenging thing if one or combination of any of the above situations is present in the video.

– When converting from the 3D world to 2D world we may lose information.
– The noise present in the videos,
– The complex motion of the object, that is, it may be translational, rotational, affine or combination of them.
– Nonrigid objects.
– Full or partial occlusion.
– Different typical shapes of the object.
– Change in the illumination of the scene
– Real time processing

Especially, the motion of the object is a much more complicated thing. But almost all of the proposed algorithms assumes that the velocity of the object changes linearly with time or constant with time.

1.3 Object Tracking Stages

In the figure 1.3 we have shown the different steps to track the object in a video sequence.

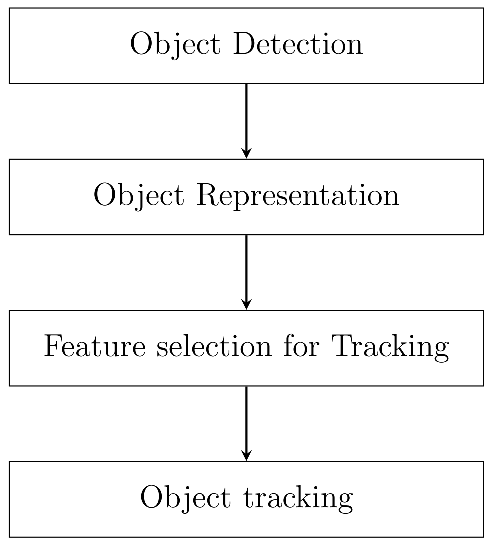


Figure 1.3: Object Tracking Stages

These stages are explained in the following sections.

1.3.1 Object Detection

Every method of tracking requires detecting the object either in the first frame or some of the frames of video. Some methods use information available in the single frame to identify the object where some of them uses the temporal information to detect the object. Point detectors, Segmentation, Background modeling, Supervised Classifiers are the examples of object detection mechanisms.

– In **Point Detection**, we will find the interesting points in the frame that have an expressive texture that is like Intersection point between two or more edge segments. Harris corner detector, SIFT detectors, etc.. are examples of this type.

– In **Background Modeling**, we build a background model of the scene, and we find the deviations from the each frame from the model. The mixture of Gaussians, Codebook construction, etc.. comes under this type.

– In **Segmentation**, we divide the image into several regions that have perceptual similarity. Mean Shift clustering, K-means clustering, etc.. comes under this type.

– In **Supervised Classifiers**, object detection can be performed by means of the supervised learning mechanism. Support Vector Machines, Neural Networks, etc.. comes under this type.

1.3.2 Object Representation

After detecting the object, we need to represent it with a shape that fits the object. Points, Geometric shapes, Skeletons, Object contour, silhouette, Articulated Shape models are commonly used object shapes for tracking. For objects that occupy a smaller region, we used to represent a single point that is the centroid of it. To represent the simple rigid objects we use primitive geometric shapes like rectangle, parabola. For tracking involved non-rigid structure, we use object contour and silhouette. Skeleton models can be used to model both articulated and rigid objects. In figure 1.4, we have shown different types of representations explained above in this paragraph.

1.3.3 Feature Selection For Tracking

Feature selection has a crucial role in the object tracking. In the feature space any object can be distinguished based on its visual feature or its uniqueness.

– **Color** is used as feature in the histogram based tracking methods. The color of an object can be influenced primarily by two factors 1) illumination, 2) Reflection Properties of the object. Naturally RGB color model is used to represent the color of the object. But in RGB color components are highly correlated, so it is not

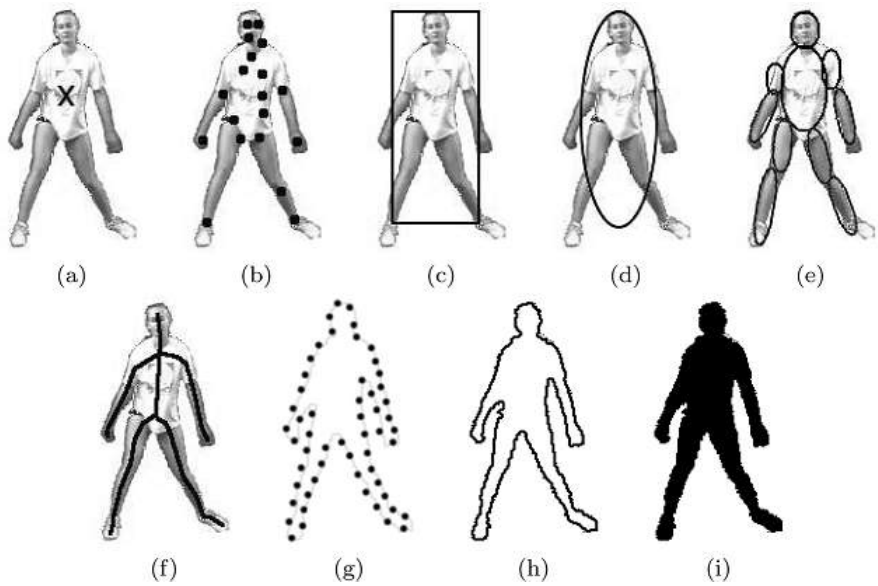


Figure 1.4: Object representations

(a) Centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) part-based multiple patches, (f) object skeleton, (g) complete object contour, (h) control points on object contour, (i) object silhouette

recommended to work with this color model. Instead of using RGB we go for L*a*b* or L*u*v* color spaces that are perceptually uniform color spaces and color components are not correlated. However, these color models can be affected by noise quickly, that is why cannot say which color space is right.

– **Edges** are used as the feature for tracking in algorithms that track the boundary of the object. A real thing with the edges is that these are less sensitive to change in illumination.

– The **Optical flow** used as feature in motion-based tracking and segmentation applications.

– The **Texture** is a measure of smoothness and regularity in an image. It is also less sensitive to noise.

1.3.4 Object Tracking

The aim of the object tracking is to finding the location of the object in every frame of the video there by finding its path. The task of identifying the object and building the correspondence between the object occurrences across frames can either be performed independently or jointly. In the first case, we go for identifying the object in each and every frame. In the second case, object location is found by updating the information available from the previous frames. Some of the categories of tracking explained below:

- **Point tracking.** Points are used to represent the object detected in the consecutive frames and based on previous frame object state that includes object position and motion the points are associated. In this method, we need to identify the object in every frame by using an external mechanism.
- **Kernel tracking.** In this method, the object is tracked by calculating the motion of the kernel in consecutive frames. Kernel refers to the primitive geometric shapes that used to represent the shape of the object.
- **Silhouette Tracking.** Silhouette refers to the region inside the contour of the object. By estimating the object region in every frame, this tracking is performed. Silhouette tracked by either contour evolution or shape matching.

1.4 Motivation

The main theme of our soccer ball Detection and tracking is detecting the soccer ball under different conditions. These conditions include partial occlusion, cluttering and ball speed. When ball is moving with dynamic velocities or varying acceleration we need to detect and track the ball. In the detection phase, we have to find the location of the ball by eliminating all obstacles like player socks, shoes, background object which look like ball. The tracking mainly involves localization of the ball by selecting a feature space. We have done this localization using a similarity function known as Bhattacharyya coefficient where color histogram of the ball is taken as tracking feature. Finally, we need to know which team has the most control on the ball. We have done this using K-mean clustering algorithm which segments the image based on color.

1.5 Thesis Organized

This thesis consists of 4 chapters organized as follows. Chapter 1 briefly explains the applications of soccer video analysis. It also illustrates different stages of tracking the ball and naturally occurring complexities while tracking. Chapter 2 is about detecting the ball. It describes various stages we followed to detect the ball and various methods developed by current researchers. Chapter 3 explains mean shift tracking and how we utilized it to track the ball by overcoming various difficulties. Chapter 4 describes the results and discusses them. It also contains conclusion of our work and future work can be done on this topic.

Chapter 2

Ball Detection

2.1 Introduction

Movement of the ball has much importance in the field of soccer video analysis. Finding the path of the ball is the primary thing to know the ball movement. For this purpose, we need to track the ball. Detection of the ball is the first step to track the ball. This chapter explains how to detect the ball using our proposed method. This chapter clearly explains the different complexities occurs while detecting and how to resolve them. This chapter also describes how to represent ball with a geometric shape to make it useful for tracking phase.

In the section 2 we have explained the different methods followed by various researchers. In the section 3 we have given our proposed algorithm and the succeeding chapter sections explains thoroughly about different steps given in section 3 with necessary algorithms.

2.2 Literature Survey

– Bodhisattwa Chakraborty[2] proposed a method to detect the ball. The method consists mainly two parts:

1. **Moving Object Detection.** In this, we build a background model using *Recursive Approximate Median Method* that explained in the algorithm 1. In the next step, the present frame is deducted from the background and result is thresholded to distinguish foreground or moving objects.

Algorithm 1: RECURSIVE Approximate Median Method

```
Consider the first frame as background frame.
Find the modulus difference of each frame from background.
if The value of the background pixel is less than that of current pixel then
    backgroundpixel+1.
else
    backgroundpixel-1.
```

2. **Ball Candidate Identification.** To identify the ball candidates, we pass the output of the first step through two strainers. They are:
   (a) **size strainer.** It will output those objects that are in size of ball.
   (b) **Shape Strainer.** It will output those objects that are in shape, circular, of the ball.

After following above steps, we will remain with the Ball candidate.

– Hongying ZHANG[21] proposed a method detect the ball based on color and Hough transform. The necessary steps of the process are as follows:

1. **Enhancement of Image:** The original images are enhanced using Retinex Algorithm[18].

2. **Ball Candidate Detection:** The possible centroids of the Balls are located using Hough Transform.

3. **Color Definition:** Converting the GN(Grey World Normalization) image into HSV image, then using GN-H, GN-S as the definition of color.

4. We measure the weighted mean to every center pixel and its 8-neighborhood pixels about their GN-S and GN-H values. If the mean values of GN-H and GN-S is in between [GN-Smin, GN-Smax] and [GN-Hmin, GN-Hmax], then this center position is the center of our identified ball and output the results end the procedures.

– Amit Kumar K.C[14] proposed a method to identify the ball using Spatio-temporal template matching. The method is as follows:

1. **Foreground Mask Detection.** Build the foreground mask. Foreground mask is the one which divides the moving objects in a frame from the stationary ones in it.

2. Assuming that ball color is known to us, we highlight the moving objects pixels that are having the color of ball in foreground mask.

3. Find the sum of the correlation between the silhouette templates of ball that are projected in multiple views and highlighted foreground objects. Threshold the results. The object which is having value more than threshold will be declared as ball.

– Sophie Xiaofan Liu[16] proposed a method based on foreground detection to detect the ball. The method is as follows:

1. **Background Learning.** Take few starting frames as training frames. By using running average method build the background. Find the variance of the current frame that will be used as threshold in future.

2. **Threshold using Standard Deviation.** Variance based threshold function is used to threshold because it is so simple to implement and updates with

illumination changes. A foreground mask will be generated by thresholding the current frame using following criteria:

$$current - pixel > background - threshold \tag{2.1}$$

$$current - pixel < background + threshold \tag{2.2}$$

3. **Remove Unwanted Noise.** This portion is for morphological operation on foreground mask. Dilation, Erosion are the simple operations in morphology. By using these morphological function we delete the objects that have 1)area 2)length 3)width more than ball.

4. **Ball Detection.** The region that having size equal to ball is declared as ball.

– J. Hossein-Khani[12] proposed a 5 step method to detect and track the soccer ball in a video sequence. In these 5 steps the first three steps dedicated to detect the ball. The 3 steps are as follows:

1. **Play-Field Detection.** We know that distinct dominant color in the soccer video is color of play field that is green. So, by detecting the green using some filter we can detect the play field. After detecting play field we will remain with field lines, players and ball.

2. **Field-line Detection.** Field lines that are in color white that is the color of the ball have more chances to misguide our procedure to detect the ball. So we need to detect them. Hough transform can detect them and later we remove them. We apply some morphological operations to remove unwanted noise from field lines deleted foreground mask.

3. **Ball Detection.** Ball that is in shape circular and having color white can be easily detected using shape sieve and color sieve. Hough circle detection used to detect the object which are in shape circular in current frame. The size of the ball also considered as one of the option to detect the ball. A size sieve used to find the size of the object.

2.3 Proposed Method

Profiting from the techniques proposed by past scientists, the accompanying steps are intended to recognize the ball:

1. Play Field Detection
2. Edge Detection
3. Moving Object Detection
4. Conjunction of Above 2 stages
5. Morphological Filter and Connected Component Analysis
6. Shape Strainer
7. Size Strainer
8. Dominant color Extraction
9. Neighborhood Condition Checking

In figure 2.1 we have shown different stages to detect the ball.

2.4 Play-field Detection

As a rule, soccer video frames comprise of players, ball, play-field, notices, crowd et cetera. On the other hand, some of them are immaterial to the event of occasions, for example, crowd and notices. As the first step, the fundamental and helpful data need to be separated from the entire picture. Accordingly, everything aside from the playfield must be evacuated.

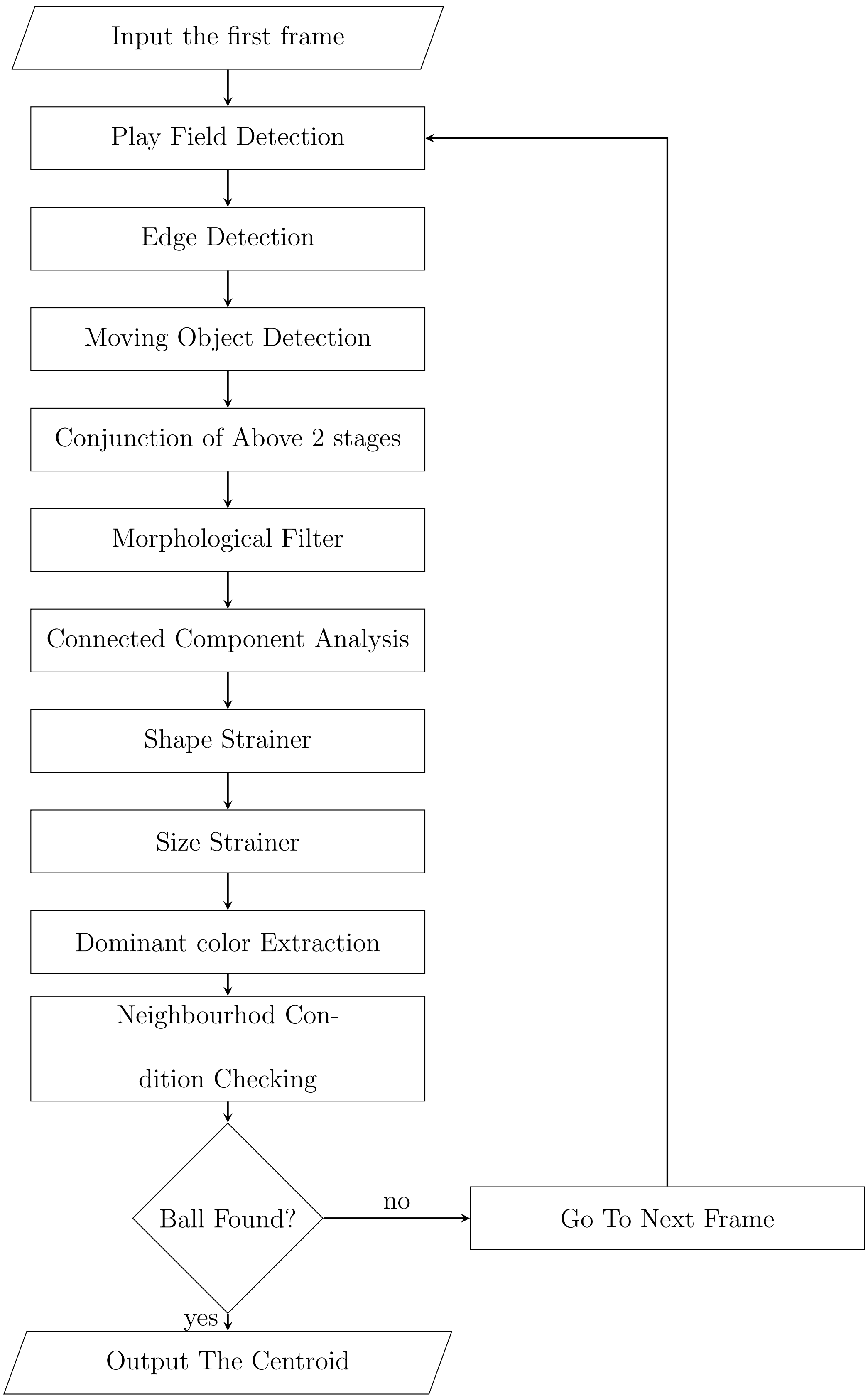


Figure 2.1: Ball Detection Stages

To this end, color can be utilized as the primary highlight, on the grounds that a soccer field has one particular predominant color, i.e., green. In any case, there are a few downsides here that ought to be considered. The shading of playfield shifts from the Stadium to the stadium and even relies on upon the illumination conditions in which the video sequences are recorded.

Figure 2.2: Play Field Detected Image

The particular prevailing color is depicted by mean estimation of every color segment, registered around their individual histogram crests. The fundamental issue in these methodologies is the determination of proper thresholds. Consequently, this technique may not yield alluring results. Besides, a few pixels that fit in with the playfield are not recognized accurately, or a few pixels that don't have a place with the playfield are removed as the playfield.

We detect the predominant color using following constraint

$$Playfield\,(x,y) = G\,(x,y) > R\,(x,y) > B\,(x,y) \tag{2.3}$$

x,y - coordinates of the pixel,

The pixes that are holding $G > R > B$ condition true declared as playfield.This condition additionally remains constant for the gray color pixels. In this way, protests, for example, field-lines, ball, player's socks, and goalpost may be distinguished as playfield as well it can be seen in figure 2.2. The part of the picture that is not in black is identified as playfield.

Algorithm 2: PLAYFIELD Detection Algorithm

Input: Video frame V having N number of pixels

Output: PlayField Detected Farme $V_{GroundDetected}$

for $i \leftarrow 1$ ***to*** N **do**
  if $V_G\left(i\right) > V_R\left(i\right) \cap V_G\left(i\right) > V_R\left(i\right)$ **then**
    Do nothing.
  else
    $V_G\left(i\right) = 0.$
    $V_R\left(i\right) = 0.$
    $V_B\left(i\right) = 0.$

2.5 Edge Detection

Our next venture in the Algorithm is to uproot players, field lines. To do this we first try for Edge Detection of the video frame. Edge detection is the methodology utilized most much of the time for fragmenting pictures in light of unexpected nearby changes of intensity. The idea of an edge is discovered much of the time in discourses dealings with regions and boundaries. Edge detection can be considered as a kind of picture division by separating the items. To discover edges we have diverse operators which are convolved with the frame.

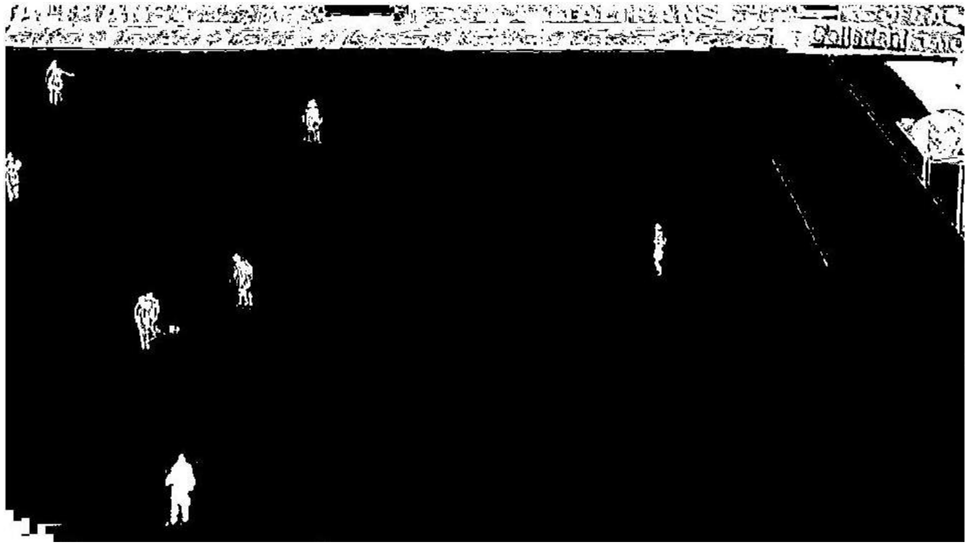


Figure 2.3: Edge Detected Image

2.5.1 Sobel Opertor

Gradient along x-direction V_X is given by convolving image with following matrix.

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Gradient along y-direction V_Y is given by convolving image with following matrix.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Edge detected image is finally given by following equation

$$V_E = \sqrt{V_X^2 + V_Y^2} \tag{2.4}$$

2.5.2 Canny Edge Detector

In canny Edge detection, we first go for smoothening of frame by Gaussian filter and afterward sobel operator is applied. The consolidated mathematical statement for this operation is given by

$$V_E = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.5}$$

figure 2.3 shows edge detected soccer frame.

Algorithm 3: EDGE Detection Algorithm

Input: Video frame V having N number of pixels

Output: Edge Detected Farme $V_{EdgeDeteted}$

for $i \leftarrow 1$ ***to*** N **do**

$V_X(i) = V_{Neighborhood}(i) * SobelhorizontalOperator$

$V_Y(i) = V_{Neighborhood}(i) * SobelVerticalOperator$

$V_E = \sqrt{V_X^2 + V_Y^2}$

$V_{combined} = V_{EdgeDetected} + V_{GroundDetected}$

Algorithm 3 shows Edge detection. The method followed to execute this process has been shown in the algorithm3. Later we combine this result with ground detected result. The advantage we got from this step is we get the full area of every object perfectly. For example, the silhouette of the players, Ball, etc.The result of the process is shown in the figure2.3.

2.6 Moving Object Detection

The background subtraction is the most prominent and regular methodology for motion detection. The thought is to subtract the present image from a reference background image, which is overhauled amid a time of time. It functions admirably just in the vicinity of stationary cameras. The subtraction leaves just non-stationary or new protests, which incorporate whole outline locale of an article. This methodology is straightforward and computationally reasonable for constant frameworks, however are greatly delicate to element scene changes from lightning and unessential occasion and so forth. In this way it is profoundly subject to a decent background support model.

Algorithm 4: MOVING OBJECT Detection Algorithm

Input: $V_i = current frame, V_{i+1} = next frame, V_{i-1} = previous frame$ having N number of pixels each.

Output: Edge Detected Farme $V_{MovingObjectDetected}$

1 $B1\left(j\right) = V_i\left(j\right) - V_{i+1}\left(j\right)$

2 $B2\left(j\right) = V_i\left(j\right) - V_{i-1}\left(j\right)$

3 **Threshold** the $B1$ and $B2$

4 $V_{MovingObjectDetected}\left(j\right) = B1\left(j\right) \cup B2\left(j\right)$

we have followed two types of background subtraction methods here.

- simple background subtraction .
- proposed backgroun subtraction.

2.6.1 Simple Background Subtraction

In this we consider past frame as the reference frame for the present frame. The reference frame will be differentiated from the current frame every time and thresholded to determine which of the pixels are foreground and which are background.

Simple background subtraction method is simple in terms of implementation. If there is a minor changes in the background it lead to a very bad result.

$$D_t\left(x,y\right)=\begin{cases}1, & if \quad \left|I_t\left(x,y\right)-R_t\left(x,y\right)\right|>\tau\\ 0, & otherwise\end{cases} \tag{2.6}$$

- $I_t\left(x,y\right)$-current frame.
- $R_t\left(x,y\right)$-reference frame.
- $D_t\left(x,y\right)$-detected mask.
- τ-threshold

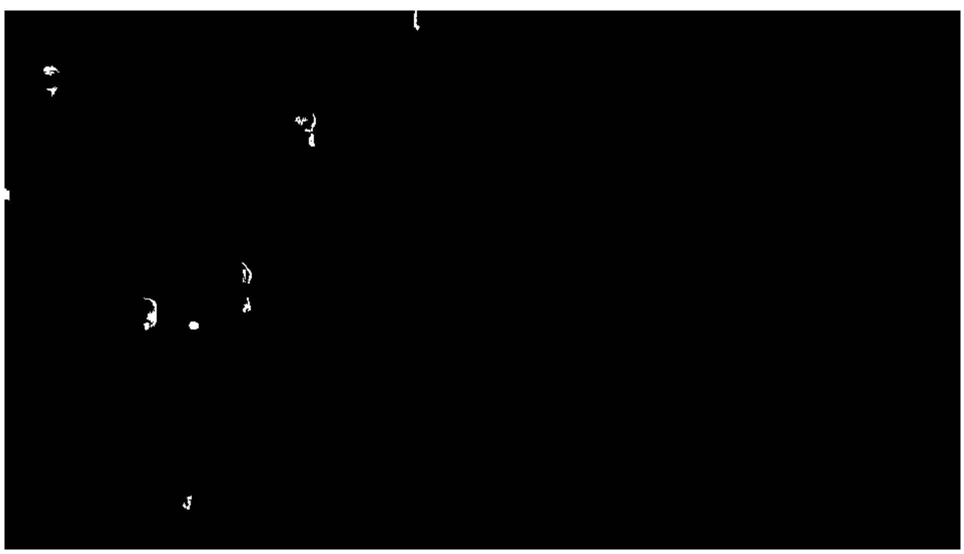


Figure 2.4: Moving Object Detected Image

2.6.2 Proposed Background Subtraction

In this method, both past and future frames are used as reference frames. Both will be subtracted from current frame. The result of both subtractions combined later to get moving object detected mask.

$$D_t\left(x,y\right) = \begin{cases} 1, \quad if & \begin{array}{c} \left|I_t\left(x,y\right) - I_{t-1}\left(x,y\right)\right| > \tau \\ \textbf{OR} \\ \left|I_t\left(x,y\right) - I_{t+1}\left(x,y\right)\right| > \tau \end{array} \\ 0, \quad otherwise & \end{cases} \tag{2.7}$$

2.7 Conjunction of Above 2 Stages

Our ultimate aim after ground detection is to remove players, field lines and stationary objects like goal post, ad holdings, crowd et cetera. This conjunction step is the final step to remove field line, goal post, ad holdings, crowd and stationary elements but not active players in the frame. It makes a way to find the ball in the frame.

If we observe figure 2.3, it contains everything like field lines, players, ball except ground. If we look at the figure 2.4, it contains an only partial region of moving objects. But our aim is to delete field lines and we want the full area of the moving objects.

Algorithm 5: CONJUNCTION Stage

Input: $V_{MovingObjectDetected}, V_{EdgeDeteted}$ having N number of pixels each.

Output: Field line deleted $V_{Conjuncted}$

for $i \leftarrow 1$ ***to*** N **do**
    if $V_{MovingObjectDetected}\left(i\right) = 1$ **then**
        $V_{Conjuncted}\left(i^{th} neighbourhood\right) = V_{EdgeDeteted}\left(i^{th} neighbourhood\right)$

For this purpose, we go for the conditional conjunction of edge detected frame and moving object detected one. The method we followed is shown in the algorithm 5. The figure 2.5 shows the output after conjunction.

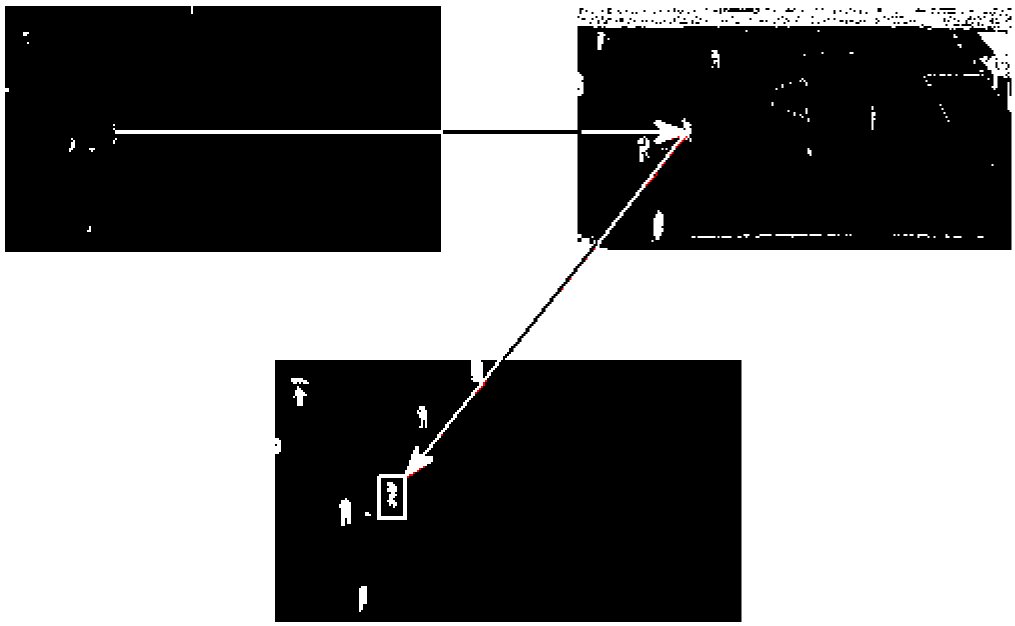


Figure 2.5: Conjunction Image

2.8 Morphological Filter

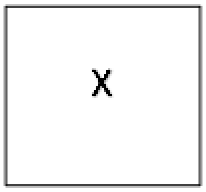
After conjunction, we may remain with some noise. To remove this noise, we go for morphology. Dilation and Erosion are the primary operations in morphology. Opening and Closing are the secondary operations. The word morphology refers to the extraction of necessary regions in an image.

2.8.1 Dilation:

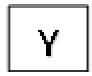
Dilation refers to grows or thickens objects in a binary image. Dilation is represented by $X \oplus Y$. We use a structuring element to do this. Structuring can be anything like square, rectangle, etc. dilation is defined as

$$X \oplus Y = \{z| (Y)_z \cap X \neq \phi\} \tag{2.8}$$

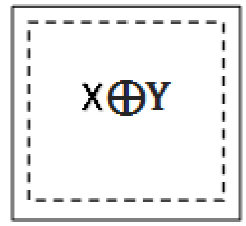
The above equation depicts that the dilation of X by Y is the set of all changes, z, such that Y and X overlay by at least single element. Dilation operation shown in figure 2.6.



(a) Set X



(b) Structuring Element Y



(c) Dilation of X by Y

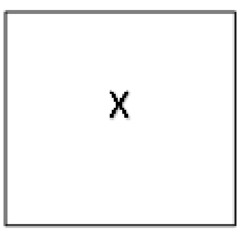
Figure 2.6: Dilation

2.8.2 Erosion:

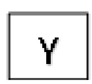
Erosion refers to shrinks objects in a binary image. Dilation is represented by $X \ominus Y$. We use a structuring element to do this. Structuring can be anything like square, rectangle, etc. erosion is defined as

$$X \ominus Y = \{z | \left(Y\right)_z \cap X^c = \phi\} \tag{2.9}$$

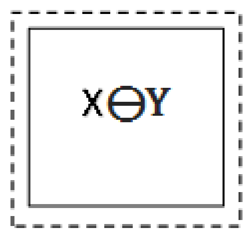
The equation (2.9) depicts that the erosion of X by Y is the set of all points z, such that Y, altered by z, is accommodated in X. Erosion operation shown in figure 2.7.



(a) Set X



(b) Structuring Element Y



(c) Erosion of X by Y

Figure 2.7: Erosion

We apply erosion operation to remove small noise in conjunction image, later, we apply closing operation to retain remaining elements. After these operations, we got the result shown in the figure 2.8.

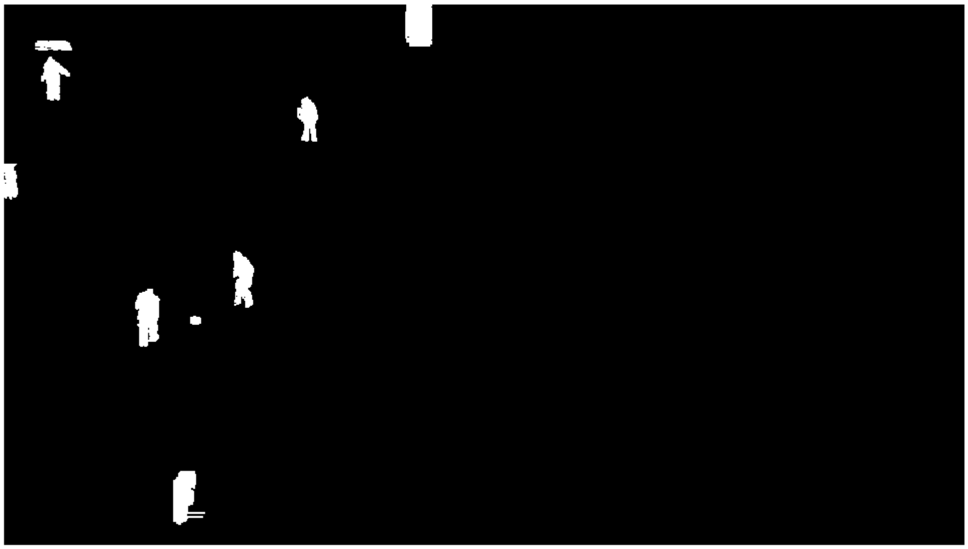


Figure 2.8: Morphed Image

2.9 Connected Component Analysis

The connected component analysis[13] is an important issue in dividing objects based on their physical properties. The connected component refers to a set of pixels that are connected by 4-adjacency or 8-adjacency in a binary image. In the morphed image, we label the all connected sets afterward we find the physical properties like area, perimeter, centroid, etc.

2.10 Shape Strainer

After labeling the image, we pass the video frame through a shape strainer. We know that the ball is in the form of spherical in 3D and circular in 2D. But when it is traverse from one frame to another frame with high-speed motion it looks like a parabola. The shape also alters when ball collided with players, objects in the field, etc. Merging of the soccer ball image with various images in the background too modifies the shape of the ball. The circularity[4] of the ball is defined as the ratio of its major axis length to minor axis length. It is observed that the ratio varies between 0.5 and 1 for football candidate. The process we followed to implement this filter is shown in algorithm 6

Algorithm 6: SHAPE Stainer

Input: $V_{Conjuncted}$ having N number of pixels each.

Output: $V_{With\ Circular\ Objects}$

Get the $V_{Labeled}$ with L connected componenets by labeling the connected components in $V_{Conjuncted}$

Find the major axis length of labeled regions

Find the minor axis length of labeled regions

for $i \leftarrow 1$ ***to*** L **do**

  $circularity\,(i) = \frac{majoraxis\ length(i)}{minoraxis\ length(i)}$

  if *circularity of* $V_{Labeled}\,(i) > 0.5$ **then**

    $V_{With\ Circular\ Objects}\,(i) = V_{Labeled}\,(i)$

  else

    $V_{With\ Circular\ Objects}\,(i) = NULL$

2.11 Size Strainer

There will be many objects in the soccer field that are in shape circle, and size is smaller or bigger than the ball. The shape strainer gives all circular objects as output. Now, this output is passed through a size sieve[1] to get only the objects that are of soccer ball size. It is observed that the size of the ball is varying between 100 and 800 pixels in our video sequences. The method we implemented is explained in the algorithm 7.

Algorithm 7: SIZE Stainer

Input: $V_{with\ circular\ Objects}$ having L number of circular objects.

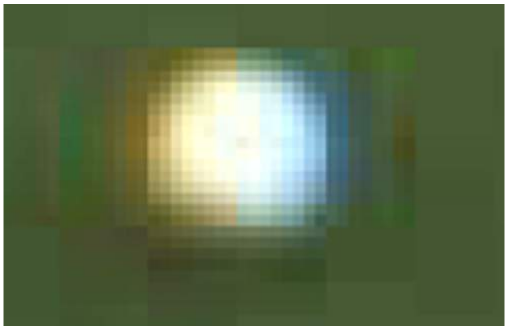
Output: $V_{with\ circular\ Objects\ of\ size\ Ball}$

```
1 Find the area of circular objects
2 for i ← 1 to L do
3     if area of V_Labeled(i) > 100 ∩ area of V_Labeled(i) < 800 then
4         V_with circular Objects of size Ball(i) = V_with circular Objects(i)
5     else
6         V_with circular Objects of size Ball(i) = NULL
```

2.12 Dominant Color Extraction

As shown in the figure 2.9b the soccer ball contains different colors they may be the company name or different designs. Even though it has different colors, the dominant color is white. But we no need consider all these complexity issues why because in long shot soccer ball looks like it has only white color as shown in the figure 2.9a. After shape and size filters, we delete the objects that are not in color white it makes our work simple to detect the ball.

To implement this step we find the Euclidean distance between white and pixel intensity values in the labeled regions. The region that have whiter pixel percentage will be taken into consideration. In experiments, it is observed that the percentage will be more than 75.

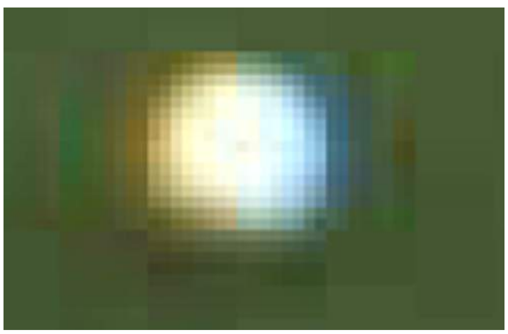


(a) Soccer ball in Long-shot

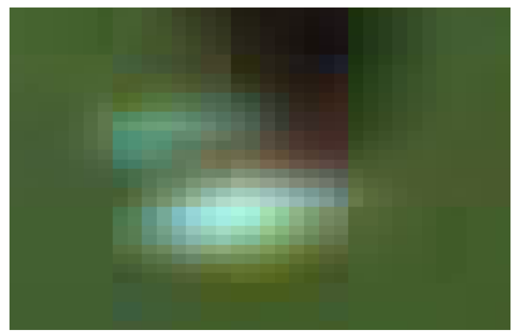
(b) Soccer ball in close-shot

Figure 2.9: Soccer Ball in different Shots

2.13 Neighborhood condition checking



(a) Soccer ball



(b) Socks

Figure 2.10: Soccer Ball in different Shotss

It is the final step of ball detection. In the soccer field, there will be many objects like player socks and player shoes they can be detected as football candidates. For example, consider the player shoes as shown in the figure 2.10b. Besides the soccer ball is shown in the figure 2.10a. If we observe these figures we can easily say that non-ball candidates that are because of players and ad holdings will have a non-green background where the most probable candidate will always have a pure green background.

We implement this step by using an algorithm 8. After this step, we remain with the ball candidate. In case if we did not come up with the result because of complex

Algorithm 8: NEIGHBORHOOD Checking

Input: $V_{with\ circular\ Objects\ of\ size\ Ball\ of\ color\ white}$ having L number of connected regions.

Output: $V_{BallDetected}$

Find the number of pixels $NUM1$ in the the connected regions

Find the number of white pixels $NUM2$ in the corresponding regions in $V_{conjuncted\ image}$.

for $i \leftarrow 1$ ***to*** L **do**

  if $NUM1 = NUM2$ **then**

    $BallRegion = V_{with\ circular\ Objects\ of\ size\ Ball\ of\ color\ white}(i)$

noise like the ball is completely merged with any player or no ball is present in the frame we go to next frame and start the process again.

2.14 Conclusion

In this chapter, we have depicted how to detect a ball within a single frame, if the ball is present there, using different steps. But we have given so many steps to follow which leads to unnecessary computation. Therefore, we can left some of the steps still we will remain with the good result. For example, we can left one of the steps in dominant color extraction and neighborhood condition checking. In Shape Strainer, we have used major and minor axis length ratio to determine the circularity instead we can go for the perimeter-based decision. All the steps explained in this chapter are executed on MATLAB and results are shown in Results and Discussion chapter.

Chapter 3

Mean-Shift Tracking

3.1 Introduction

Mean shift tracker is a popular tracker. We want to track a particular target that having Lot of motion, Lot of zooming in, zooming out. How can we do that? We have this window we want to find the centroid of the window. How we are connecting these centroids become tracks that are shown in blue as shown in figure 3.1. This tracker is highly suitable for tracking non-rigid objects with different colors and textures. It is also highly immune to partial occlusions, small camera motion, rotation and clutter. The mean shift procedure is destined find the more similar target candidate based on histogram calculation and using Bhattacharya coefficient that is going to be introduced.

In this chapter, we explained the basic mean shift in section 1 and 2. We described the tracking algorithm in section 2. Section 3 is dedicated for extensions of the

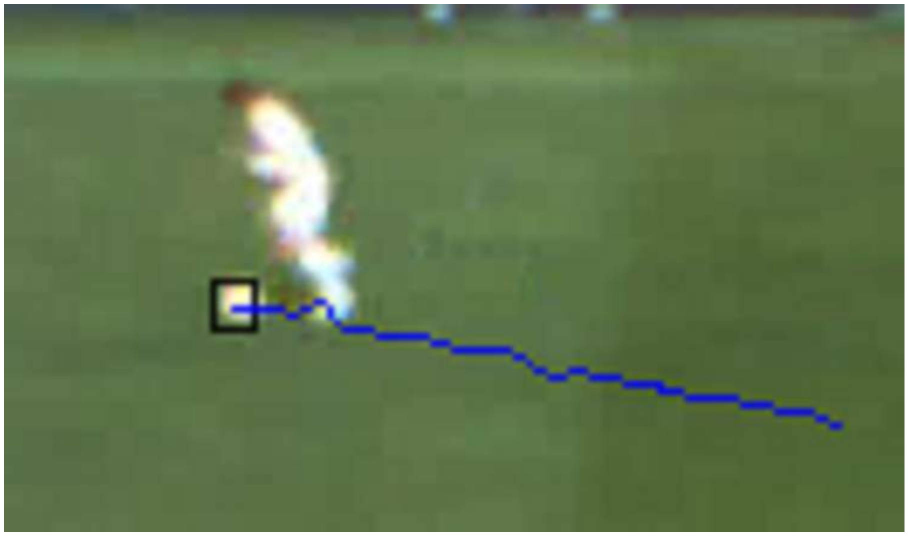


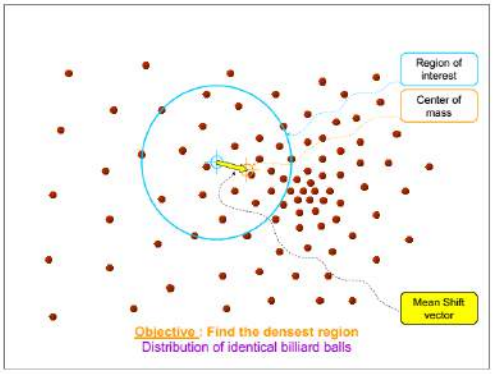
Figure 3.1: Mean shift tracker output

tracking to make it much robust. In the section, we explained the continuous adaptive mean shift.

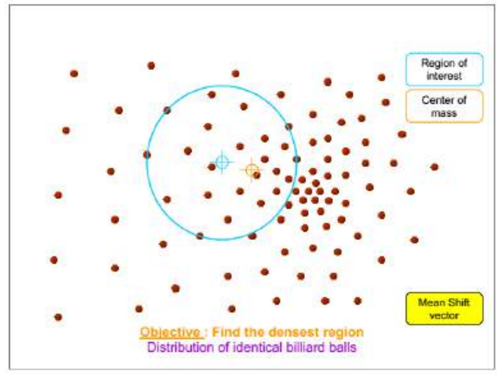
3.2 Simple Mean Shift

We have these billiards balls these are the points as shown in figure 3.2a. These points are in 2D. We want to find the densest region of this distribution. Among the dense areas, we want to find the area that is very dense. We don't know anything so we just randomly start. We have some many shell estimate. We will find the center of mass which is like centroid. We take these points and add them up and divide by number of points will give you center. That will give you a vector from the previous estimate to the center of the mass, and that vector basically is called mean shift vector. Because finding the mean and shifting that and that vector shifts that.

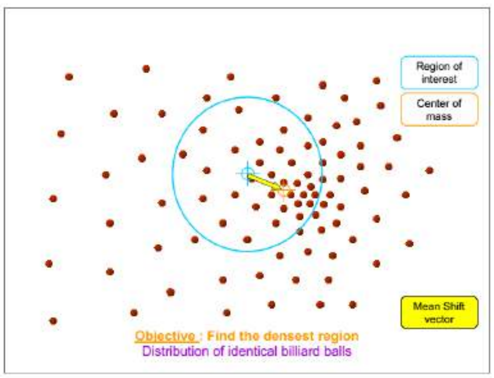
This is the identical distribution of balls our goal is to find the densest region. Star with some initial interest and finding the center of mass and we shift there. We shift things there again we have a region of interest we find the centroid again and then we



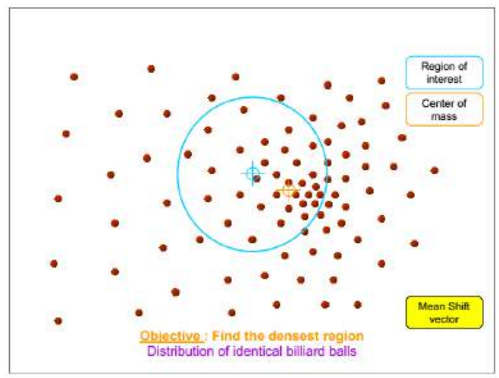
(a)



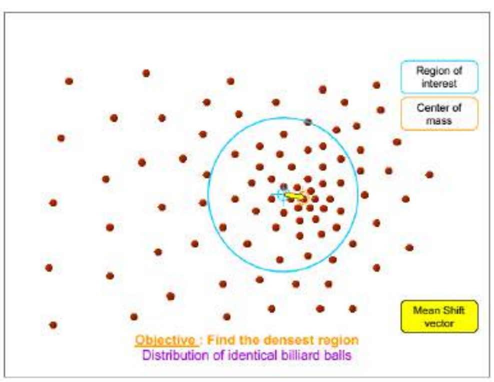
(b)



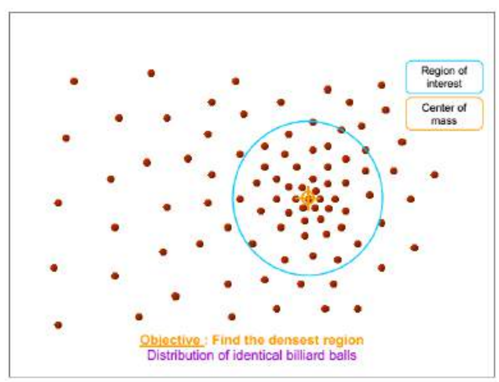
(c)



(d)



(e)



(f)

Figure 3.2: Intuitive Description of Mean Shift

shift there and again we keep doing that finding again centroid and keep moving and like that. This process is giving us mean shift vector how the mean is shifting and once it converges that the region where we have the densest region of data points. The Intuitive Description of the whole process shown in figure 3.2.

3.2.1 Mean Shift Vector

From the given data points, approximate the location of the mean of it by selecting a random point. Estimate perfect spot of the mean of the data by finding the mean shift vector from the initial mean using formula.

$$M_h(y) = \left[\frac{1}{n_x} \sum_{i=1}^{n_x} X_i \right] - y_0 \tag{3.1}$$

We will do it iteratively until mean shift vector is zero. We will compute mean of the points $\{X_1, X_2, X_3, X_4 ... X_{n_x}\}$ are points we sum them up they can be any dimensional in this case 2D. We divide them with the N_x number of points. We subtract the initial mean y_0. And that vector is called mean shift vector.

Mean shift vector[7] always direct towards the maximum increase in density that is the Mean move towards the densest region or where we get convergence more.

3.2.2 Mean Shift Vector(Weighted)

$$M_h(y_0) = \left[\frac{\sum_{i=1}^{N_x} w_i(y_0) X_i}{\sum_{i=1}^{N_x} w_i(y_0)} \right] - y_0 \tag{3.2}$$

– N_x- number of points in the kernel

– y_0- initial mean

– X_i- data points

– h- kernel radius

$W_i$is the weight that depends on the distance from the initial location of the mean$y_0$to that point. Apply the weight finds the average since we are applying the weights we sum up the weights and normalize with that. And subtract from y_0 that is the mean shift vector. N_x number of points in the kernel or region of interest as we had before.

Kernels determine weights. We discuss it later. We have the uniform kernel assigning equal weights as we did before and Gaussian kernel or epanechnikov kernel from the Russian researcher.

3.2.3 What is mean-shift?

This process is an excellent tool to find the mode are the peaks in the distribution. And this can be used for the probability density function. So as we know PDF like Gaussian we can get population or height of people in a room and the samples u can fit Gaussian bell curve that's a distribution Which means that I can find out what's probability that somebody in this room height say 6 feet some body have 5 feet or something as we know that if we use a bell curve then most of the people at the mean/average and very few people will be at end. That is one distribution and another distribution is the uniform distribution. If u roll a dies 6 possible outcomes from 1 to 6 each face of the die has equal probability. So they are many many probability distributions or density functions. Given this data now what we want to do or what mean shift is enough for that it helps us to analyze the distribution that is not parametric. So these distribution Gaussian, uniform, Poisson, exponential there are lot of lots of distributions these are parametric distributions we can write a formula we can have an analytical explanation and we can define those distributions by parameters. For example, for Gaussian we can have mean and standard deviation and for the multidimensional covariance matrix. So then becomes a simple description of density function once we have samples we can compute these parameters then we can forget about samples we don't have to have samples. The problem is many times the distribution is not Gaussian so we can fit a Gaussian distribution into that samples and we will have a problem. The first thing we can do is multiple Gaussians that is why the mixture of Gaussians. So not just one Gaussian use 3 Gaussians that sometimes helps us out. But in some cases we know the mixture of Gaussians does not fit the distribution. Therefore, there are these distributions that are called non-parametric distributions. And idea there is simple that first there is no equation for this distribution, second is we essentially we store all the samples, the advantage of parametric distribution like Gaussian is we don't have to store all these samples Once we have these parameters of the distribution . But here we can model any arbitrary distribution but we have to store all the samples that sometimes becomes a lot of data because there is millions of millions of data will become time-consuming. So now we are going to look into that this is the lots of points in 2D and their distribution is like this in the figure.

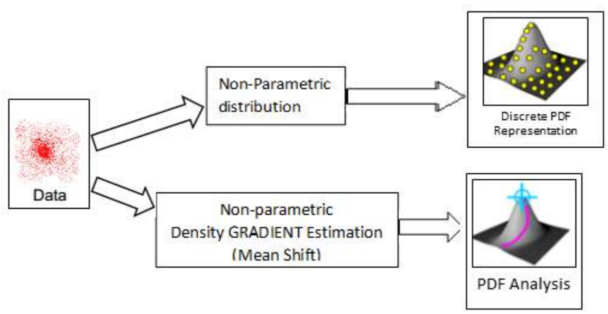


Figure 3.3: PDF of data and its analysis through Mean Shift

It is looks like Gaussian its not a Gaussian .so what we are saying there is x-axis and y-axis then we are saying what is the probability that if I pick up randomly any point here it will have probability in y-axis most of the points will have probability close to peak. Very low probability which will have y-axis value far from the peak. We want use this idea of mean shift and find out the gradient of this distribution and that we can do using the mean shift essentially. We will find the mode that is the peak of this distribution that is non-parametric distribution in that mode corresponding to like gradient of this distribution that we can do using this mean shift.

3.2.4 Non parametric density estimation:

We have a more real data each point has a value in x and y. This may be their distribution. A simple example of distribution is the histogram of an image. In the image, it is 1d we are looking at the intensity but here it is d we are looking at x and y values of the distribution. The distribution and points are related in a way that distribution is telling where these points are. As shown in figure mapping is. Data point density implies PDF value.

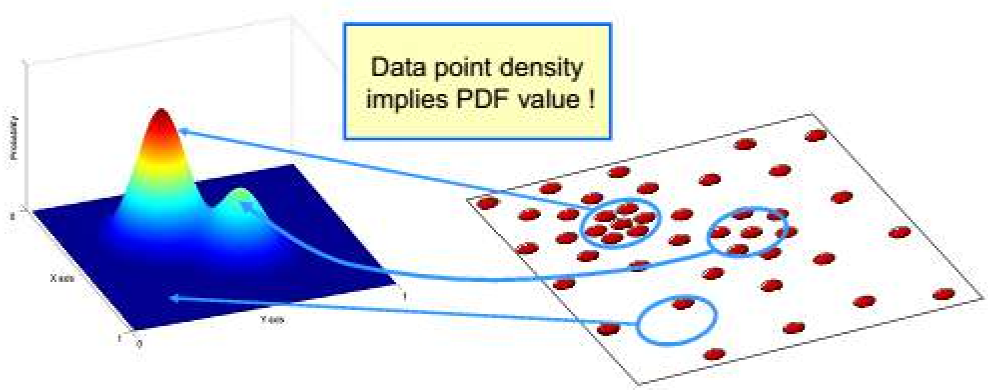


Figure 3.4: Non parametric density estimation

3.2.5 Density Estimation:

In order to represent distribution, we are going to use what is called kernel density estimate. In order to find pdf of x we are going to find the distance of x from all points u_1, u_2 and we are applying exponential operation and we add up all these. Finding the probability of x depends upon contribution of the points and contribution will depend on how close they are to the x. This is called density estimation shown in following equation.

$$PDF(X) = \sum_i c_i e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \tag{3.3}$$

3.2.6 Kernel Density Estimation:

Kernel will tell you depending on distance from x how much weight we want to assign. A function of some finite number of data points $\{X_1, X_2, X_3, X_4 ... X_{n_x}\}$ then kernel given by

$P\left(X\right) = \frac{1}{n}\sum_{i=1}^{n} K\left(X - X_i\right)$

Kernel Examles are

Epanechnikov kernel:

$$K_E\left(X\right) = \begin{Bmatrix} c\left(1 - \|X\|^2\right) & \|X\| \leq 1 \\ 0 & otherwise \end{Bmatrix} \tag{3.4}$$

Figure 3.5: Epanechnikov kernel

Uniform Kernel:

$$K_U\left(X\right) = \begin{Bmatrix} c & \|X\| \leq 1 \\ 0 & otherwise \end{Bmatrix} \tag{3.5}$$

It gives equal weights to all points.

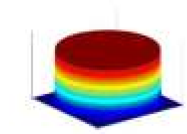


Figure 3.6: Uniform Kernel

Gaussian Kernel:

$$K_N(X) = c.exp\left(-\frac{1}{2}\|X\|^2\right) \tag{3.6}$$

It gives high weights to the points that are close to the center and low weights to the

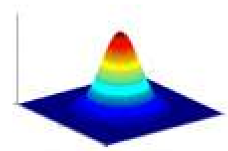


Figure 3.7: Gaussian Kernel

points that are far away from the center.

3.2.7 Properties of a Kernel:

- **Normalized:** $\int_{R^d} K(X)\, dX = 1$
- **Symmetrical:** $\int_{R^d} XK(X)\, dX = 0$
- **Exponential Weight Decay:** $lim_{X\to\infty} \|X\|^d K(X) = 0$

A kernel must posses above properties to get converged in mean shift process.

3.2.8 Profile of a Kernel:

Profile k of a kernel[8] function K is designed such that $K(X) = ck\left(\|X\|^2\right)$ Where $k[0,\infty) \to R$ Now kernel density estimate can be written as

$$P(X) = \frac{1}{n}\sum_{i=1}^{n} K(X - X_i) = \frac{1}{n}c\sum_{i=1}^{n} k\left(\|X - X_i\|^2\right) \tag{3.7}$$

3.2.9 Kernel Density Gradient Estimation

The gradient of the density estimate[5]

$$P(X) = \frac{1}{n}c\sum_{i=1}^{n} k\left(\|X - X_i\|^2\right)$$

is given by

$$\triangledown P(X) = \frac{1}{n}c\sum_{i=1}^{n} \triangledown k\left(\|X - X_i\|^2\right)$$

$$\triangledown P(X) = \frac{1}{n}2c\sum_{i=1}^{n} (X - X_i)\, k'\left(\|X - X_i\|^2\right)$$

$$\triangledown P(X) = \frac{1}{n}2c\sum_{i=1}^{n} (X_i - X)\, g\left(\|X - X_i\|^2\right)$$

where

$$g(X) = -k'(X)$$

A kernel G is defined as

$$G(X) = cg\left(\|X\|^2\right)$$

$$\triangledown P(X) = \frac{1}{n}2c\sum_{i=1}^{n} X_i g\left(\|X - X_i\|^2\right) - \frac{1}{n}2c\sum_{i=1}^{n} X g\left(\|X - X_i\|^2\right)$$

$$\triangledown P(X) = \frac{1}{n}2c\sum_{i=1}^{n} g\left(\|X - X_i\|^2\right)\left[\frac{\sum_{i=1}^{n} X_i g\left(\|X - X_i\|^2\right)}{\sum_{i=1}^{n} g\left(\|X - X_i\|^2\right)} - X\right] \tag{3.8}$$

$$\triangledown P(X) = \frac{1}{n}2c\sum_{i=1}^{n} g_i\left[\frac{\sum_{i=1}^{n} X_i g_i}{\sum_{i=1}^{n} g_i} - X\right] \tag{3.9}$$

Assume $g\left(\|X - X_i\|^2\right)$ as non zero. Note that the gradient of Epanechnikov kernel will be uniform where the gradient of Gaussian normal will remain Gaussian normal. In equation(3.8) the right side term represents mean shift vector that is

$$M_{h,G}(X) = \left[\frac{\sum_{i=1}^{n} X_i g\left(\|X - X_i\|^2\right)}{\sum_{i=1}^{n} g\left(\|X - X_i\|^2\right)} - X\right] \tag{3.10}$$

The location of the new centroid is given by

$$Y = \frac{\sum_{i=1}^{n} X_i g\left(\|X - X_i\|^2\right)}{\sum_{i=1}^{n} g\left(\|X - X_i\|^2\right)} \tag{3.11}$$

3.2.10 Mean Shift Properties

- Automatic speed of convergence
- Close maxima, the steps are little and refine
- The convergence will occur in finite number of steps only
- If we use uniform kernel the convergence will happen in few steps only
- If we use gaussian kernel, exhibits a smooth pathyet is slower than Uniform Kernel

3.3 Bhattachrya coefficient

The Bhattacharya coefficient is utilized as an imprecise measure of similarity between two analytical samples. Assume that the object or target model Z has probability density function Q_z. $Q_z = \{Q_u\}_{u=\{1\ldots m\}}$such that $\sum_{u=1}^{m} Q_u = 1$ where we have the m-bin histogram of the target model. The target candidate with center x0 have the probability distribution $P(X_0)$. $P(X_0) = \{P_u(X_0)\}_{u=\{1\ldots m\}}$such that $\sum_{u=1}^{m} P_u(X_0) = 1$ for the m-bin histogram model. The Bhattacharya coefficient make use of this both probability distributions q and p then the similarity between coefficients is given by

$$\begin{aligned} \hat{\rho}\left(X_0\right) &= \rho\left[\hat{P}\left(X_0\right), \hat{Q}\right] \\ &= \textstyle\sum_{u=1}^{m} \sqrt{\hat{P}_u\left(X_0\right) \hat{Q}_u} \end{aligned} \tag{3.12}$$

The larger the probability of error the more the similarity. The Bhattacharya coefficient can be described as the dot product between p and q. Geometrical representation of Bhattacharya coefficient is the cosine of the angle between m-dimensional unit vectors $\left(\sqrt{\hat{p}_1}, \sqrt{\hat{p}_2}...\sqrt{\hat{p}_m}\right)$ and $\left(\sqrt{\hat{Q}_1}, \sqrt{\hat{Q}_2}...\sqrt{\hat{Q}_m}\right)$. Using above formula distance between two distributions can be depicted as

$$d\left(X_0\right) = \sqrt{1 - \rho\left[\hat{P}\left(X_0\right), \hat{Q}\right]} \tag{3.13}$$

This can be used for object localization since

1. It is better than fisher information.
2. It works good for almost all arbitrary distributions
3. Because of non-parametric nature it interrupts at least one distance axioms.
4. It gives optimal solution every time.

Comparative measures were at that point utilized as a part of computer vision. The Chernoff and Bhattacharya limits have been utilized into focus the effectiveness of edge detectors. The Kullback dissimilarity has been employed as a part of for finding the posture of an object in a picture.

The next section demonstrates to minimize as a capacity of y in the neighborhood of a given area, by utilizing the mean shift iterations. Only the distribution of the object colors will be considered, despite the fact that the texture distribution can be coordinated into the same structure.

3.4 Target Representation

To describe the target, first a feature space is picked. The reference target model is characterized by its PDF Q in the feature space. For example, the reference model can be decided to be the color PDF of the target. Without loss of consensus, the target model can be considered as focused at the spatial location 0. In the upcoming frame, a target candidate is characterized at location y and is portrayed by the PDF $P(X)$. Both PDFs are to be evaluated from the information. To fulfill the low computational expense forced by real-time processing discrete densities, i.e.,m-bin histograms ought to be utilized. Therefore, we have[10]

target model: $Q_z = \{Q_u\}_{u=\{1...m\}} \quad \sum_{u=1}^{m} Q_u = 1$

target candidate: $P(X_0) = \{P_u(X_0)\}_{u=\{1...m\}} \quad \sum_{u=1}^{m} P_u(X_0) = 1$

The histogram may not be the perfect to calculate density estimate but it is quite sufficient for our criteria. We have

$$\hat{\rho}(X_0) \quad = \quad \rho\left[\hat{P}(X_0), \hat{Q}\right] \tag{3.14}$$

to measure percentage of matching/correlation between target candidate and target model. The $\hat{\rho}(X_0)$ plays a crucial role in the in mean shift while converging to target candidate,The maximum likelihood value in $\hat{\rho}(X_0)$ represents that there is a object region which is similar to the Q existing in that frame. We regularize the similarity function by covering the items with an isotropic kernel in the spatial space. At the point when the kernel weights, conveying constant spatial information, are utilized as a part of characterizing the feature space representations, turns into a smooth function in y.

3.4.1 Target model

A target is represented by rectangular region in the image. To get rid of the different target dimensions, the target is normalized to some value, [20 20] in this case, by rescaling the row and column dimensions.

Let us assume that $\{z_i^*\}_{i=1..n}$ be the pixel locations of the target model. These pixels are centered at location 0. A kernel which is radially symmetric, convex and monotonically decreasing in profile[6] assigns weights to the pixels such that the pixels that are close to the center will get more weight and the pixels that are distant from the center will get small weights. This weighting mechanism makes our method more robust due to partial occlusion or cluttering of peripheral pixels.

The function b which represents the feature, color or texture, of the pixel $\{z_i^*\}$ such that $b : R^2 \rightarrow \{1....m\}$ where m represents number of bins. Then the probability of the bin $u = 1...m$ in the target model is given by

$$\hat{Q}_u = C \sum_{i=1}^{n} k(\|z_i^*\|^2) \delta \left[b(z_i^*) - u \right] \tag{3.15}$$

Where δ is the delta function, C is constant which is used impose the condition $\sum_{u=1}^{m} \hat{Q}_u = 1$, so C is given by

$$C = \frac{1}{\sum_{i=1}^{n} k(\|z_i^*\|^2)} \tag{3.16}$$

3.4.2 Target Candiadte

Let us assume that $Z^*_{i\ i=1...n_h}$ be the pixel locations of the target model. These pixels are centered at location X_0. The normalization process is adapted from the target model. Using the same kernel k, but with h bandwidth[9], the probability feature $u = 1...m$ of the target candidate is given by

$$\hat{P}_u (X_0) = C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{X_0 - z_i}{h} \right\|^2 \right) \delta \left[b(z_i^*) - u \right] \tag{3.17}$$

Where

$$C_h = \frac{1}{\sum_{i=1}^{n} k\left(\left\|\frac{X_0 - z_i}{h}\right\|^2\right)} \tag{3.18}$$

Is normalization constant to impose condition $\sum_{u=1}^{m} \hat{P}_u = 1$. Note that C_h independent of the previous center X_0 . for the different values of h we can calculate C_h. The h will decide how much part of the candidate we are considering that is the number of pixels of candidate taken into account for the representation purpose.

3.5 Target Localization

The distance which is a function of X_0 should be minimized to find the location of a target in the current frame. Starting from the location of the target model in the past frame localization process ends with searching in the neighbourhood. Since our distance function(3.13) is smooth, the method uses gradient information that is given by the mean shift vector.

The feature for tracking is selected as color, of course, the same method is used for texture and edges or any combination of them. We assume that the following information is available 1) The perfect localization in target model 2) possible updates of the target model due to considerable changes in color.

3.5.1 Distance Minimization

Minimizing the distance (3.13) is equivalent to maximizing the Bhattacharya coefficient(3.12). The quest for the new target position in the current frame starts at the position of the target in the past frame. In the current frame first we have to calculate the probabilities of target candidate in given position. Using Taylor expansion around the values, the linear approximation of the Bhattacharya coefficient (3.12) is acquired after some changes as

$$\rho\left[\hat{P}(X),\hat{q}\right]=\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{P}_u(X_0)\,\hat{Q}_u}+\frac{1}{2}\sum_{u=1}^{m}\hat{P}_u(X)\sqrt{\frac{\hat{Q}_u}{\hat{P}_u(X)}} \tag{3.19}$$

The above estimation is always true when the present target position $\hat{P}(X)$ does not change much from the previous position $\hat{P}_u(X_0)$. we assume that the $\hat{P}_u(X_0)$ is always greater than zero for $u = 1...m$. By substituting the (31.7) in the equation we get

$$\rho\left[\hat{P}(X),\hat{q}\right]=\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{P}_u(X_0)\,\hat{Q}_u}+\frac{1}{2}\sum_{u=1}^{m}C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{X-z_i}{h}\right\|^2\right)\sqrt{\frac{\hat{Q}_u}{\hat{P}_u(X)}}\delta\left[b(z_i^*)-u\right] \tag{3.20}$$

take

$$w_i=\sum_{u=1}^{m}\sqrt{\frac{\hat{Q}_u}{\hat{P}_u(X)}}\delta\left[b(z_i^*)-u\right] \tag{3.21}$$

then

$$\rho\left[\hat{P}(X),\hat{Q}\right]=\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{P}_u(X_0)\,\hat{Q}_u}+\frac{1}{2}C_h\sum_{i=1}^{n_h}w_i k\left(\left\|\frac{X-z_i}{h}\right\|^2\right) \tag{3.22}$$

To maximize the Bhattacharya coefficient (3.12) or to minimize the distance (3.13) we need to maximize the second term in the equation (3.22) why because the second term does not dependent on X. The second term represents the kernel density estimate $k(z)$ about center X and weighted by w_i.

The location of the object changed from the X_0 to X after implementation of mean shift procedure. The new target position is calculated by equation (3.23).

$$\hat{X}_1=\frac{\sum_{i=1}^{n} z_i w_i g\left(\|X_0-X_i\|^2\right)}{\sum_{i=1}^{n} w_i g\left(\|X_0-X_i\|^2\right)} \tag{3.23}$$

3.6 Tracking Algorithm

Before going to Tracking algorithm, we need to explain 2 basic things needed in process. They are Target Representations, Bhattacharya distance minimization. In the preceeding section, we explained both these terms. In this section, we explained tracking algorithm. The algorithm 9 explained the Mean shift tracking procedure.

Algorithm 9: MEAN SHIFT Algorithm

Input: We have Q_u and initial location X_0 in the initial frame

Output: New object location X_1

1 **Initialize** the location of the target in the current frame as X_0.

2 **find** the $\left\{\hat{P}_u\left(X_0\right)\right\}_{i=1\ldots m}$ using (3.17) and compute

$$\rho\left[\hat{P}\left(X_0\right), \hat{Q}\right] = \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{P}_u\left(X_0\right)\hat{Q}_u}$$

.

3 **Evaluate** Bhattacharya coefficient using (3.12).

4 **Find** the weights $\{w_i\}_{i=1\ldots m}$ using (3.21)

5 **Derive** the new location using equation (3.23)

6 **Update** $\left\{\hat{P}_u\left(X_0\right)\right\}_{i=1\ldots m}$ and compute

$$\rho\left[\hat{P}\left(X_1\right), \hat{Q}\right] = \sum_{u=1}^{m}\sqrt{\hat{P}_u\left(X_1\right)\hat{Q}_u}$$

.

7 **while** $\rho\left[\hat{P}\left(X_1\right), \hat{Q}\right] < \rho\left[\hat{P}\left(X_1\right), \hat{Q}\right]$ **do**

8 $\quad X_1 \leftarrow \frac{1}{2}\left(X_0 + X_1\right)$

9 **if** $\|X_1 - X_0\| << \varepsilon$ **then**

10 $\quad$ **Stop**

11 **else**

12 $\quad$ **set** $X_0 \leftarrow X_1$

13 $\quad$ **Go** to step 2

14 **return** X_1

3.6.1 Implementation Of The Algorithm

The implementation of mean shift algorithm is much easier than the presented algorithm 9. The step 5 is not required because Bhattacharya in 99.9 percent cases leads to increase in the value. Therefore, we no need to calculate Bhattacharya coefficient in steps 2 and 6. to check the similarity in the end only we calculate Bhattacharya coefficient. In the same way, we no need to go to step 9 because in 99 percent cases the procedure converges towards the target. So, instead of step 9 we repeat the procedure for certain number of times.

3.6.2 Complexities

Even though implementation is easy it has some difficulties.

1. It works fine for slow and steadily moving objects but if the object is moving with high speed and acceleration it fails to track.

2. Color is selected as feature space for tracking. If any object in the background has same color distribution, our algorithm may fail to track the correct target.

3. In clustering applications,Decision of bandwidth parameter h is discriminating. A huge h may bring about erroneous clustering and may combine particular clusters. A little h may bring about an excess of clusters.

3.7 Extensions Of The Algorithm

Here we explained the two extensions of the algorithm to make it robust. Still there are lot of other ways to enhance the algorithm we are not presenting here.

3.7.1 Background-Weighted Histogram

The background information should be considered for the following reasons.

1. The localization of the target will be declined if part of the features of a target to be tracked have their presence in the background.
2. Most of the times, the target model contains features of background so it will make things difficult to localize precisely the target.
3. At the same time, improper use of features of background will affect the scale selection algorithm and in turn making things impossible to get similarity across scales.

Let $\{BG_u\}_{u=1\ldots m}$ Be the histogram in the discrete representation of the background in a region around a target such that $\sum_{u=1}^{m} BG_u = 1$. Let us assume that the $\{BG^*\}$ be the smallest non-zero entry in the histogram. The extent of a region around the target will dependent on the application. Then we have to find the weights

$$\left\{G_u = min\left(\frac{BG_*}{BG_u}\right)\right\}_{u=1\ldots m} \tag{3.24}$$

Now the new kernel density estimate of the target model is given by

$$\hat{Q}_u = CG_u \sum_{i=1}^{n} k(\|z_i^*\|^2)\delta\left[b(z_i^*) - u\right] \tag{3.25}$$

such that $\sum_{u=1}^{m} \hat{Q}_u = 1$. And C as normalization constant is given by

$$C = \frac{1}{\sum_{i=1}^{n} k(\|z_i^*\|^2) \sum_{u=1}^{m} G_u\delta\left[b(z_i^*) - u\right]}$$

In the same way the kernel density estimate of the target candidate is given by

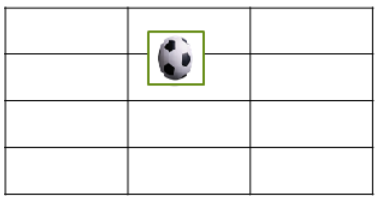
$$\hat{P}_u\left(X_0\right) = C_h G_u \sum_{i=1}^{n_h} k\left(\left\|\frac{X_0 - z_i}{h}\right\|^2\right)\delta\left[b(z_i^*) - u\right] \tag{3.26}$$

such that $\sum_{u=1}^{m} \hat{P}_u = 1$. Normalization constant $C_h$is given by

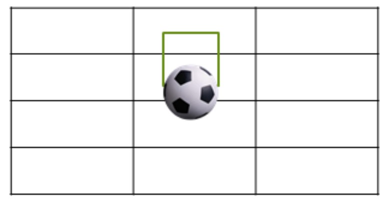
$$C_h = \frac{1}{\sum_{i=1}^{n} k\left(\left\|\frac{X_0 - z_i}{h}\right\|^2\right) \sum_{u=1}^{m} G_u\delta\left[b(z_i^*) - u\right]}$$

3.7.2 Search Area

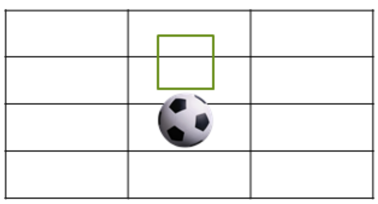
Mean shift tracking will fail when a target is moving with high acceleration or moving with high velocity. In figure current ball position is shown in the figure 3.8a and second figure 3.8b and third figure 3.8c shows next possible positions of the ball. As in second if part of the ball is overlap with the previous ball region then our algorithm will give a good result. But, if it does not have any intersection with as shown in figure then mean shift tracker will fail to track.



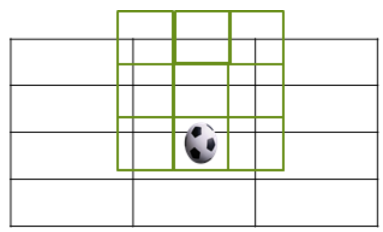
(a)



(b)



(c)



(d)

Figure 3.8: Search Area

To resolve this issue, we will increase the search area that is besides starting from the previous position of the target we will also search in a region neighborhood to the position as shown in figure 3.8d. So we will search for the neighborhood blocks of starting block. The block that has more similarity with target model will be taken for applying mean shift procedure. This process is shown in Algorithm 10.

Algorithm 10: Neighborhood Checking

Input: Previous position and its Neighborhoods of the ball

Output: Most Probable Region Of The Ball

Output: Most Probable Region Of The Ball

Find the color histogram of the target model ch.

find the color histogram of the blocks from $\{ch_i\}_{i=1...9}$. **for** $i \leftarrow 1$ ***to*** 9 **do**

    find $T = ch * ch_i$.

    if $T \geq Throshold$ **then**

        Break.

return ch_i

3.8 False Check

After tracking the object, we need to check whether the tracked one is the required target or not. For this purpose, we follow the following steps.

1. We will draw a bounding box around the target model.
2. In that bounding box we will find the radius of a highest possible circle.
3. In the tracking procedure after tracking every time we will check whether there is a possibility to have an approximately same size circle or not.

If it is possible, we will declare it as a target. Otherwise, we will go for detection algorithm shown in chapter 2 to find a target.

3.9 Flow Chart of Modified Algorithm

After adding the extensions make our algorithm robust to the different situations the steps in ti it looks like flow chart 3.9.

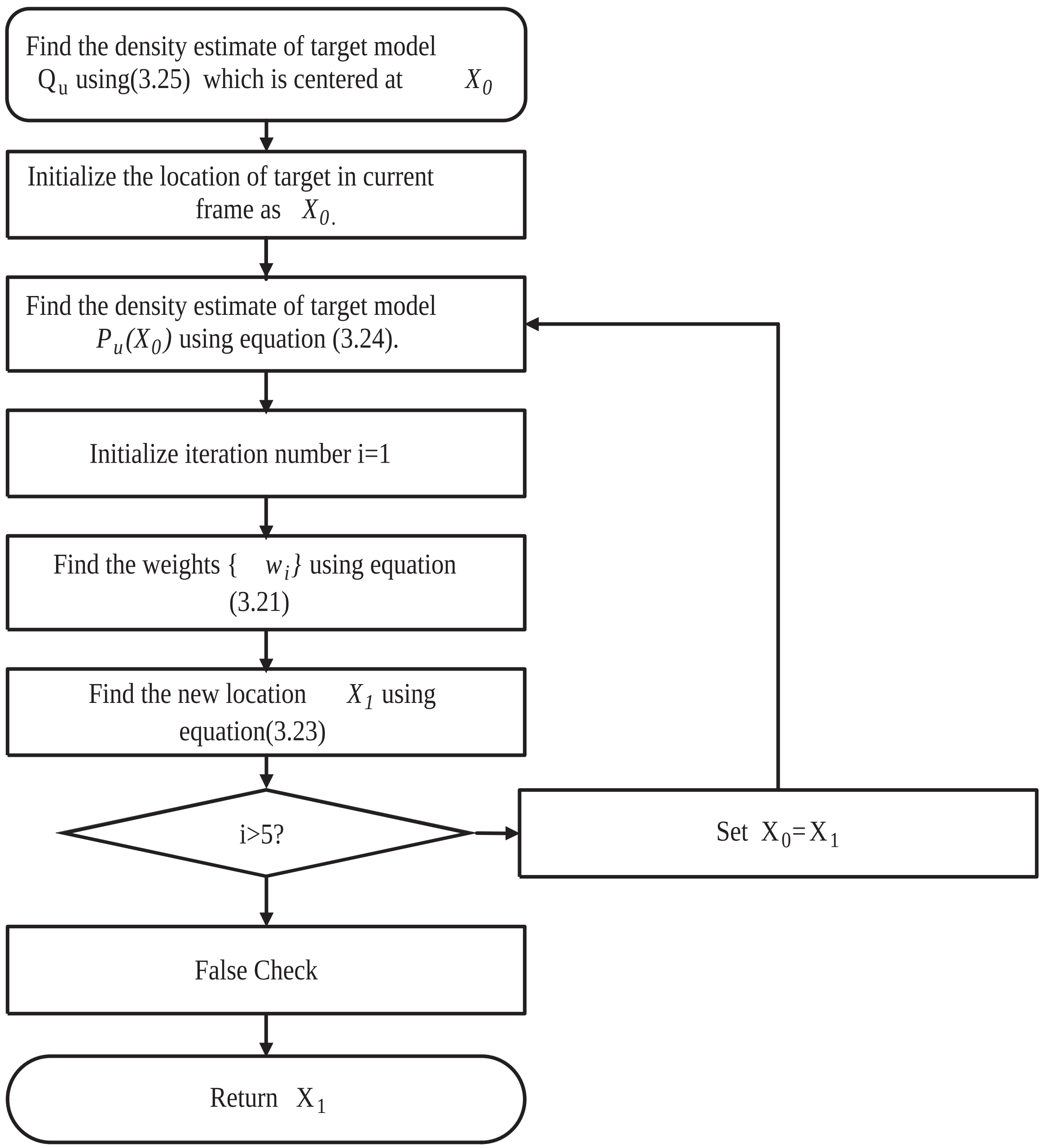


Figure 3.9: Modified Mean Shift

3.10 CAM-Shift Algorithm

CAM Shift[15] is nothing but the advanced version of the Mean Shift procedure. We are not Giving full description of it. But, we will give whole idea of it in figure 3.10

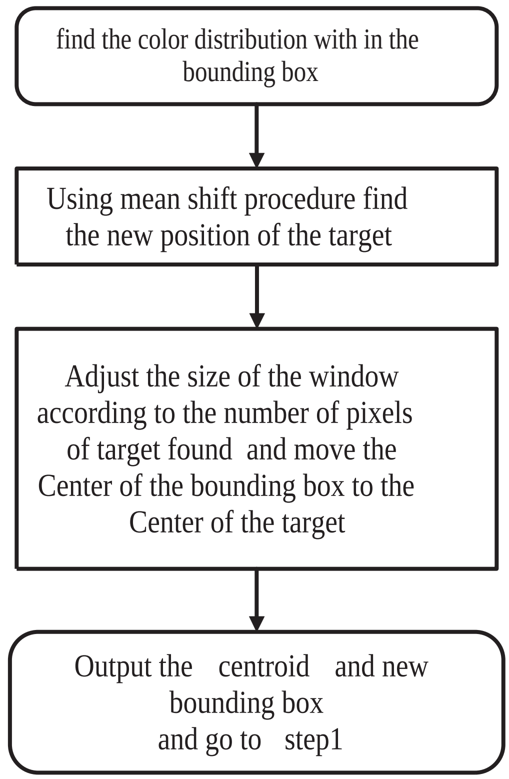


Figure 3.10: CAM-Shift Procedure

3.11 K-Means Clustering

K-means clustering[19] is one of the best methods to segment the image into different clusters based on a feature color. It is a popular way to divide pixels into the various groups. This method gives clusters after few iterations that lead to convergence. The iterations will start by selecting few pixels as centers. After determining centers, in the first iteration, we find the Euclidean distance from each and every pixel of the image to the each and every center. Then the pixels that are close to the first center will consider as the first cluster and pixels that are close to the second center will consider as the second cluster and so on. After the first iteration, we remain with clusters, and we will find the average of every cluster. By taking the averages as centers, we will start the second iteration. We will continue this process until the center in the previous iteration and center in the current iteration are approximately same. The method is shown in the algorithm 11[17, 11].

Algorithm 11: K-MEANS Algorithm

Input: Image with $\{z_j\}_{j=1\ldots N}$ be the pixels

Output: Centroids V_i

1 **Set** the number of clustesr K such that $K \geq 2$.

2 **Initialize** the cluster centers $V_i^{(}0)$ as random values.

3 **set** initial iteration step $s = 0$.

4 **Find** disatance between z_k and V_i using $d_{ik} = \|z_k - V_i\|$

5 **Create** U^s by $u_{ik} = \begin{cases} 1 & d_{ik} = \min_{1 \leq j \leq k}(d_{jk}) \\ & \\ 0 & otherwise \end{cases} \quad 1 \leq k \leq N \quad 1 \leq i \leq K$ where

$U \in V_{KN} \mid u_{ik} \in \{0, 1\}$.

6 **Update** K cluster centers V_{s+1} by $V_i = \frac{\sum_{k=1}^{N} u_{ik} z_k}{\sum_{k=1}^{N} u_{ik}}$

7 **if** $\left\| V_i^{s+1} - V_i^s \right\| < \varepsilon \ \forall i$ **then**

8   **Stop.**

9 **else**

10   **Set** $b = b + 1$

11   **Go** to step 3.

3.12 Conclusion

This chapter discussed thoroughly mean shift procedure. We have discussed the complexities and how to resolve them in this method using extensions. We have implemented all these things and results are shown in results and discussion chapter that is chapter 5. We have also introduced an adaptive method that makes the algorithm much robust. Using K-mean clustering we are going to decide which team has more control on ball.

Chapter 4

Results and Discussion

4.1 Detection Phase Results

In this section results, we got through detection work. In the figure, we have shown one frame from 4 videos we used. In the first video, we have 8 frames in which 4 frames contains the ball. Our algorithm detected 3 frames. In remaining frames, ball is very close to players so while performing morphology operations they are merging with players and recognized as players. In the figure, we have shown one frame from each video. In the first video, we have 8 frames in which 4 frames contains the ball. Our algorithm detected 3 frames. In the figure, we have shown one frame from each video. In the first video, we have 8 frames in which 4 frames contains the ball. Our algorithm detected 3 frames. In the figure, we have shown one frame from each video. In the first video, we have 8 frames in which 4 frames contains the ball. Our algorithm detected 3 frames. Details are shown in the table.

(a) (b) (c) (d)

Figure 4.1: Illustaration Of Detection

Video Name	Number Of Frames	Ball Present	Correctly Detected	Wrongly/Not Detected
Video_1	224	179	159	20
Video_2	99	99	88	11
Video_3	131	131	99	32
Video_4	174	115	115	0

Table 4.1: Detection Phase Analysis

4.2 Tracking Phase Results

In this section, we have shown results that we got through detection and tracking. Our algorithm has given good results when we applied on 4 videos. The results we got are illustrated in the figure. The path of the Ball is shown in blue color by plotting the centroid of the ball in every frame. In every video sequence, the missed ball positions are plotted by the prediction that is connecting the centroid of the ball at its last appearance and the centroid of the ball in current position. The results we got depicted in the table 4.2.

(a)

(b)

(c)

(d)

Figure 4.2: Illustration Of Tracking

Video Name	Number Of Frames	Ball tracked	Predicted Frames	Wrongly tracked
Video_1	224	177	65	2
Video_2	99	99	11	0
Video_3	131	131	0	0
Video_4	174	108	56	10

Table 4.2: Tracking Phase Results

4.3 Control Phase Results

In this section, we have shown the results of control phase. The table shows how much percentage control each team has on the ball. By using K-means clustering, we have decided under which team control ball is in. In the figure 4.3, every picture has two parts. Above figures show the full view of the field. Below figures show the region near the ball. In the figure, we have white color more that mean white shirts team has the ball. In the figure, there are two colors one is white and another one is blue, but the white color is closer to the centroid of the ball then blue. So, white color shirt team has control of the ball.

(a) (b) (c)

Figure 4.3: Illustration Of Control

Video Name	Number Of Frames	Frames Under White Team Control	Frames Under Blue Team Control
Video_1	224	224	0
Video_2	99	99	0
Video_3	131	0	131
Video_4	174	174	0

Table 4.3: Control Phase Results

4.4 Conclusion And Future Work

Here in this thesis we have presented the method to detect and track the ball under partial clutter and occlusion. Naturally we have two types of tracking systems. First one is detecting in every frame and connecting detected points. But the detection always involves a lot of morphological operations which consumes more time. The second one is detecting in the first frame and considering this value as Priori tracking afterward by selecting one of the features of the target. Suppose think that color is taken as feature for tracking. If you track through any method after some time, it may Mis justify any other object that has the same color as a ball. So we need a false detection mechanism too.

Our proposed algorithm is the one that combines all these three things. In the first frame it goes for detection and second frame it starts tracking through mean shift tracker. After tracking every time, it checks whether the tracked item is a ball or not. If it is a ball, the tracking continues otherwise it goes for detection again. Because of we are doing like this we have saved a lot of time that takes for detection and we have restricted the Mis justification of tracking module.

At the same time, our algorithm also provided an improvisation to the mean shift algorithm to make it robust for high speed moving objects. Naturally a mean shift tracker can track the object that moves with low and constant speed but by adding an extended search we got good results for the ball. The extended search will detect the region, in the neighborhood of previous ball position that is most probable to have the ball.

In the future, our algorithm can be much extended. We have different filters that assist the mean shift tracker to make it robust to occlusion and cluttering. These filters include Kalman filter, extended Kalman filter, particle filter. These filters are much useful to predict the ball position in the next frame and we can apply mean shift tracker to delineate the ball at that position.

Bibliography

[1] Bodhisattwa Chakraborty and Sukadev Meher. Real-time position estimation and tracking of a basketball. In *Signal Processing, Computing and Control (ISPCC), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

[2] Bodhisattwa Chakraborty and Sukadev Meher. A trajectory-based ball detection and tracking system with applications to shot-type identification in volleyball videos. In *Signal Processing and Communications (SPCOM), 2012 International Conference on*, pages 1–5. IEEE, 2012.

[3] Bodhisattwa Chakraborty and Sukadev Meher. A real-time trajectory-based ball detection-and-tracking framework for basketball video. *Journal of Optics*, 42(2):156–170, 2013.

[4] Bodhisattwa Chakraborty and Sukadev Meher. A trajectory-based ball detection and tracking system with applications to shooting angle and velocity estimation in basketball videos. 2013.

[5] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1197–1203. IEEE, 1999.

[6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[7] Dorin Comaniciu and Visvanathan Ramesh. Real-time tracking of non-rigid objects using mean shift, July 8 2003. US Patent 6,590,999.

[8] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

[9] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. The variable bandwidth mean shift and data-driven scale selection. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 438–445. IEEE, 2001.

[10] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

[11] Liju Dong, Philip Ogunbona, Wanqing Li, Ge Yu, Linan Fan, and Gang Zheng. A fast algorithm for color image segmentation. In *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on*, volume 2, pages 685–688. IEEE, 2006.

[12] J Hossein-Khani, H Soltanian-Zadeh, M Kamarei, and O Staadt. Ball detection with the aim of corner event detection in soccer video. In *Parallel and Distributed Processing with Applications Workshops (ISPAW), 2011 Ninth IEEE International Symposium on*, pages 147–152. IEEE, 2011.

[13] Yu Huang, Hongbing Li, Jun Tian, and Hong Heather Yu. Image processing system and method for object tracking, January 1 2013. US Patent 8,345,102.

[14] KC Amit Kumar, Pascaline Parisot, and Christophe De Vleeschouwer. Demo: Spatio-temporal template matching for ball detection. In *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*, pages 1–2. IEEE, 2011.

[15] Dengfeng Li, Hu Li, and Xiuling Wei. Kalman filtering-based modified cam-shift vehicle tracking algorithm for highway traffic conditions. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 14, pages V14–271. IEEE, 2010.

[16] Sophie Xiaofan Liu, Lijun Jiang, Jacob Garner, and Sharayah Vermette. Video based soccer ball tracking. In *Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, pages 53–56. IEEE, 2010.

[17] Peter Lukáč, Róbert Hudec, Miroslav Benčo, Zuzana Dubcová, Martina Zachariášová, and Patrik Kamencay. The evaluation criterion for color image segmentation algorithms. *Journal of Electrical Engineering*, 63(1):13–20, 2012.

[18] Zia-ur Rahman, Daniel J Jobson, and Glenn A Woodell. Retinex processing for automatic image enhancement. *Journal of Electronic Imaging*, 13(1):100–110, 2004.

[19] Amit Yerpude and Sipi Dubey. Colour image segmentation using k–medoids clustering. *International Journal of Computer Technology & Applications*, 3(1), 2012.

[20] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[21] Hongying Zhang, Yadong Wu, and Fan Yang. Ball detection based on color information and hough transform. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, volume 2, pages 393–397. IEEE, 2009.