2D Trajectory Prediction of Tennis Ball Throws



Developed By

**Hamza Zareen**

(IS/106/BSCS/B/E/A-14)

Supervised By

**Mr. Nauman Qadeer**

Project Coordinator

**Sir Bilal Khan**

**Department of Computer Science**

Federal Urdu University of Arts, Science & Technology, Islamabad

**2014-2018**

**Acknowledgement**

We would like to thank Allah Almighty, our Parents and teachers. Especially our supervisor

**Mr. Nauman Qadeer** who kept us towards our goal and being a constant source of inspiration.

We are also grateful to **Sir. Bilal Khan** Project Coordinator. We are extremely thankful and indebted to him for sharing expertise, sincere, valuable guidance and encouragement extended to us.

We would also like to thank Family and Friends who believed in us and kept us motivated what May the condition and situation were.

**Hamza Zareen**

**Abstract**

Ball detection and tracking in Broadcast Tennis Video (BTV) is a challenging task in tennis video semantic analysis. Informally, the challenges are due to the camera motion and the other causes such as the small size of the tennis ball and many objects resembles like ball, while the player, the human body along with the tennis racket is not detected completely. In this document proposed an improved object tracking technique in BTV. In order to track the ball, image diﬀerence is performed and because of fast movement of ball its shape didn’t looks like a circle. So an equation is used to detect circle type objects from that the ball candidates are detected by applying threshold values and dilated. Finally the ball is tracked. Then we used Kalman Filter to predict the next position of the ball. The experimental result shows the proposed approach achieved the higher accuracy in object identiﬁcation, and their tracking. It is achieved a high hit rate and less fail rate for ball tracking.

Contents

[1. Introduction 6](#_Toc522200143)

[1.1 Purpose 6](#_Toc522200144)

[1.2 Scope 7](#_Toc522200145)

[1.2.1 Overall goal /Objectives (practical) 7](#_Toc522200146)

[1.2.2 Application oriented goals 7](#_Toc522200147)

[1.2.3. Definitions, Acronyms and Abbreviations 7](#_Toc522200148)

[1.3 Overview 7](#_Toc522200149)

[1.4 System Main Features 8](#_Toc522200150)

[2. Literature Review 10](#_Toc522200151)

[3. Methodology 13](#_Toc522200152)

[3.1. Video Reader 14](#_Toc522200153)

[3.2. Has Frames 14](#_Toc522200154)

[3.3. Cropping image 14](#_Toc522200155)

[3.4. Image Differencing 18](#_Toc522200156)

[3.5. RGB to Gray 22](#_Toc522200157)

[3.6. Converting image into Binary 26](#_Toc522200158)

[3.7. Dilation of image 30](#_Toc522200159)

[3.8. Closing the Image 34](#_Toc522200160)

[3.9. Binary area open 38](#_Toc522200161)

[3.10. Fill holes in image 39](#_Toc522200162)

[3.11. Opening Image 40](#_Toc522200163)

[3.12. Applying Circle Equation 44](#_Toc522200164)

[3.13. Kalman Filter 49](#_Toc522200165)

[3.2.3. Kalman Filter Calculating Kalman Gain 50](#_Toc522200166)

[3.2.3. Estimating Next State 51](#_Toc522200167)

[3.2.3. Kalman Filter Equations 52](#_Toc522200168)

[4. Screen Snapshots 54](#_Toc522200169)

[5. Future Work / Conclusion 56](#_Toc522200170)

[5.1. Future Work 56](#_Toc522200171)

[5.2. Conclusion 56](#_Toc522200172)

[6. References 57](#_Toc522200173)

[6.1 Additional References 57](#_Toc522200174)

CHAPTER 1: INCEPTION

# Introduction

An automatic analysis of sports video is an interesting area which attracts many research attentions for several applications; Sports video contains rich audio and video information within a well-organized structure. Owing to increase in the growth of videos on broadcast and internet, there is a need to access semantic events among the full length videos arises, The target object i.e., the size of the ball is too small in diﬀerent angles and views. Based on various lighting conditions, the ball may not be visible. Instead of accessing the whole lengthy voluminous videos, access of highlights and skipping the less interesting parts of the videos will save not only the viewer’s time but also the cost. To attract the users the content based views are developed based on their own preferences. Consider the tennis video; the moving object is ball and player around the ground region. Tracking of ball in tennis video faces many challenges, like tracking the ball based on trajectory, is little bit complicated because of fast ball and camera movement. Since the ball focus the attention of viewers in tennis. The main aim of tracking is event detection in Broadcast Tennis Video (BTV) based on the tracking results of ball. In tennis the court length is 78 feet, the width is 27 feet for Singles and 36 feet for Doubles. To track the ball some of the challenges are,

1. We have taken a tennis match video of Nadal VS Federer - Australian Open 2014 - Semi-Final - Full Match HD - From YouTube ([1](https://www.youtube.com/watch?v=q7AiwWwiF_k)) and apply different techniques on it. To achieve ball detection.
2. The target object i.e., the size of the ball is too small in diﬀerent angles and views. Based on various lighting conditions, the ball may not be visible.
3. Tracking the ball based on trajectory, is little bit complicated because of fast ball and camera movement.
4. In this context of tracking a tennis ball, noise is a big issue because of the ball size. Due to the quality of the frame, noise appears very frequently among images, which interferes with the process of object detection.
5. Prediction of ball is done using Kalman filter in which we have used the constant acceleration motion model for 2D prediction of tennis ball.

## Purpose

The objective of this software is to detect, predict and track the tennis ball. It can challenge the umpire’s decision. It can use for match analysis. The Match analyst will use it to predict classify the playing style of particular player and the Tennis player use it to improve his game and to understand opponent’s game. It will predict by learning data set the next location of the ball.

## Scope

### Overall goal /Objectives (practical)

* learning of prediction of spatiotemporal data based upon supervised learning.(experiment from past)

### Application oriented goals

* It will detect, predict and track the tennis ball.
* It can challenge the umpire’s decision.
* It can use for match analysis.
* It will predict by learning data set the next location of the ball.
* It will also predict the non-linear movement of the ball.

### Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| 2-D TPOTBT: | 2-D Trajectory Prediction of Tennis Ball Throws |
| XP: | Extreme programming(Agile) |
| MS Visio: | Designing tool |
| UML: | unified model language |
| GUI: | Graphical User Interface |
| UI: | User Interface |
| FUUAST: | Federal Urdu University of Arts, science & technology |

## Overview

This part of the document contains all the requirements of the 2-D TPOTBT. The functionalities of the application included in this document after a detailed review of previous systems and the present systems.

## System Main Features

* Recognition and Localization of ball within video frames.
* Conversion of localize data into experiment datasets.
* Learning for prediction using datasets.
* Next Trajectory prediction of given initial video of thrown ball.

CHAPTER 2: LITRATURE REVIEW

# Literature Review

Ball, player detection and tracking in Broadcast Tennis Video (BTV) is a challenging task in tennis video semantic analysis. Informally, the challenges are due to the camera motion and the other causes such as the small size of the tennis ball and many objects resembles like ball, while the player, the human body along with the tennis racket is not detected completely. In this paper proposed an improved object tracking technique in BTV. In order to track the ball, logical AND operation is applied between the created background and image diﬀerence is performed, from that the ball candidates are detected by applying threshold values and dilated. Finally the ball is tracked. [[1](../Documents/1-s2.0-S1877050915021717-main.pdf)]

Tennis game annotation using broadcast video is at task with a wide range of applications. In particular, ball trajectories carry rich semantic information for the annotation. However, tracking a ball in broadcast tennis video is extremely challenging. In this chapter, we explicitly address the challenges, and propose a layered data association algorithm for tracking multiple tennis balls fully automatically. [[2](../Documents/c72b644b793f8adc1a411ab643563a88744e.pdf)]

Sports video analysis tools are gaining enormous popularity as they enable enhanced visualization and analysis of the game. An intriguing problem is localizing and tracking the ball in a game of tennis. However, achieving a practically suitable tradeoff between high detection accuracy and speed of the ball is a challenging problem in automatic ball tracking algorithms. In this paper, we propose machine learning based automated ball tracking algorithm in tennis match videos acquired by a camera mounted on a quad copter. We begin with applying a video stabilization technique followed by random forest segmentation for detecting tennis ball candidates in the video frames. [[3](../Documents/07414772.pdf)]

This paper presents a novel trajectory-based detection and tracking algorithm for locating the ball in broadcast soccer video (BSV). The problem of ball detection and tracking in BSV is well known to be very challenging because of the wide variation in the appearance of the ball over frames. Direct detection algorithms do not work well because the image of the ball may be distorted due to the high speed of the ball, occlusion, or merging with other objects in the frame. To overcome these challenges, we propose a two-phase trajectory-based algorithm in which we ﬁrst generate a set of ball-candidates for each frame, and then use them to compute the set of ball trajectories. [[4](../Documents/2007-01-IEEE-TMM-Trajectory-04014227.pdf)]

This paper describes our algorithms for players tracking and ball detection for an automatic broadcast tennis video annotation. The system detects and tracks the players using a robust non-parametric procedure for estimating density gradients called the mean shift algorithm. The basic mean shift tracking algorithm assumes that the target object has to separate sufficiently from background, but this assumption is not always true especially when tracking is carried out in dynamic backgrounds such as in sport videos. To cope with this problem, in our proposed system, we embrace the motion segmentation and use the 8x8x8 color histogram to be feature distribution for mean shift tennis players tracking. In order to determine the players’ actions precisely, the system also detect and track ball positions using frame differencing as well as applying some correlation techniques to eliminate false detections. [[5](../Documents/2010_ICARCV_TennisPlayerandBallTracking_Kosit.pdf)]

Ball locations over frames facilitate tennis video analysis to a great extent. But so far no algorithm is able to obtain satisfactory result in locating the ball in broadcast tennis video (BTV). .flus paper presents a trajectory based algorithm to detect and track the ball in BTV. Unlike the object-based algorithm, it does not decide whether an object is the ball. Instead it decides whether a candidate trajectory is a ball trajectory. The algorithm is able to obtain ball locations for most frames in a BTV, making use of four cues, namely, (1) an anti-model method to produce ball candidates from each frame, (2) a trajectory-based scheme to generate, identify, and extend the ball trajectories from a set of candidates, (3) a method to infer the ball locations according to players’ locations and the points of hitting, (4) a method to estimate missing ball locations from known hall locations. [[6](../Documents/TrajectoryBased%20ball%20detection%20and%20tracking%20algorithm%20in%20broadcast%20tennis%20video.pdf)]

CHAPTER 3: CONSTRUCTION

# Methodology

Ball detections are achieved by frame differencing between the current and consecutive images and an equation which detect the circle type objects as we know that after the player hits the ball its speed increases and in frame it will not look like an circle so this equation extracts all the objects which are near to circle. The results are then verified against the size and shape parameters. But this technique has a problem to solve this problem. I have set a threshold value for the circle equation that if the shape of an object is near to .75 then it will treat it as a circle. To accomplish this task I have used the following techniques. In each frame this procedure is repeated to find the ball.

1. Reading video file by browsing it in from the computer.
2. Now checking that the video has frames or not and count its frame.
3. Now from the counted frames applying the detection procedure to bunch of frames.
4. Then a single frame is cropped with a specific rectangle to limit objects in the image the same rectangle will be used for all images.
5. Then the difference of the image is taken from which we will get the objects which are in motion at that time.
6. Then the image is converted into RGB to gray.
7. Then the image is converted into a binary image.
8. Then a structuring element is set to dilate the resultant image.
9. Then with another structuring element is used to close the objects which are separated while applying the differencing procedure.
10. Then a function is used which will eliminate those objects which has pixels less than 30.
11. Then we will fill the holes like by applying these procedures there may have objects which has holes in them so we will fill those holes.
12. Then we will apply another morphological operator image open to open those objects which are been combined during this procedure.
13. Now we use the equation to find the circle type objects.
14. Then we will apply the threshold value that the object greater than .75 values which will be calculated by the equation will be a ball.
15. Then the ball centroid like its center pixel is detected.
16. The detected centroid values like x and y axis coordinates are then send to excel file to be used for the prediction purpose.
17. These all procedures will apply for all frames to detect the ball.
18. To predict the next ball coordinates or missing coordinates first the x axis and y axis both excel files are loaded in Matlab.
19. Then the Kalman filter is used to predict the coordinates like if there is no coordinates of ball are present then it will only predict the ball at that position and if there are coordinates of ball in file then it will first predict the location then correct the position of the ball by using its equations.

The points which are mentioned above will be explained below in detail.

* 1. Video Reader

Video Reader constructs a multimedia reader object, OBJ that can read in video data from a multimedia file. A filename string specifying the name of a multimedia file will be given to the video reader. There are no restrictions on file extensions. By default, MATLAB looks for the file FILENAME on the MATLAB path.

* 1. Has Frames

The has frame function will check that the video has frames or not and it will extract the frames one by one while using another function read frame which read the specific frame from that video and on which we will then apply all procedures.

* 1. Cropping image

In this we have used imcrop tool of Matlab to crop the image specified part to lower the rate of objects present in the image because if there are more objects in the image it will be more difficult to find the ball coordinates from that image. To crop the image I have used a rectangle that means.

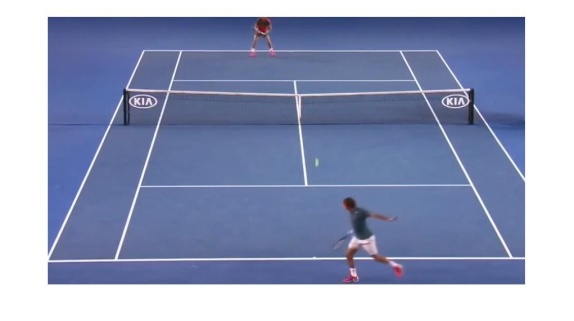
If you specify RECT as an input argument, then the input image can be logical or numeric, and must be real and non-sparse. RECT is double.

In this a rectangle is used of with specific height and width and all frames uses the same rectangle to crop the image because if the same rectangle is not used then it will be difficult to find the exact coordinates of the ball because of pixel values then the pixels values will be changed so we will use the same rectangle so that the ambiguity will not be created. Because if ambiguity created it will all affect the prediction and detection both processes.

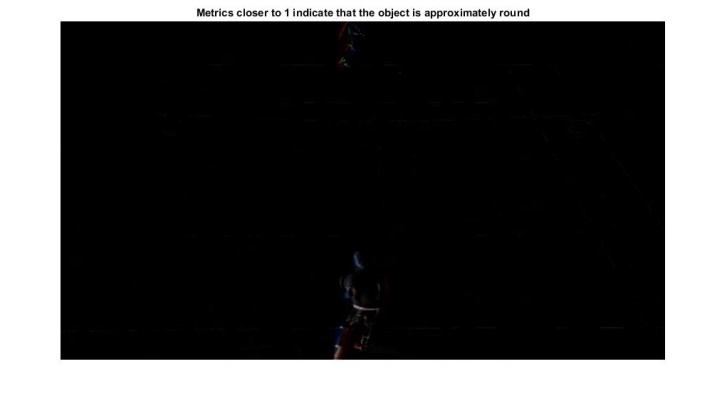
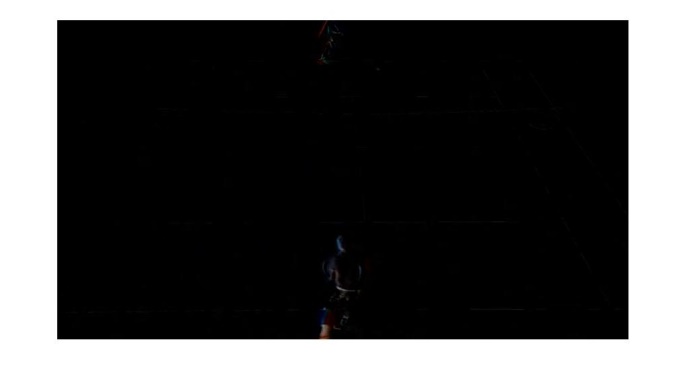
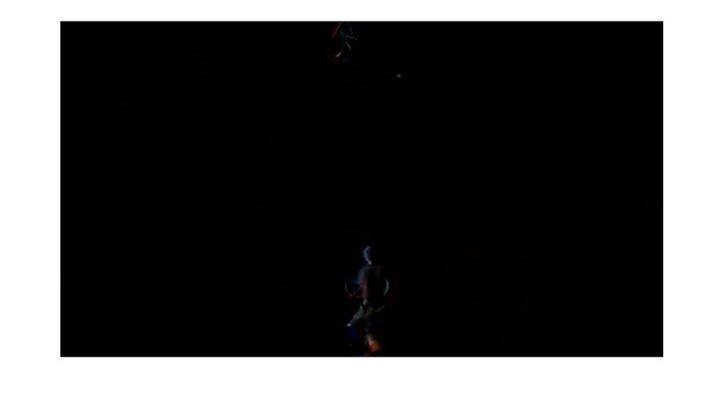
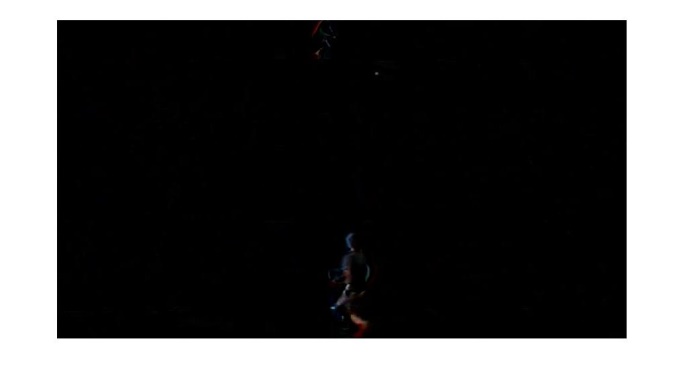
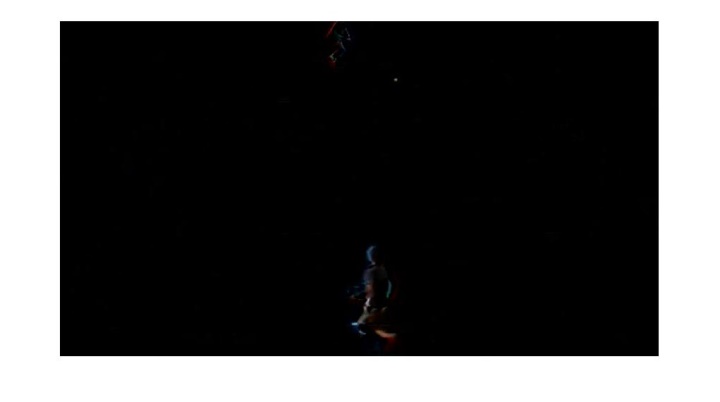
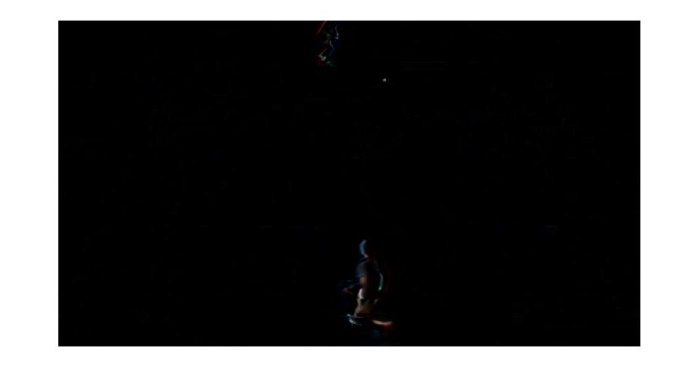
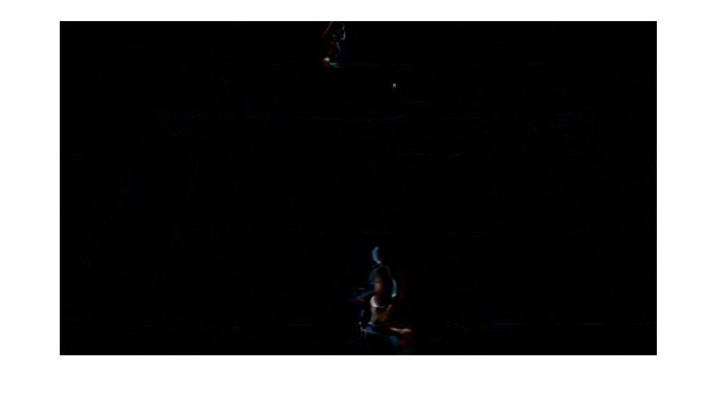
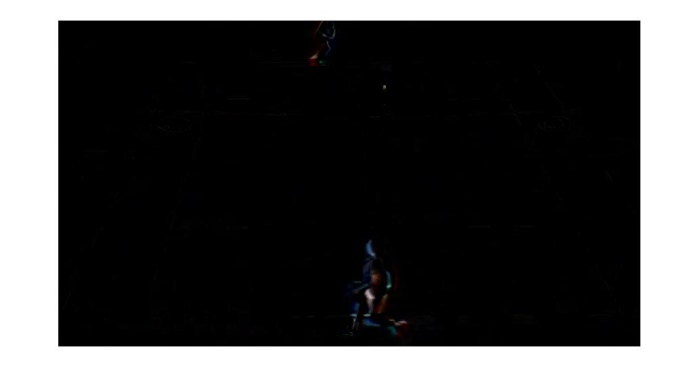
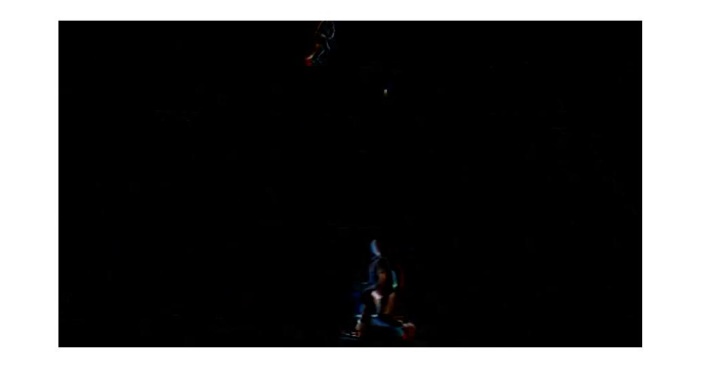
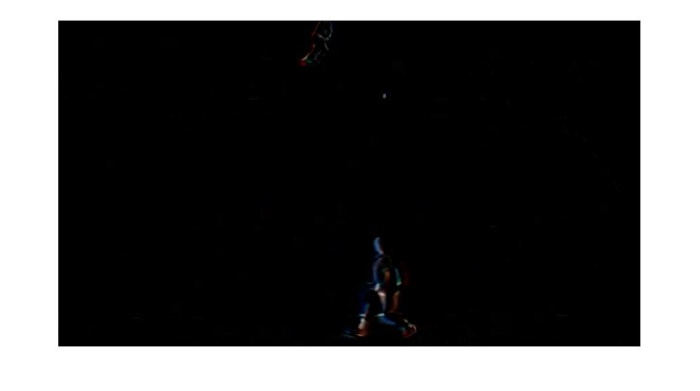
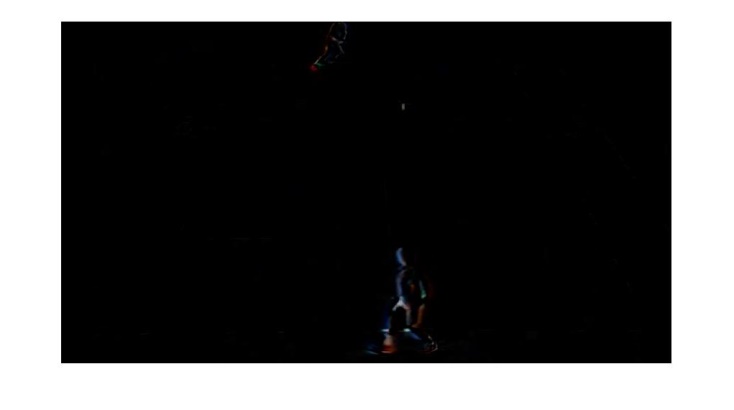
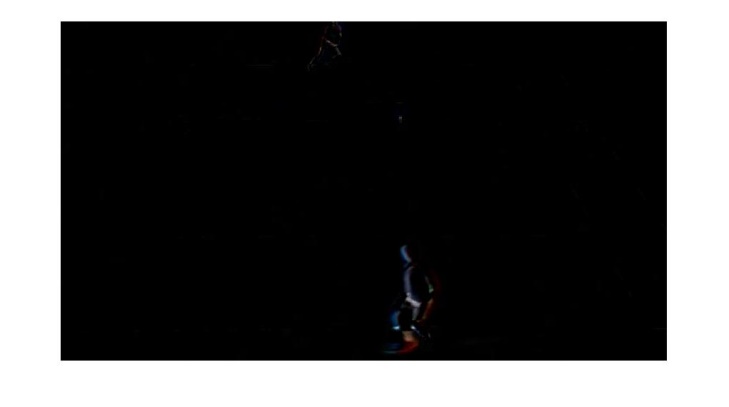
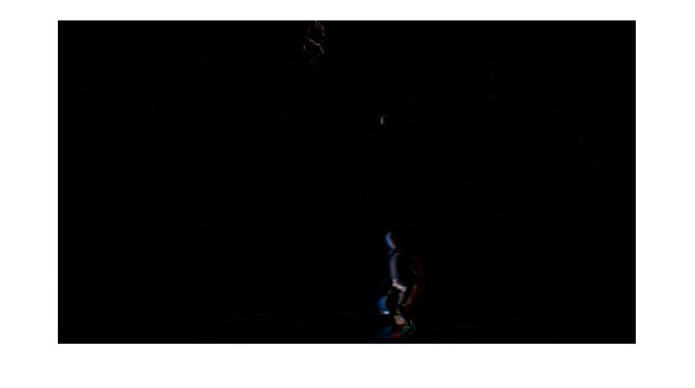
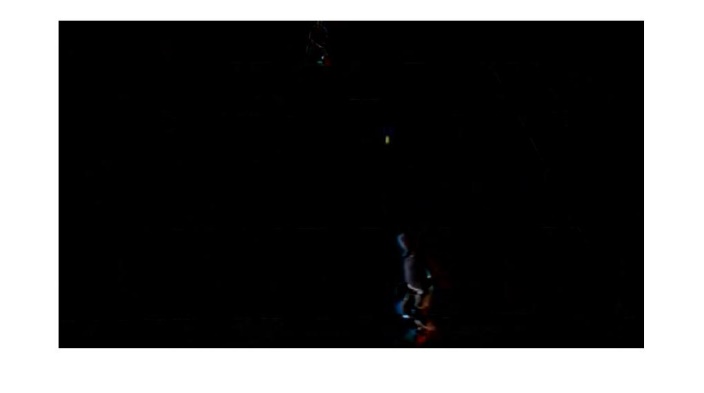
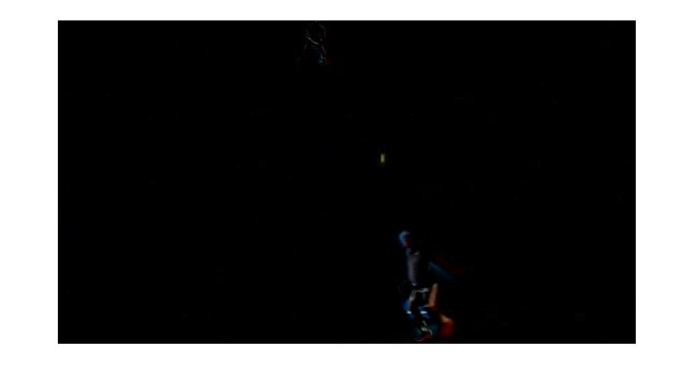
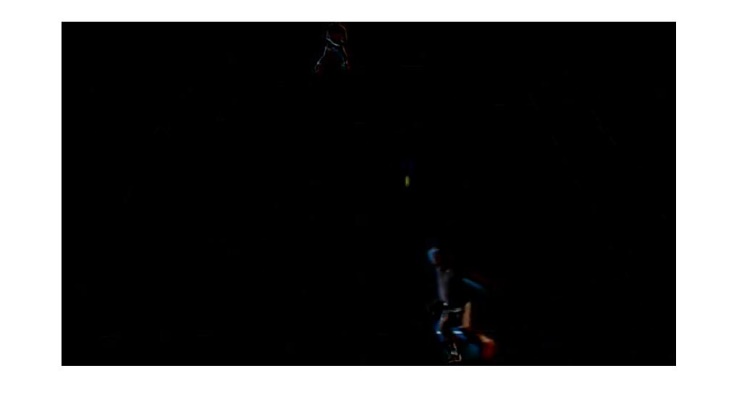
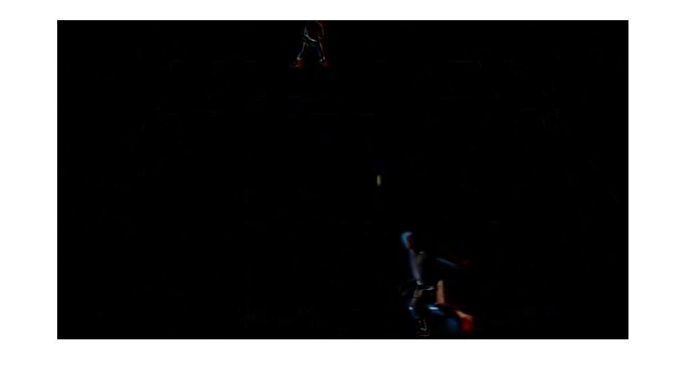
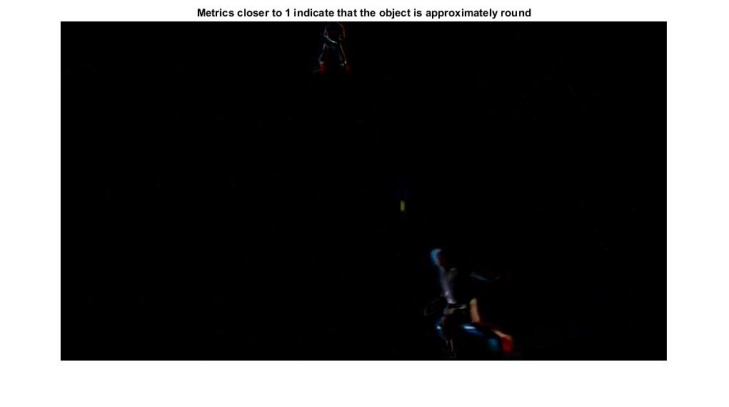
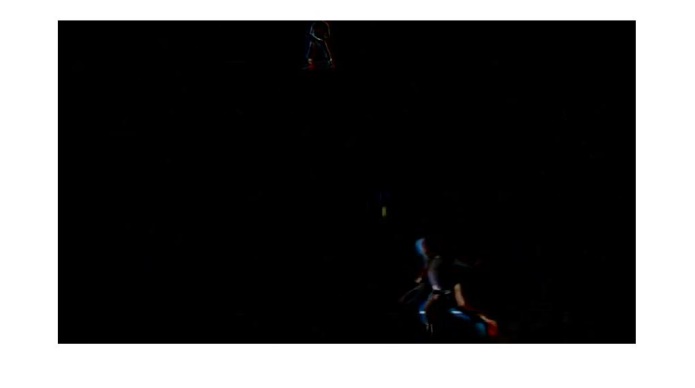
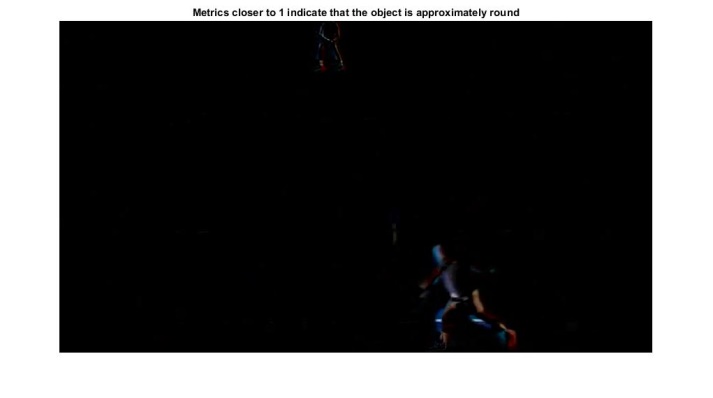
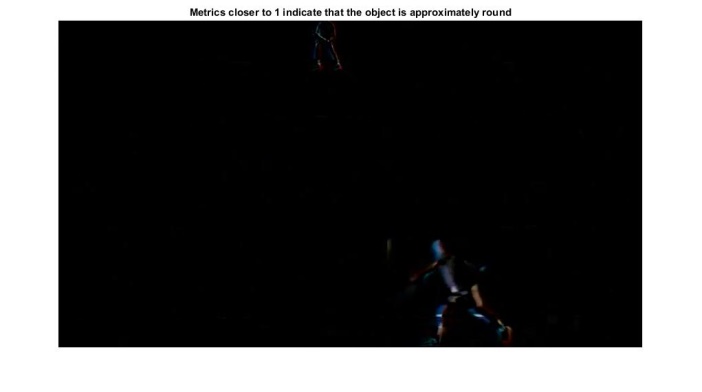
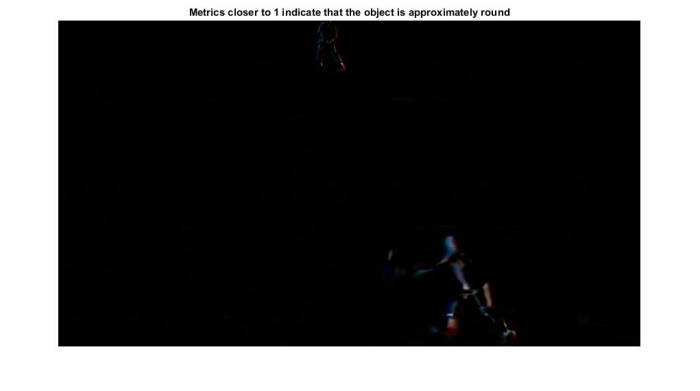
There are some cropped images as given below.







* 1. Image Differencing

In this image difference is taken of first image with the next image. In above figures you can see that the player and ball are in motion so difference of first with second will have only the moved player pixels and moved ball pictures. As we know that the ball will move faster than the player so the parts of player while differencing will be cut. But we will get a separated ball pixels in that difference. A single throw of ball images are given below. 

The differencing of images is like a core part of detecting the ball because we have lots of objects in the ground like net peoples and poles etc. And between in these objects it is like so difficult to find the ball. Because there are some objects which look like ball which make things difficult there may have some objects which looks like ball because while differencing some of player parts are cut because of their small movement and so they look like ball and it is also a big challenge for us to remove this ambiguity. So we use morphological operators.

* 1. RGB to Gray

RGB to gray converts RGB images to gray scale by eliminating the hue and saturation information while retaining the luminance. Because we need a gray scale image so that after words we can convert it into binary image so then we can apply the morphological operations.

RGB to gray converts RGB values to gray scale values by forming a weighted sum of the R, G, and B components:

**0.2989 \* R + 0.5870 \* G + 0.1140 \* B**

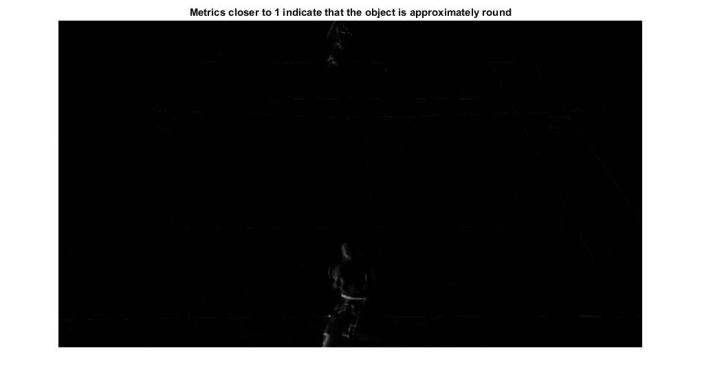
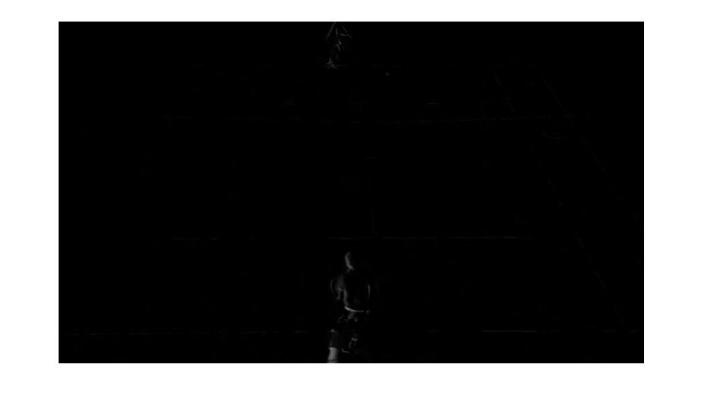
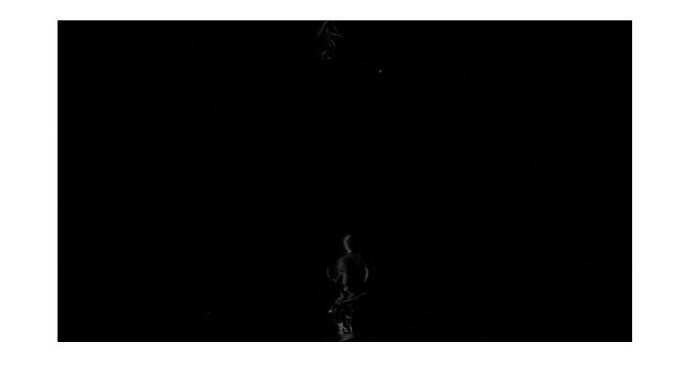
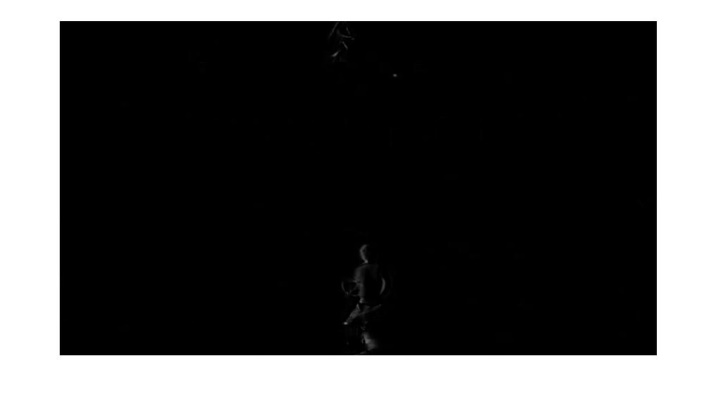
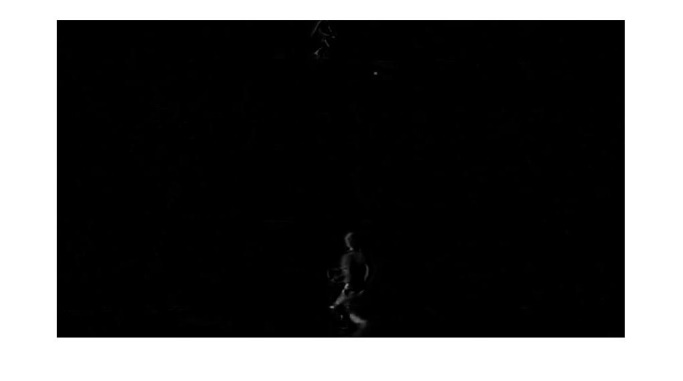
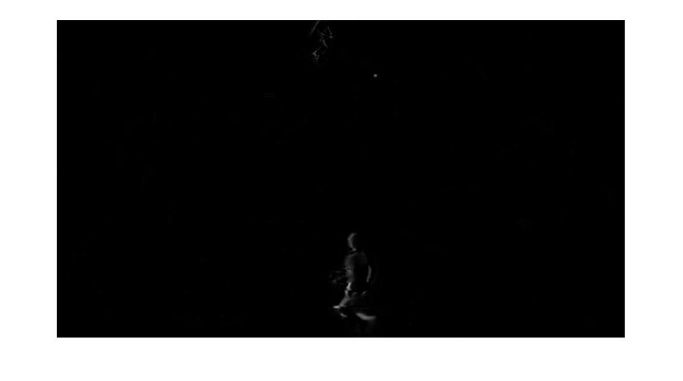
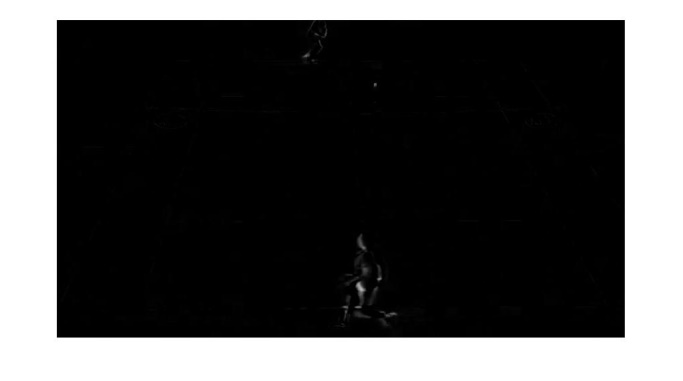
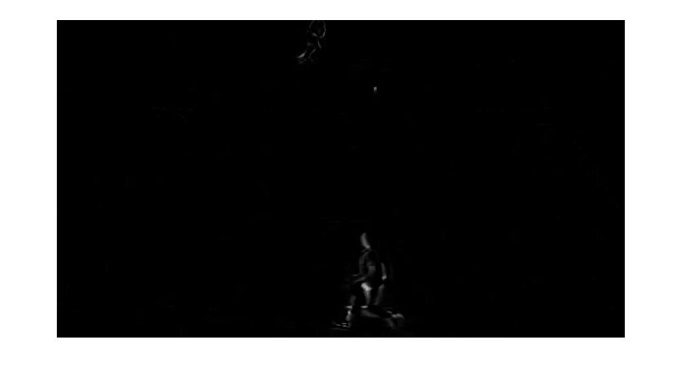
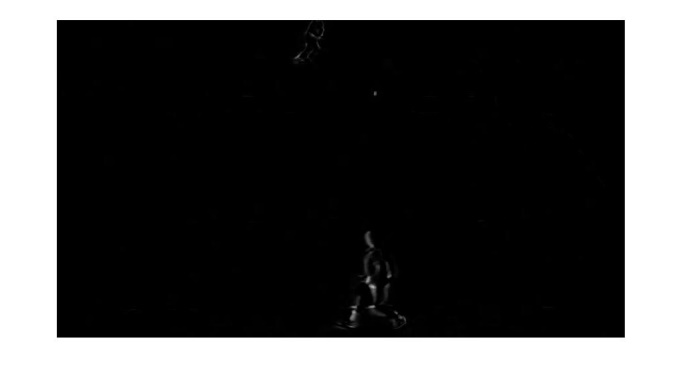
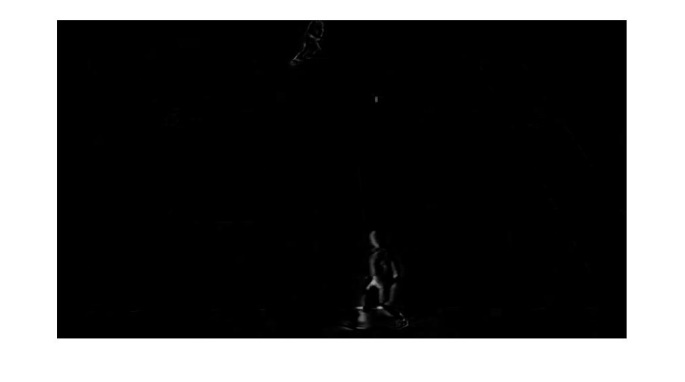
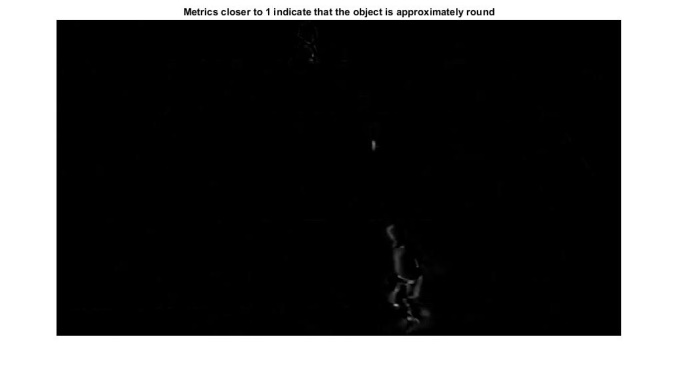
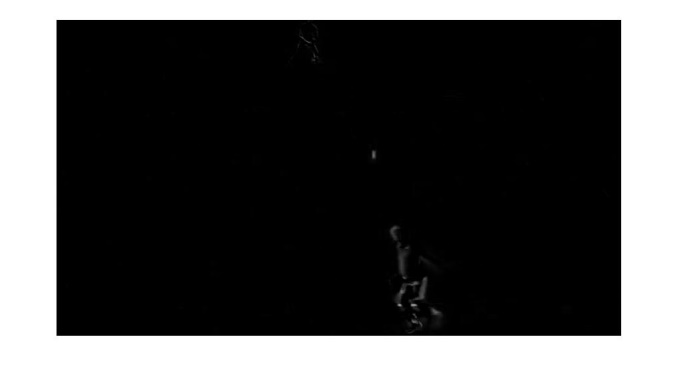
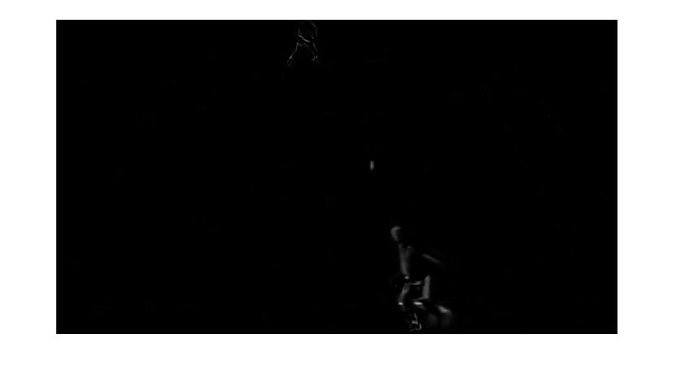
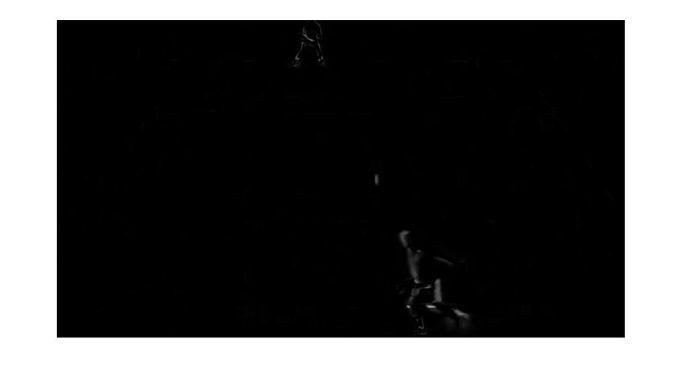
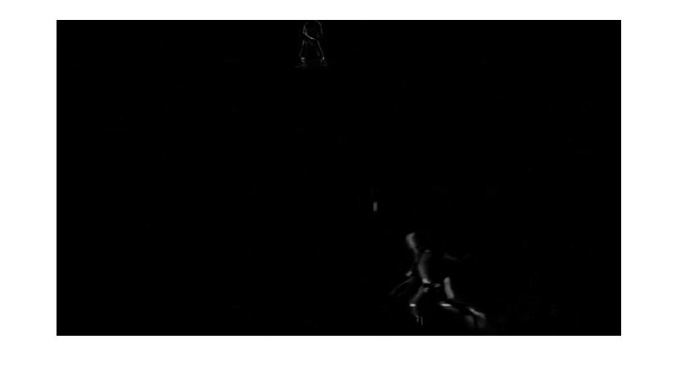
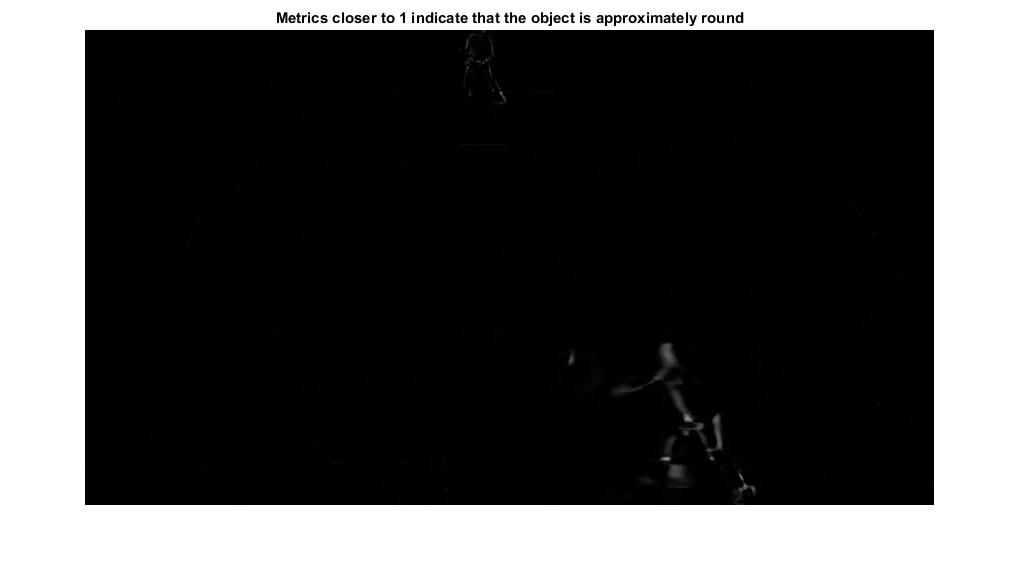
The coefficients used to calculate gray scale values in RGB to gray are identical to those used to calculate luminance (E'y) in

Rec.ITU-R BT.601-7 after rounding to 3 decimal places.

Rec.ITU-R BT.601-7 calculates E'y using the following formula:

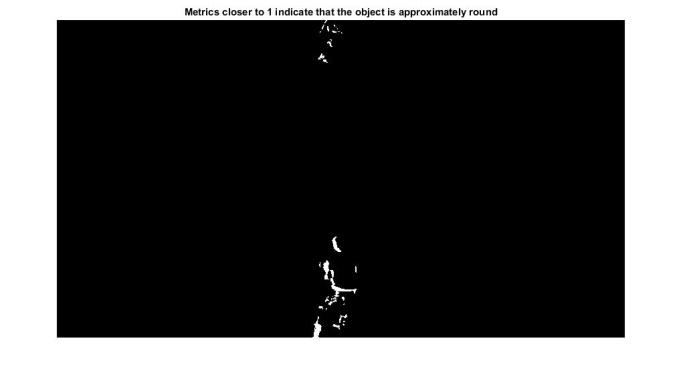
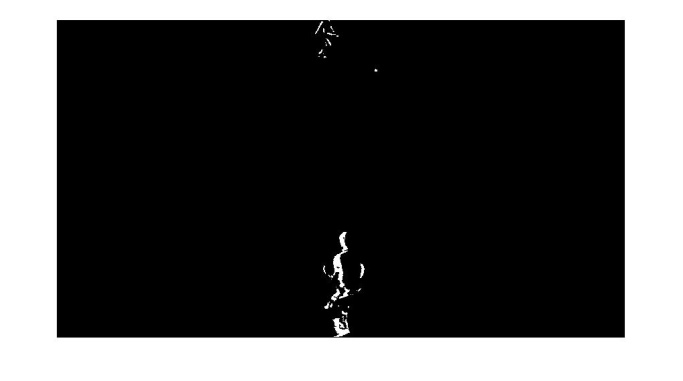
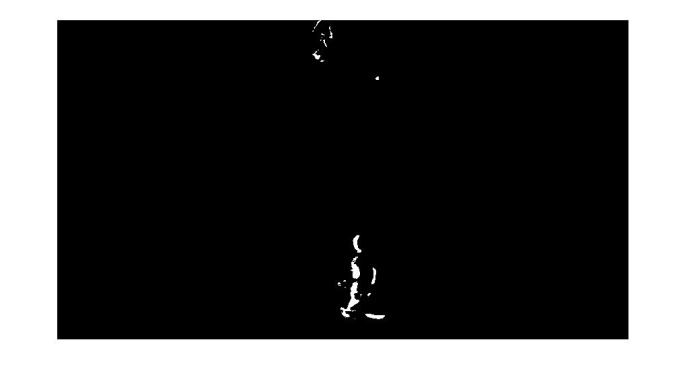
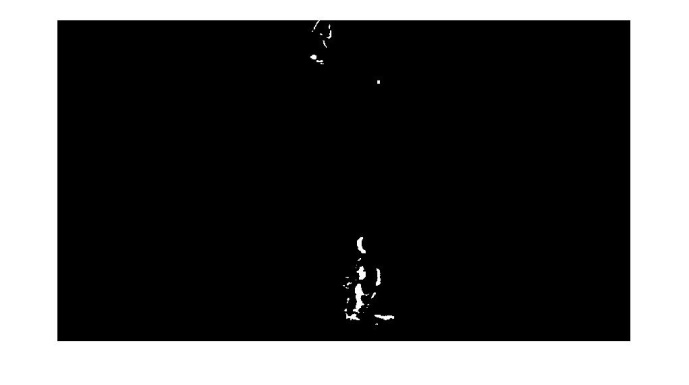
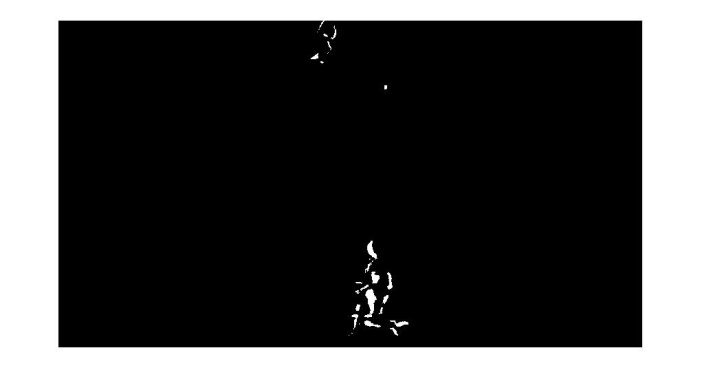
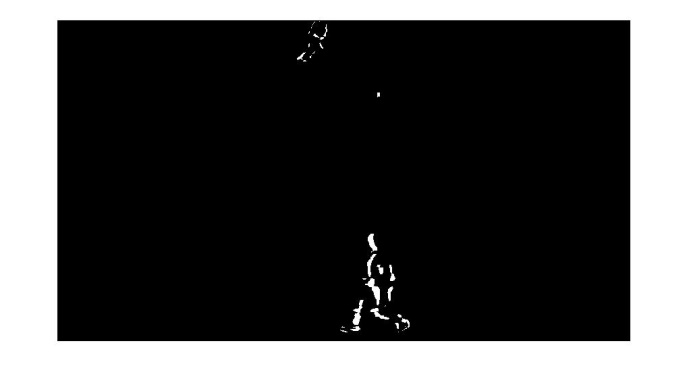
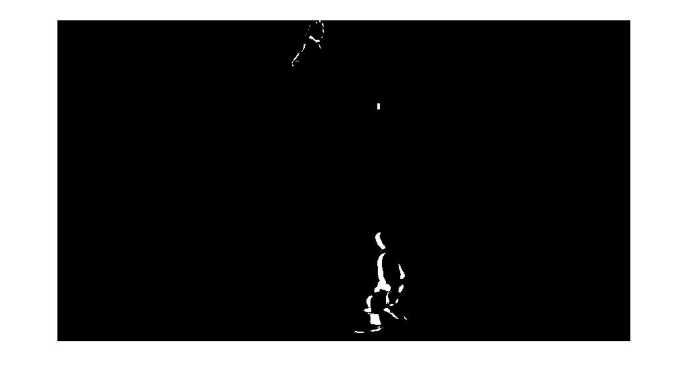
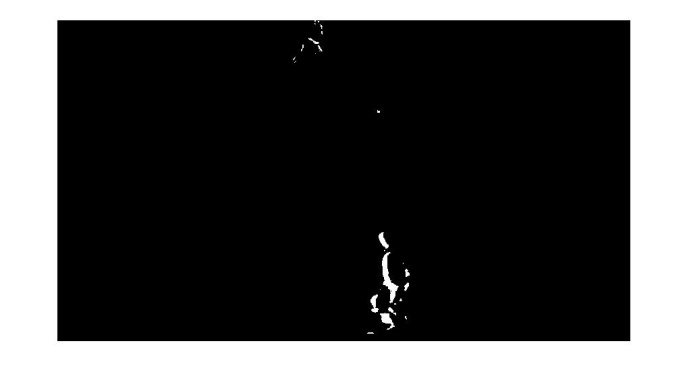
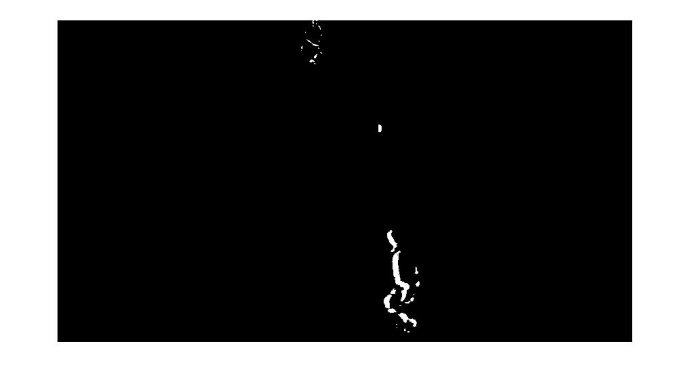
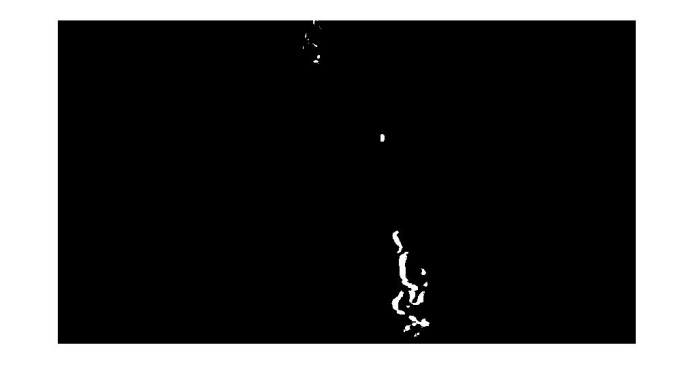
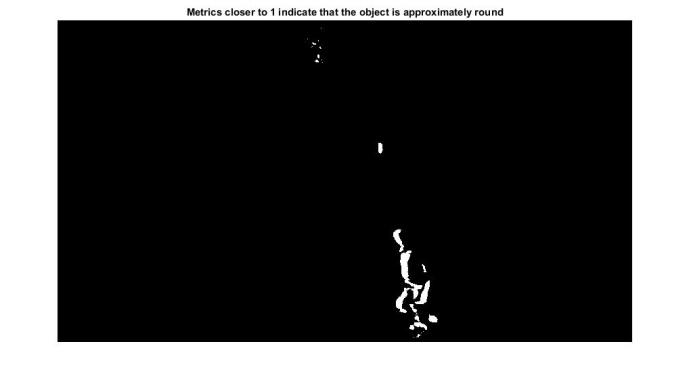
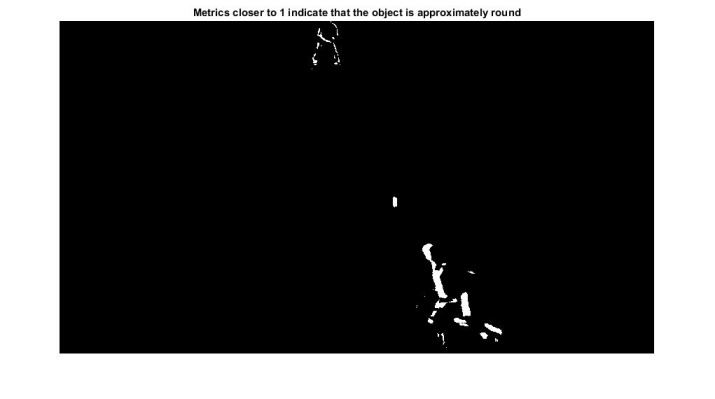
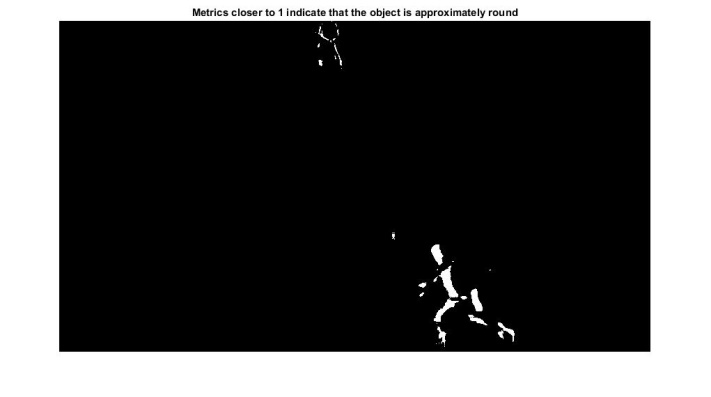
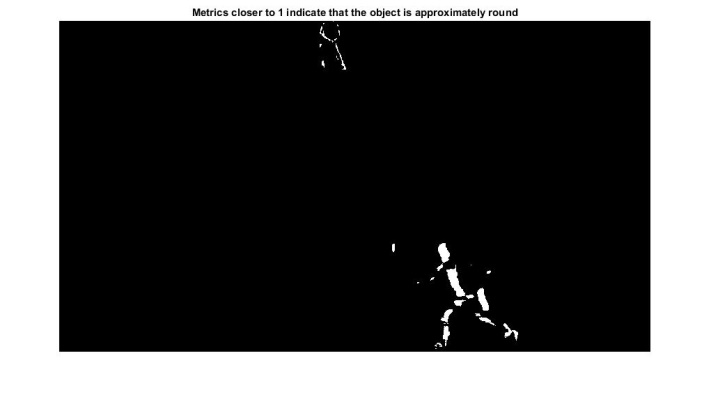
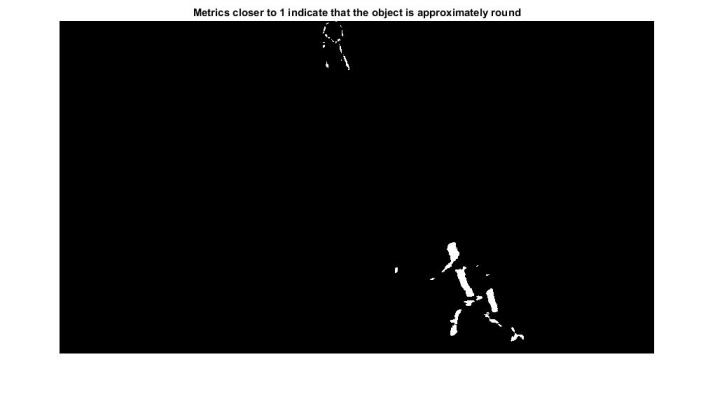
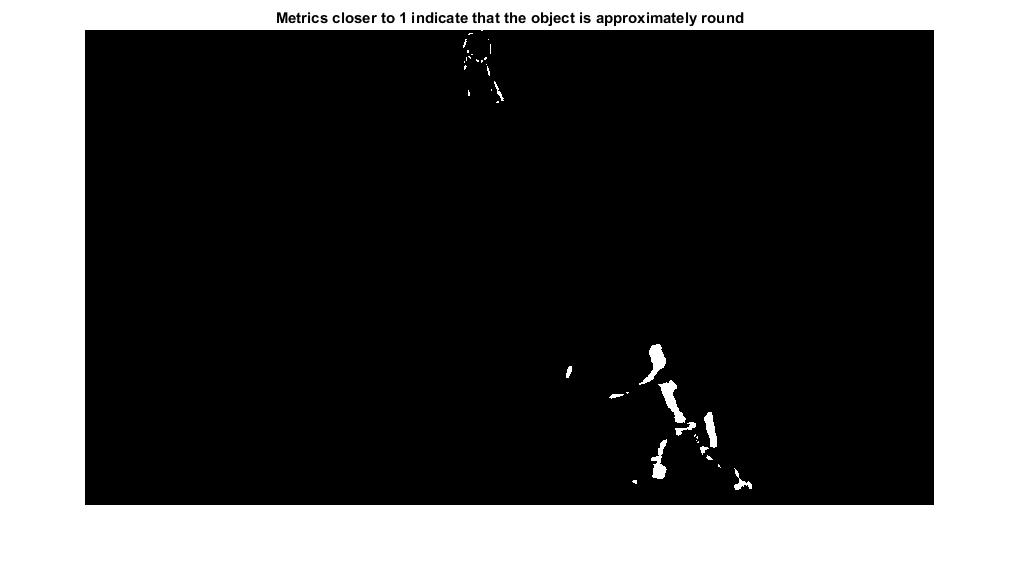
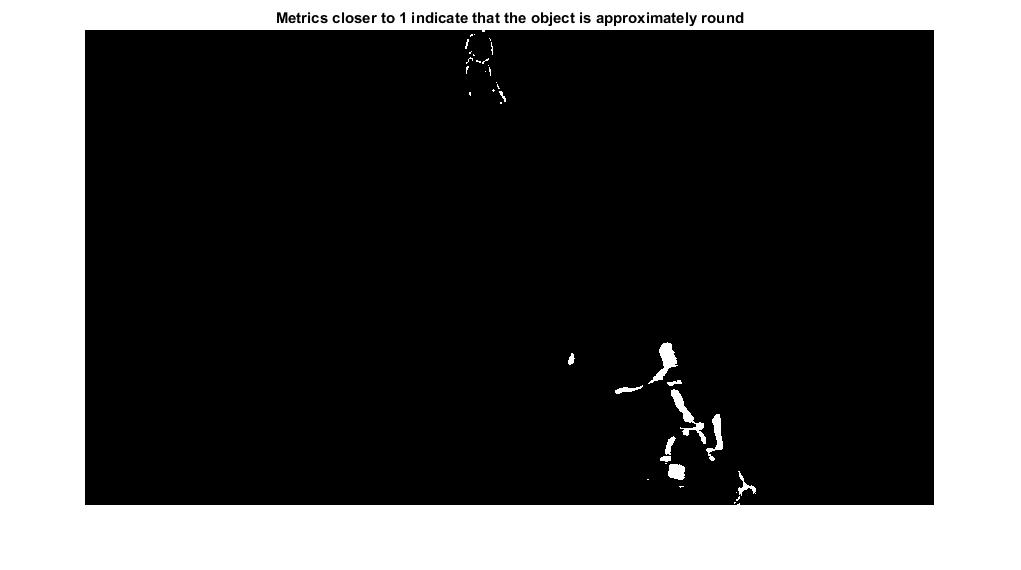
**0.299 \* R + 0.587 \* G + 0.114 \* B**

A single throw of ball images are given below.

******

* 1. Converting image into Binary

It binaries image I with a global threshold computed using Otsu's method, which chooses the threshold to minimize the intra class variance of the threshold black and white pixels. BW is the output binary image. We need a binaries image because morphological operators can’t be used in gray scale image or in RGB image because the morphological operations are performed pixel by pixel to accomplish the task. A single throw of ball images which are binaries are given below.



In these above figures we can see that each pixel is binaries as white and black. Instead of gray scale image it only converts the image into gray image. We converted it into binary because when morphological operations are applied it converts pixel value like black pixel value into white and white into black as using the structuring element given to it.

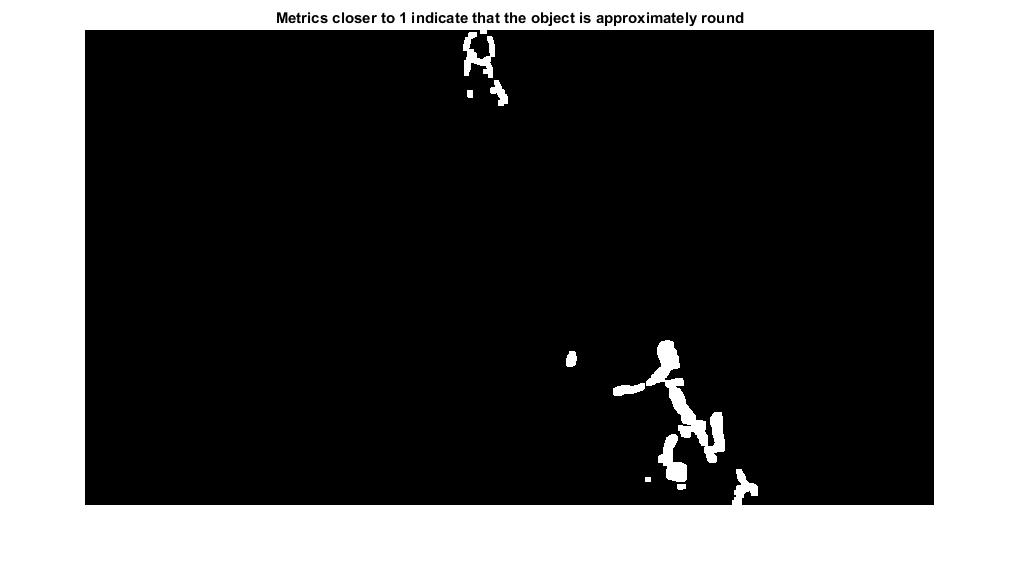
* 1. Dilation of image

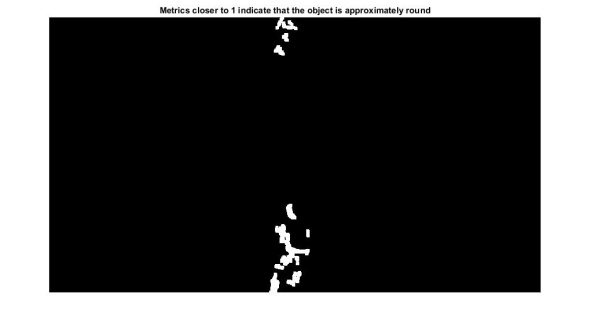
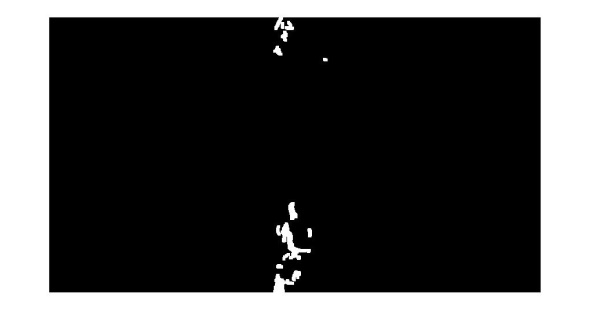
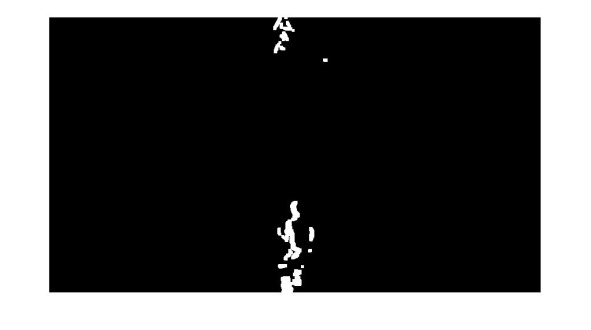
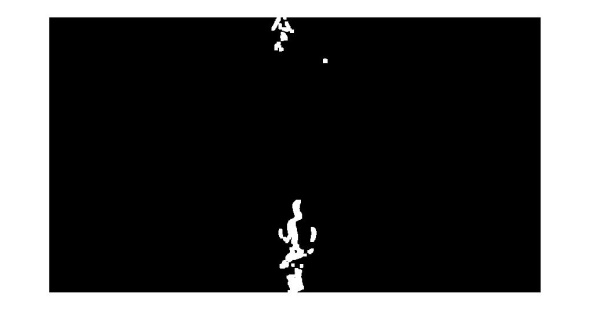
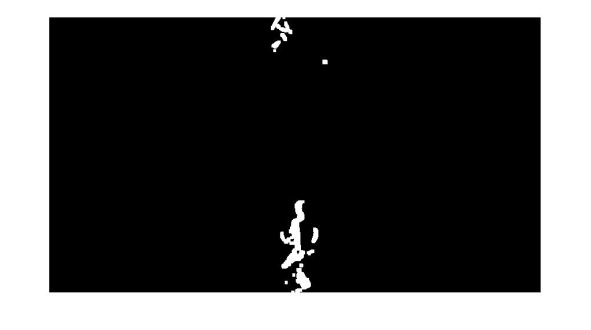
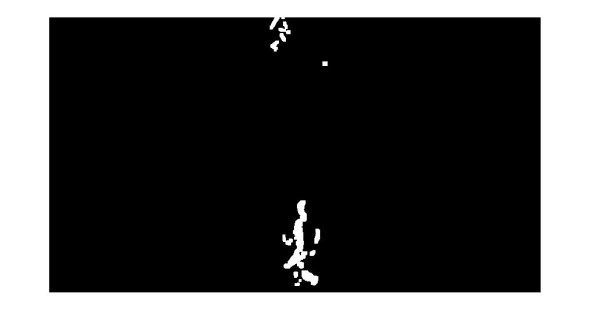
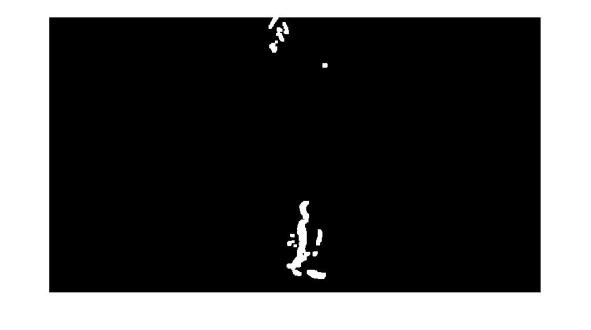
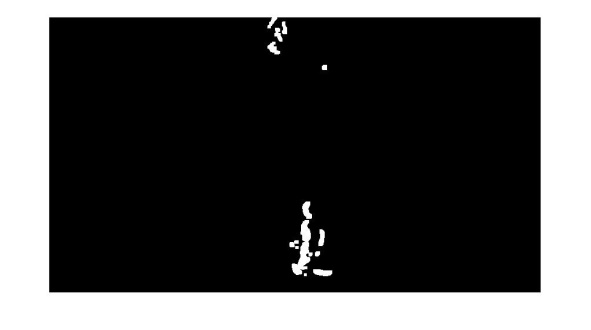
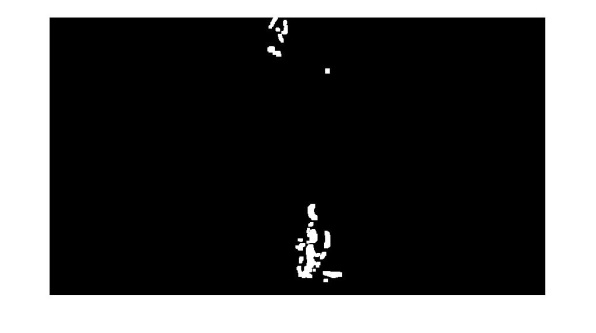
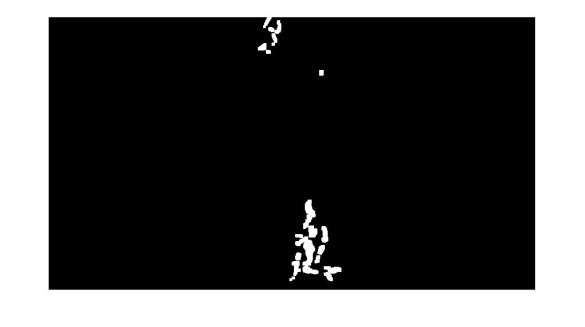
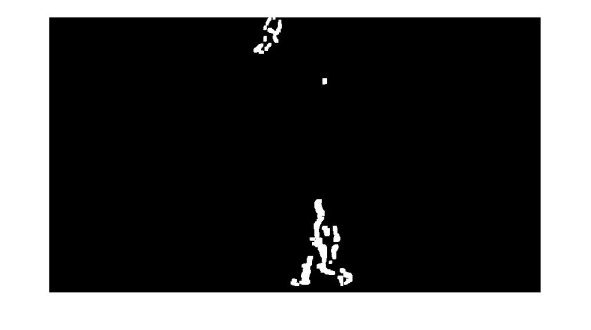
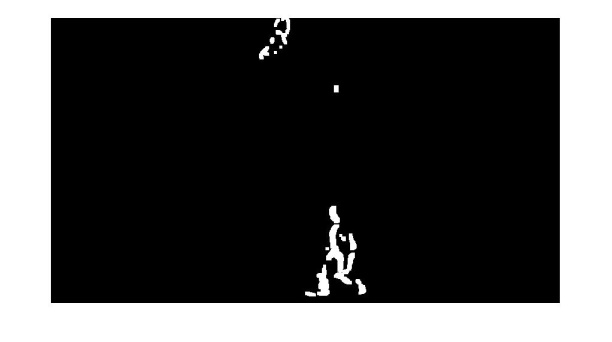
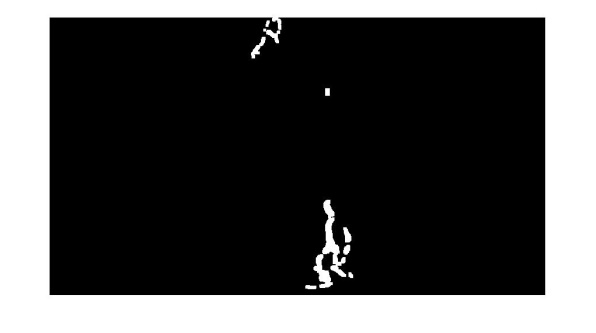
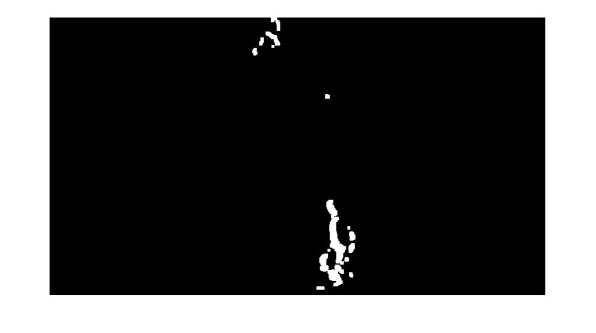
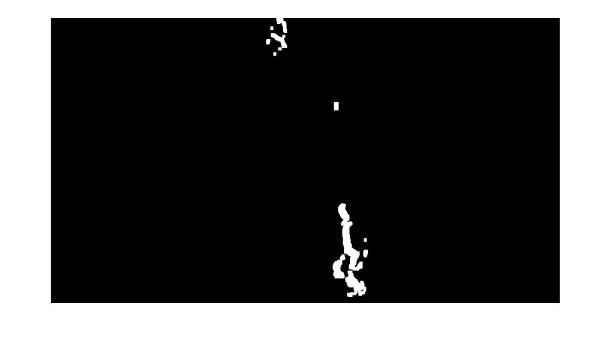
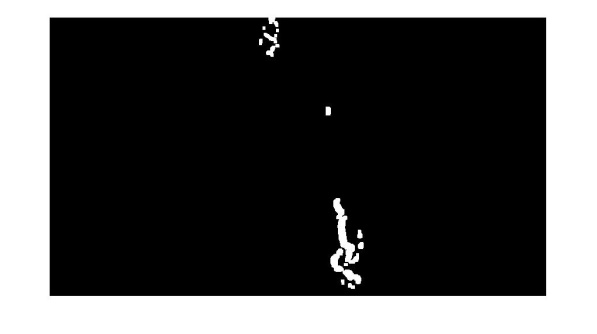
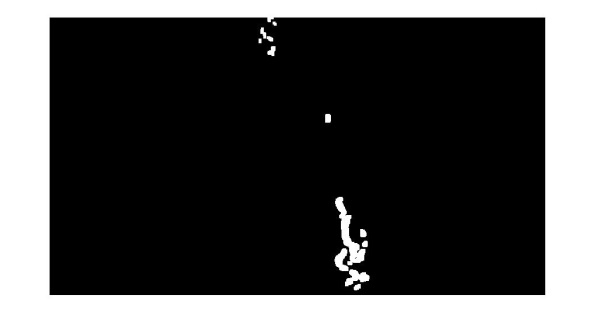
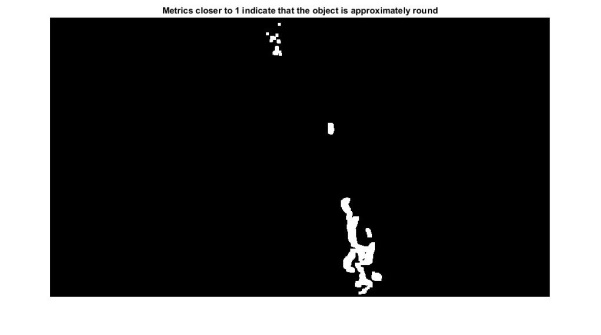
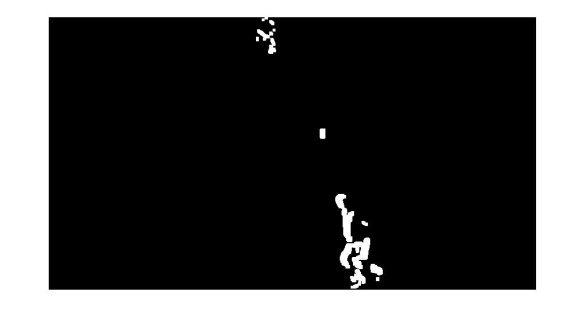
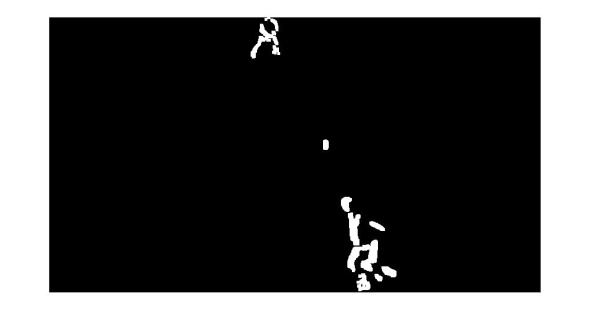
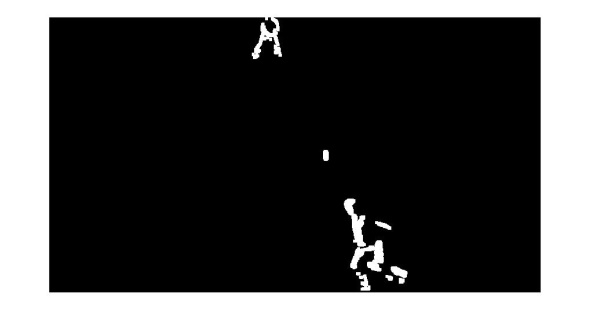
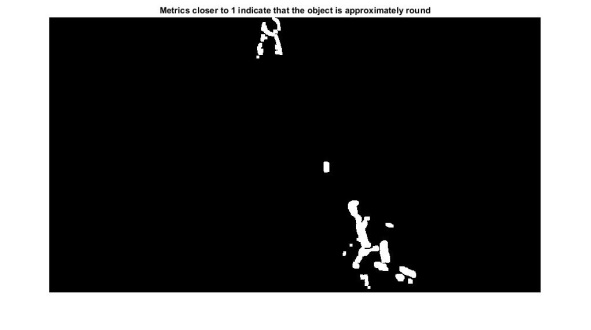
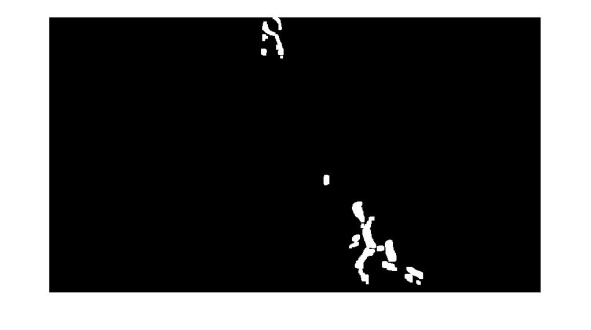
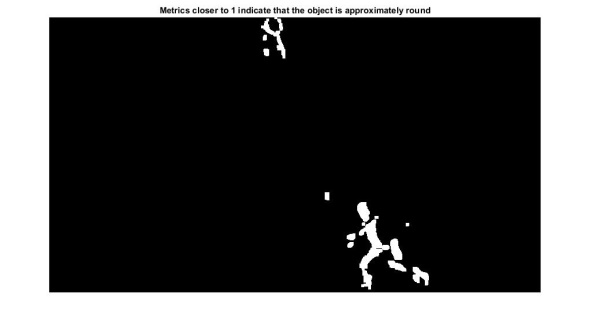
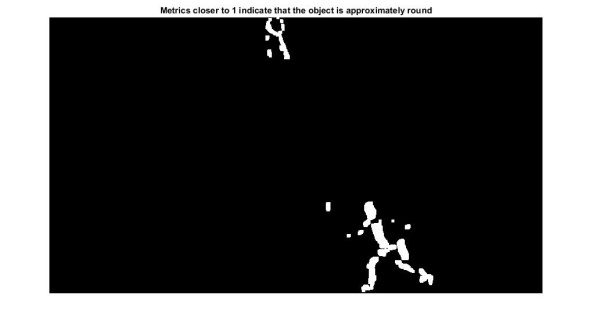
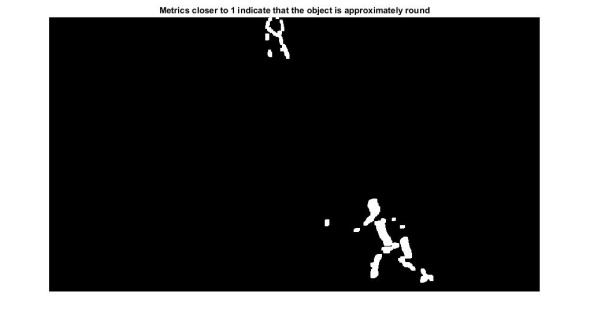
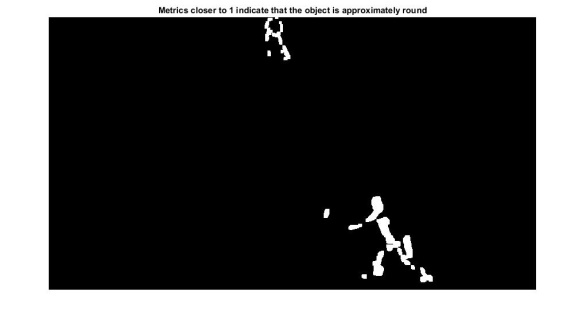
It is used to meet the broken pixels of the image like when we take a difference of first image with the second image some of player and ball pixels may have broken to repair those pixels we used image dilate which is described below.

Image dilate (IM, SE) dilates the gray scale, binary, or packed binary image IM, returning the dilated image, IM2. SE is a structuring element object, or array of structuring element objects, returned by the STREL or OFFSETSTREL functions. If IM is logical (binary), then the structuring element must be flat and image dilate performs binary dilation. Otherwise, it performs gray scale dilation. If SE is an array of structuring element objects, image dilate performs multiple dilations, using each structuring element in SE in succession.

The working of image dilate is it checks pixel by pixel and checks using structuring element. Like when it is checking a pixel value it checks its neighborhood value using structuring element and convert its value to 1 or 0 or white or black.

A single throw of ball dilated images are given below.





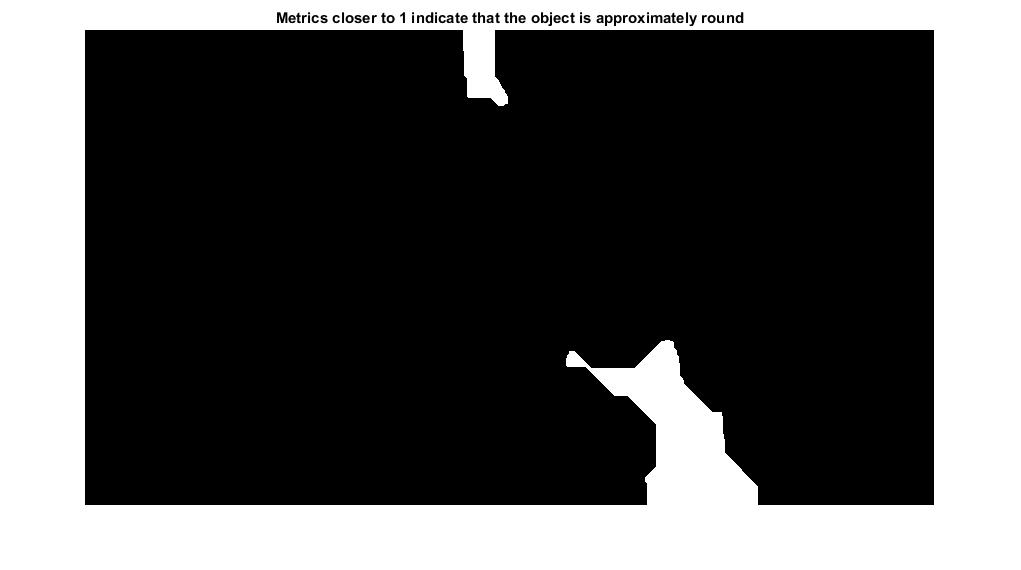
As in above images we can see that some of the broken parts of the player are been repaired.

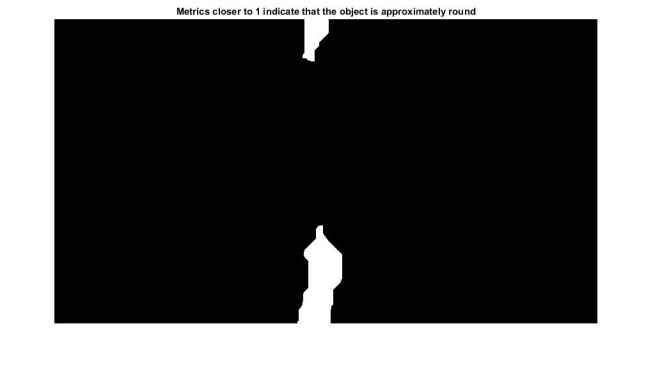
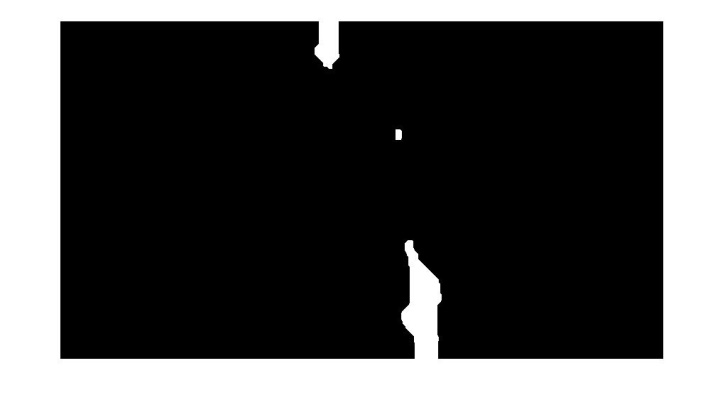
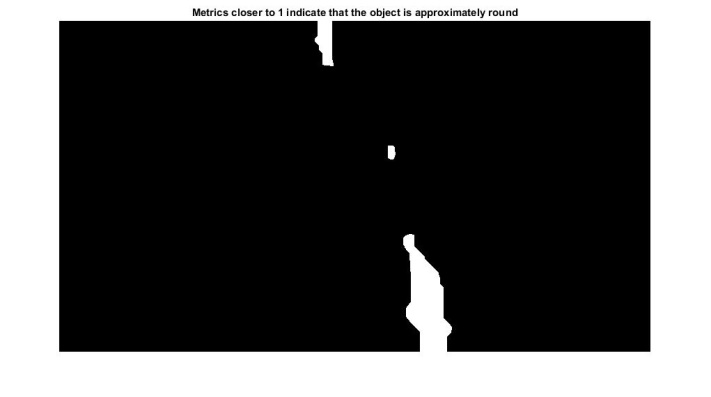
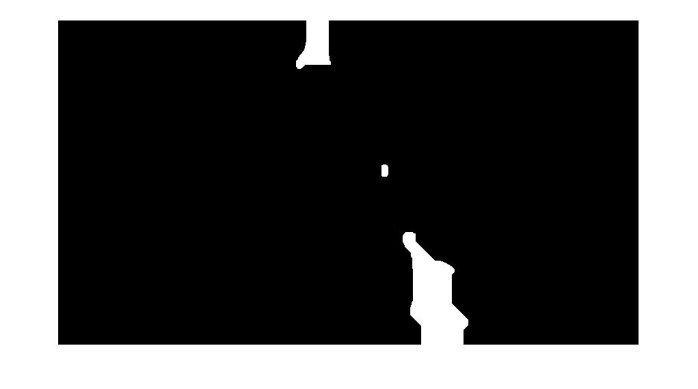
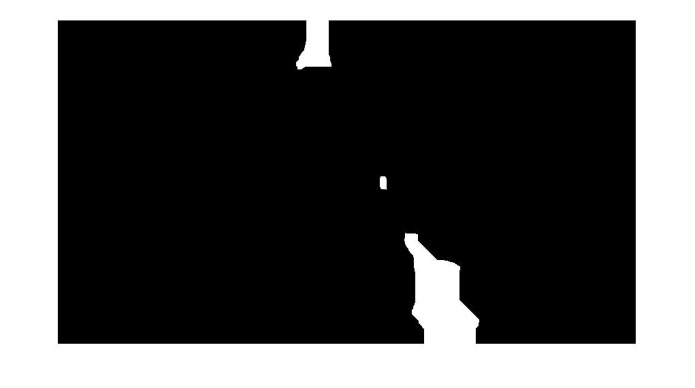
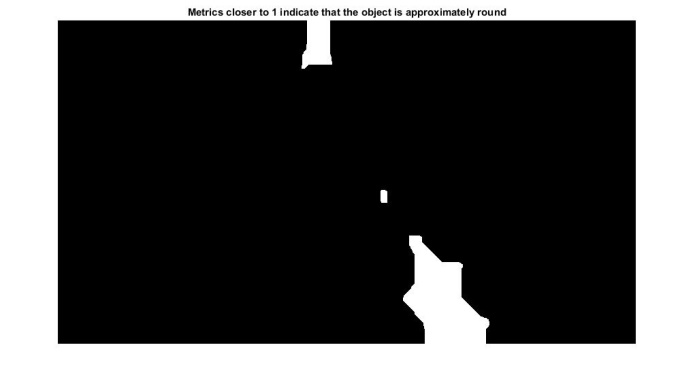
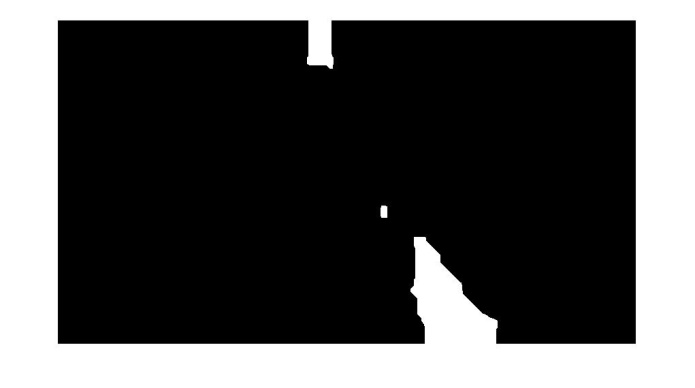
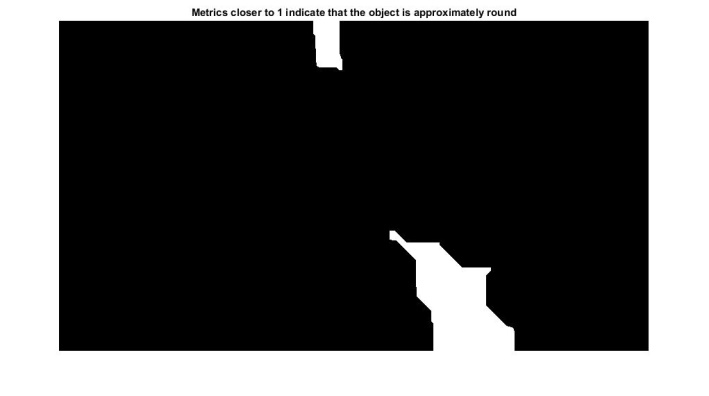
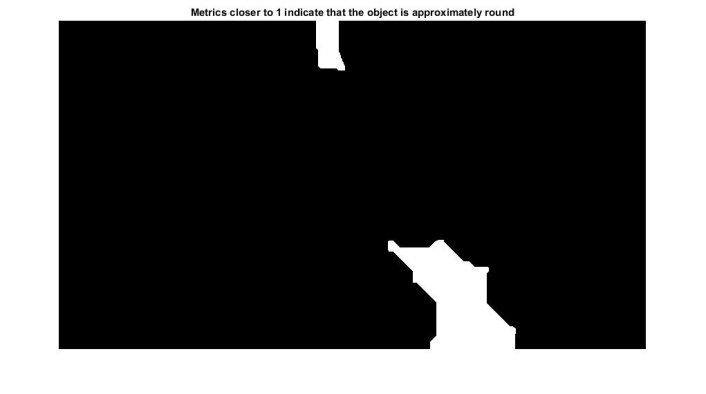
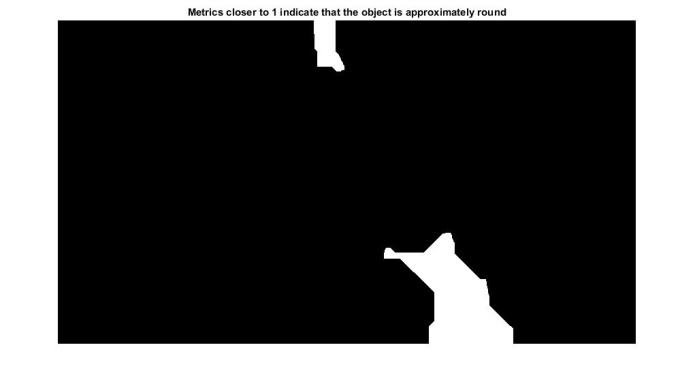
* 1. Closing the Image

The morphological close operation is a dilation followed by erosion, using the same structuring element for both operations. Image close (IM, SE) performs morphological closing on the Gray scale or binary image IM with the structuring element SE. SE must be a single structuring element object, as opposed to an array of objects.

We have used the closing morphological operator to close the player parts which are broken while image differencing. To accomplish that we have taken a disk structuring element to make cluster of those objects which are close to their neighborhood in image. While closing the image we have faced some problems like if a structuring element is too big and when ball is near to the player it will also merge the ball with the player. So for this problem we have used some other techniques which will be described after words.

A single throw of ball closing images are shown below.



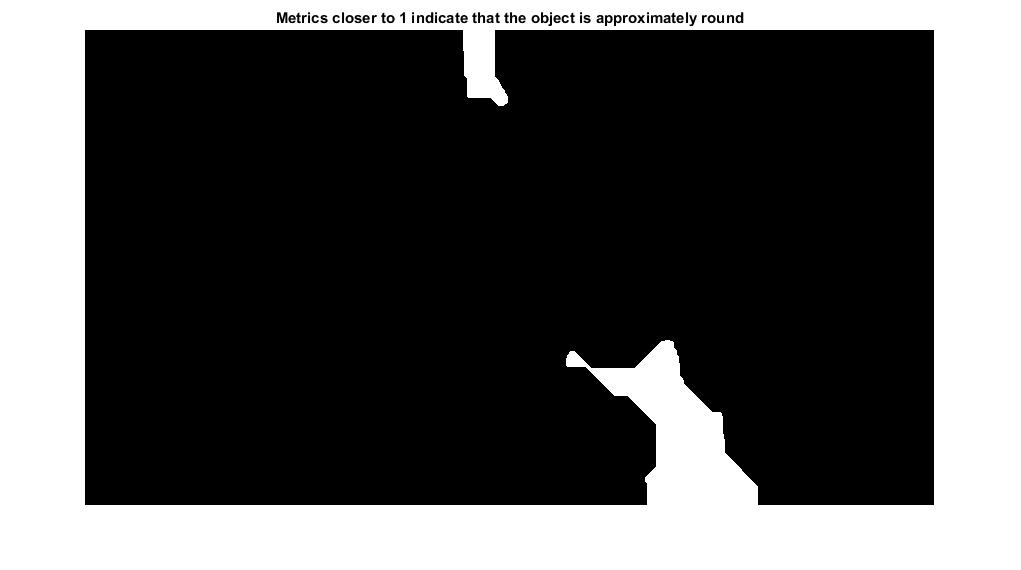


As we can see in the above figures that when the ball near to the player it becomes the part of it so to separate them we will use some techniques to separate them.

* 1. Binary area open

Binary area open (BW, P) removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image BW2. This operation is known as an area opening. The default connectivity is 8 for two dimensions, 26 for three dimensions, and CONNDEF (NDIMS (BW),'maximal') for higher dimensions.

We uses the binary area open to remove all objects which are having less than 30 pixel in our case because of noise or of any interference in the image there may have very small objects in it so to remove them we use binary area open. A figure is given below.

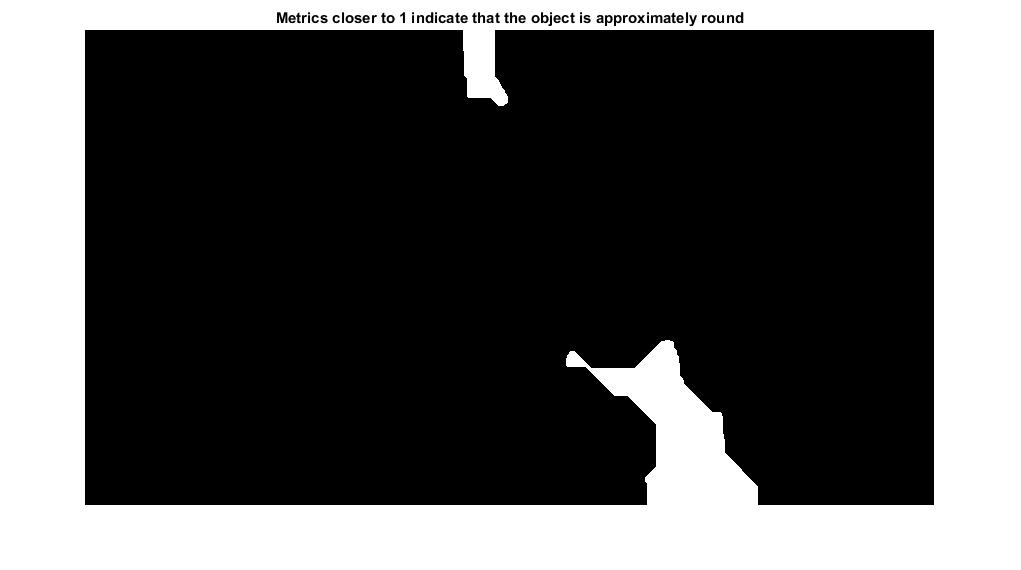


In the above image there are no such objects is present there may have some videos which may have those objects so it will remove those objects from it.

* 1. Fill holes in image

Image fill (BW1, LOCATIONS) performs a flood-fill operation on background pixels of the input binary image BW1, starting from the points specified in LOCATIONS. LOCATIONS can be a P-by-1 vector, in which case it contains the linear indices of the starting locations. LOCATIONS can also be a P-by-NDIMS (IM1) matrix, in which case each row contains the array indices of one of the starting locations.

In our case we will fill holes if there are holes in objects while applying these operations it will fill those holes and return a complete object without holes in it. A figure is given below.

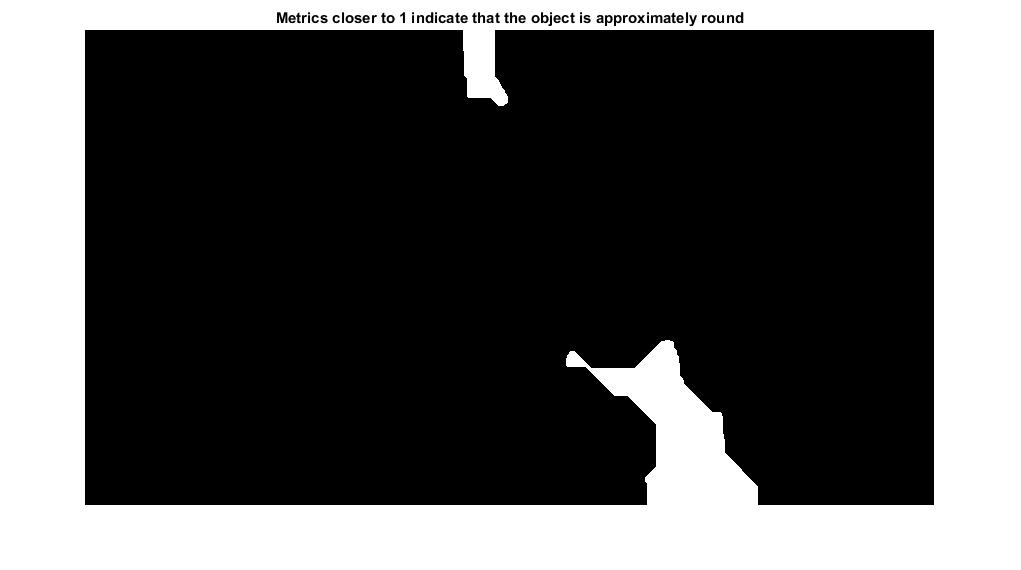


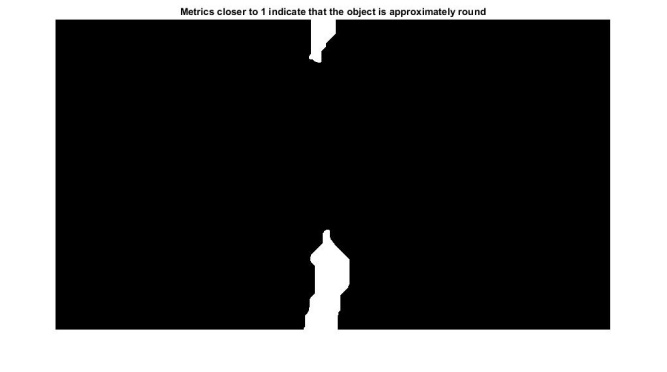
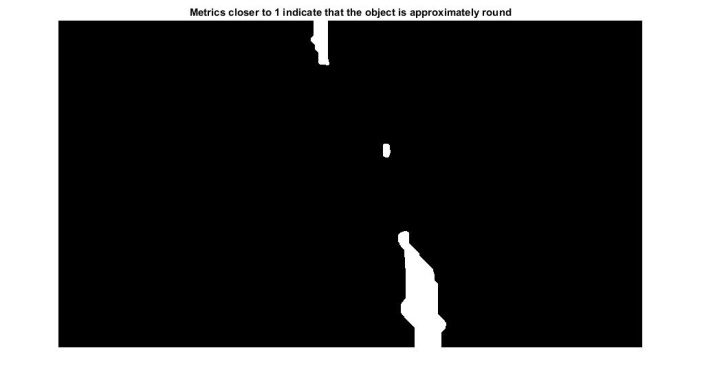
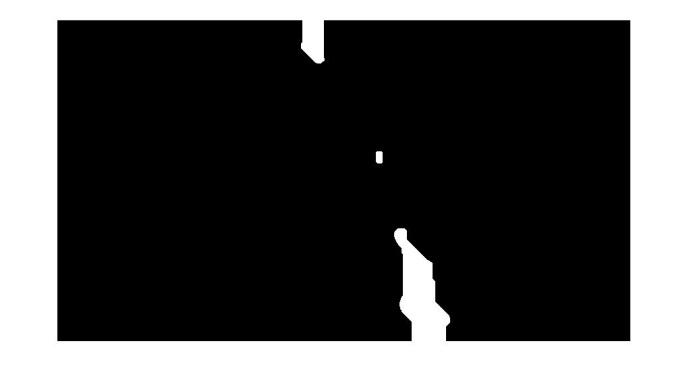
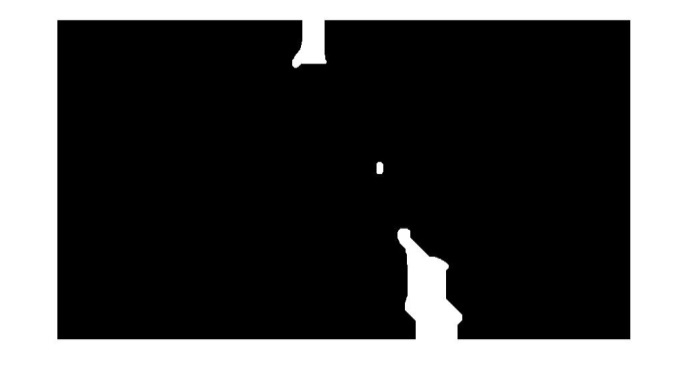
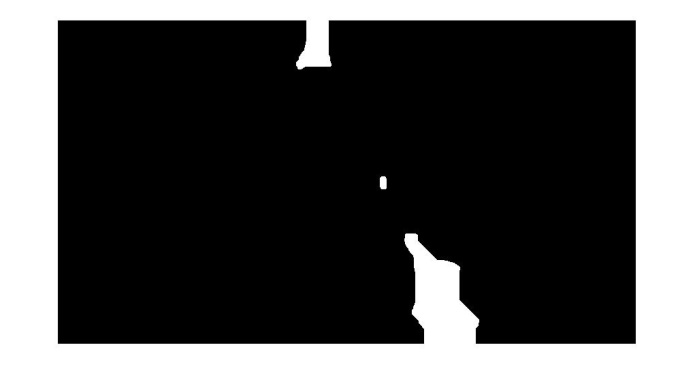
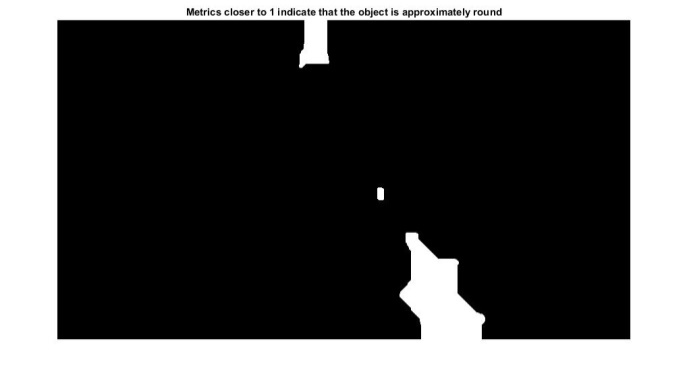
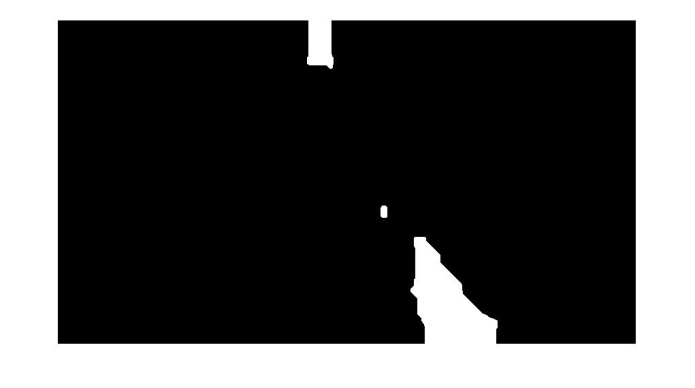
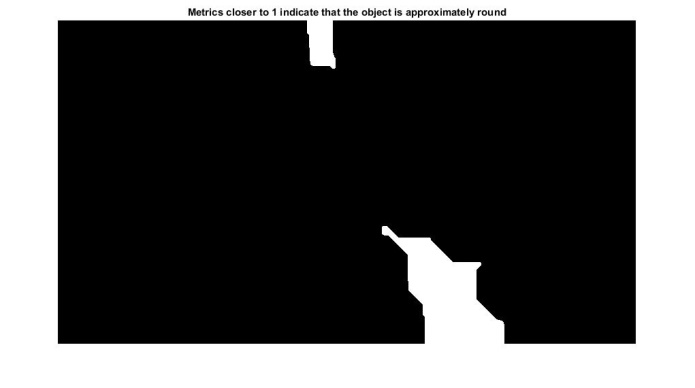
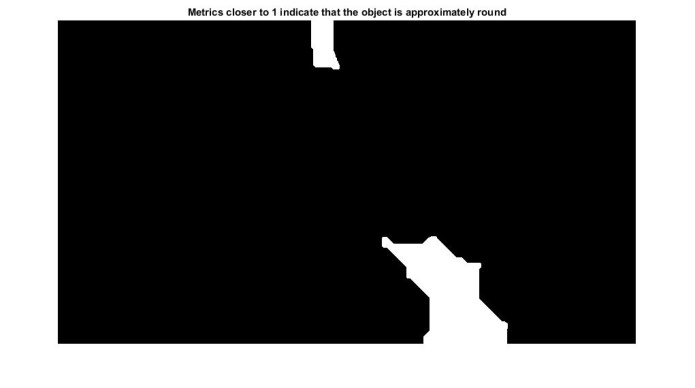
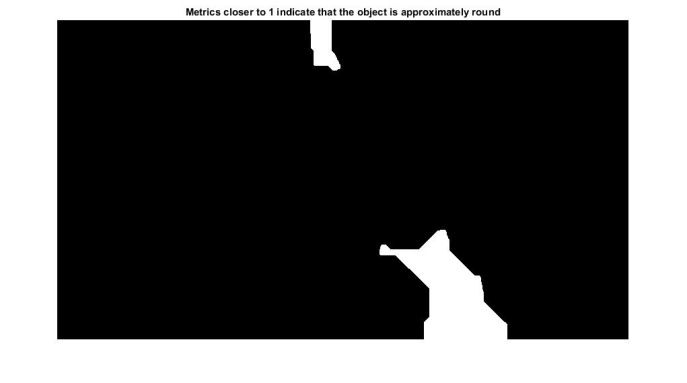
* 1. Opening Image

The morphological open operation is erosion followed by dilation, using the same structuring element for both operations. Image open (IM, SE) performs morphological opening on the gray scale or binary image IM with the structuring element SE. SE must be a single structuring element object, as opposed to an array of objects.

We used this procedure as we want to separate those objects which are been combined using image close so now we want to separate them. Because when ball came near to player body the image close function thinks that it’s a part of its body so to separate the ball from player’s body we used image open. The working of image open is that it separates those objects which are far from the neighborhood so as like ball is when far from the body of player but added as a part of player body because of image close so it then separates it.

A single throw of ball opening images are given below.





* 1. Applying Circle Equation

First of all we will remove those objects which have more than 180 pixels so to do that the area which is calculated by the circle equation is used and from which we eliminate the objects. Then the from the circle equation result it gives us objects shape that how much this object is near to circle because ball is a fast moving object so when player hits the ball and because of low quality and because of less frame rate it is difficult to find the ball. Because its shape will just look like an ellipse but after applying all these techniques and converting it in to binary it shapes changes so to measure the shape that how much it is near to a circle we used the following equation.

**obtain (X,Y) boundary coordinates corresponding to label 'k'**

**boundary = B{k};**

**compute a simple estimate of the object's perimeter**

**delta\_sq = diff (boundary). ^2;**

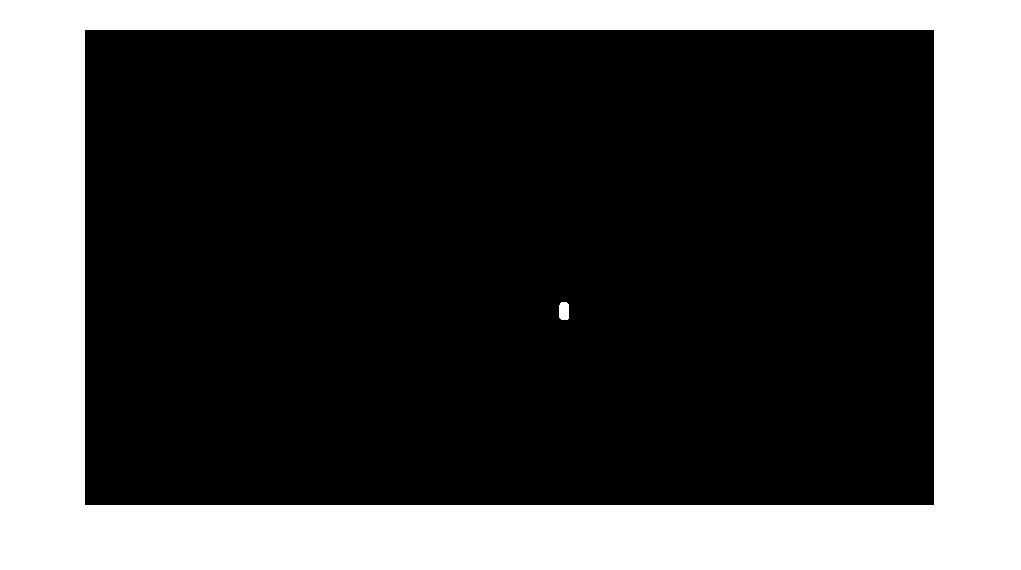
**Perimeter = sum (sqrt (sum (delta\_sq,2)));**

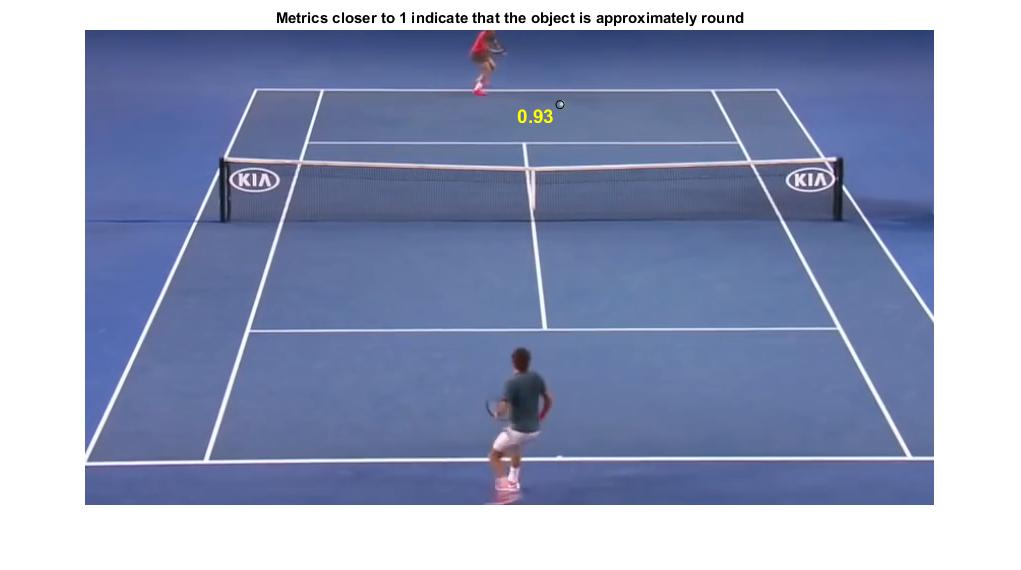
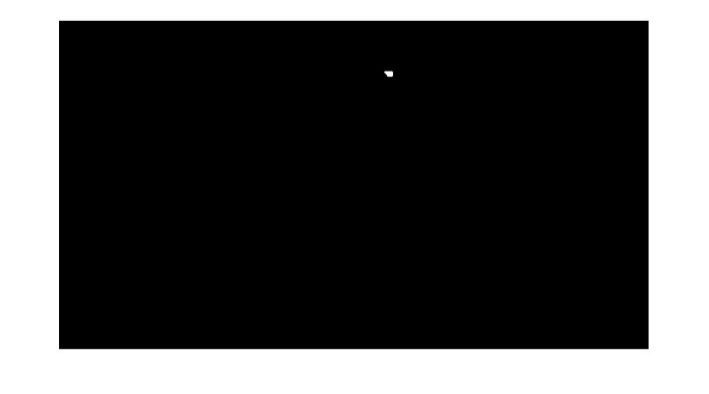
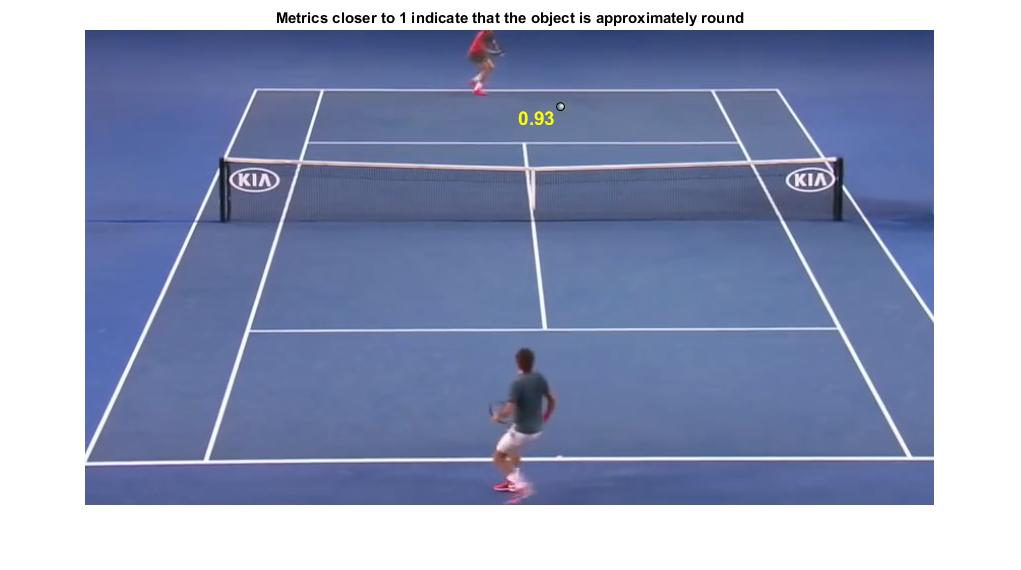
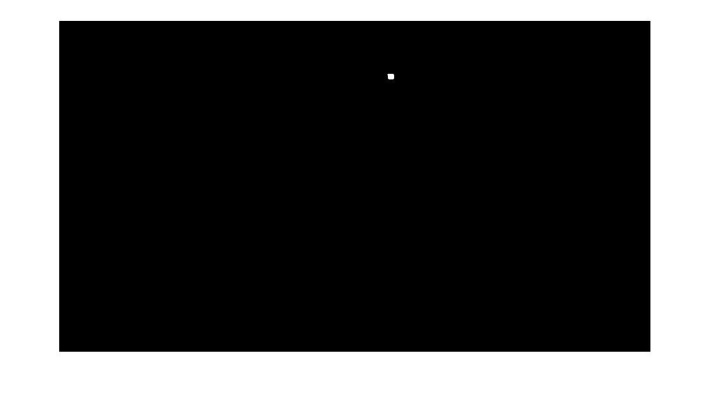
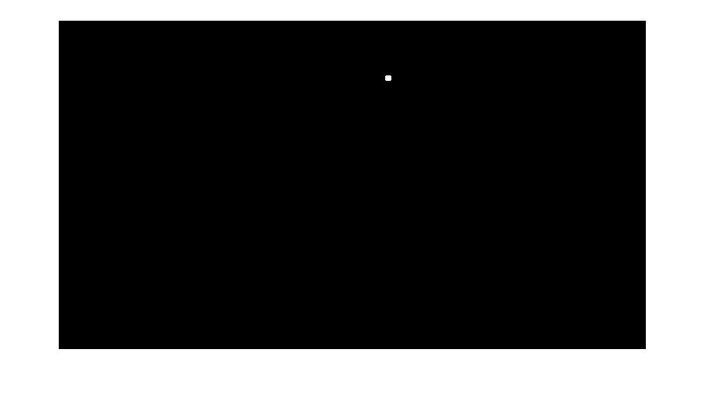
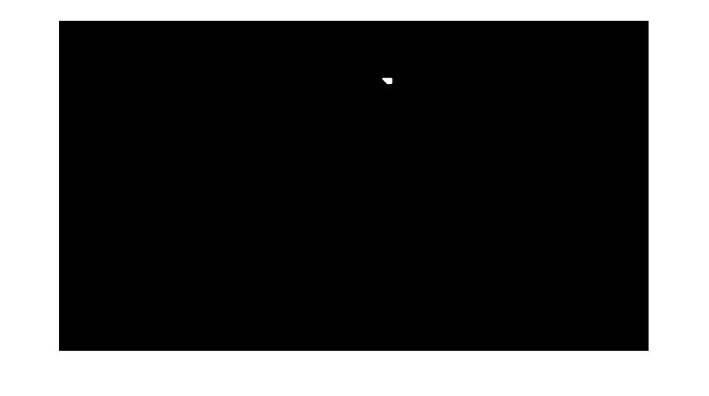
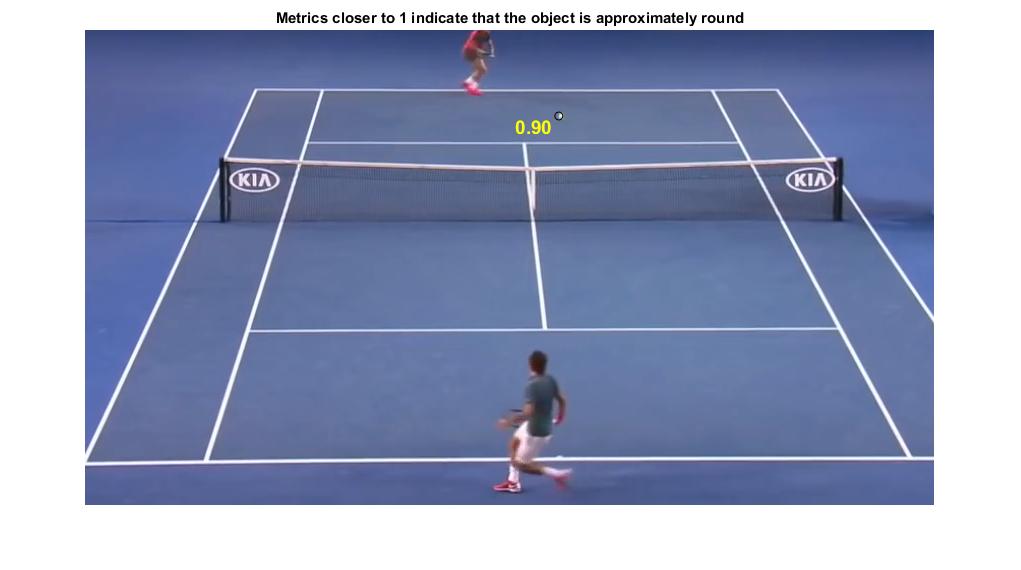
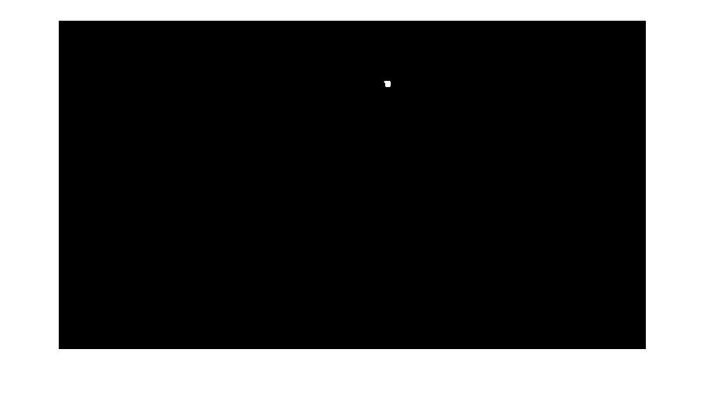
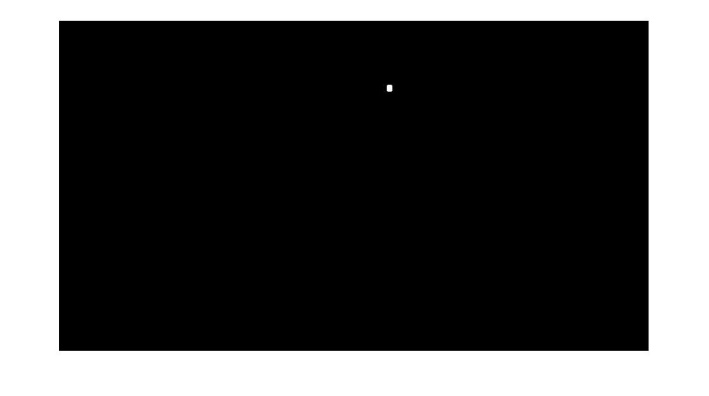
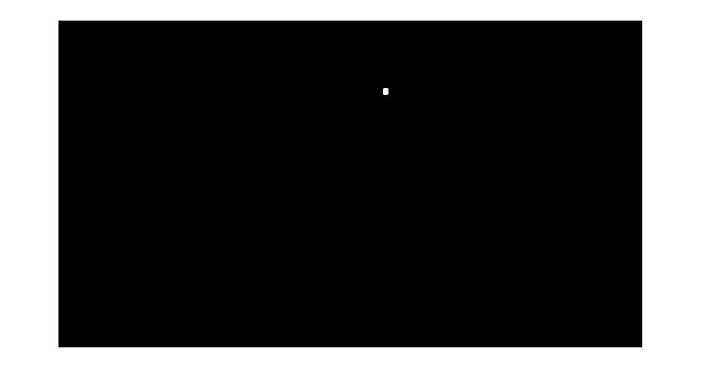
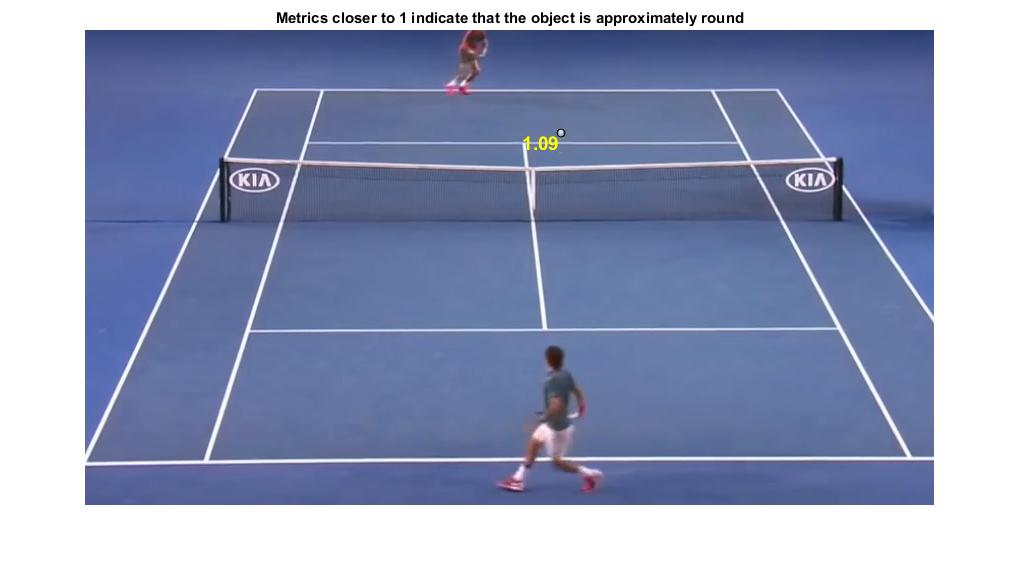
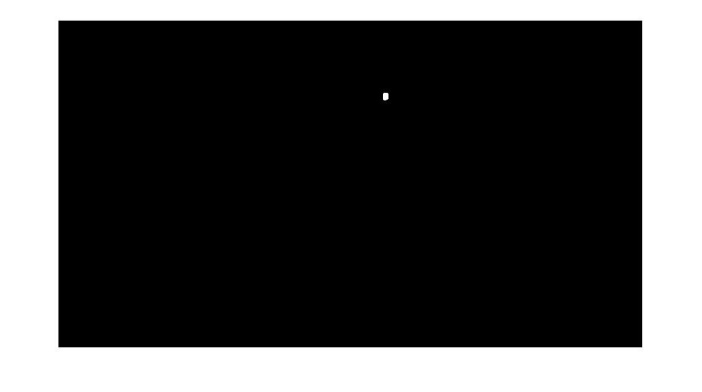
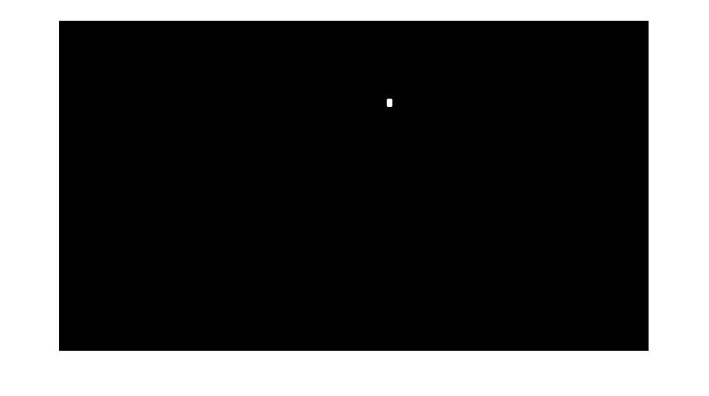
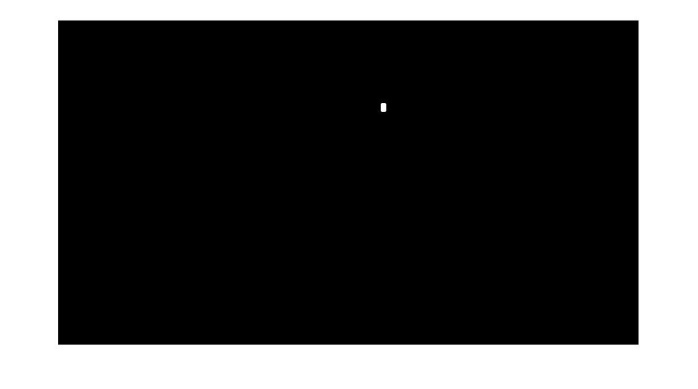
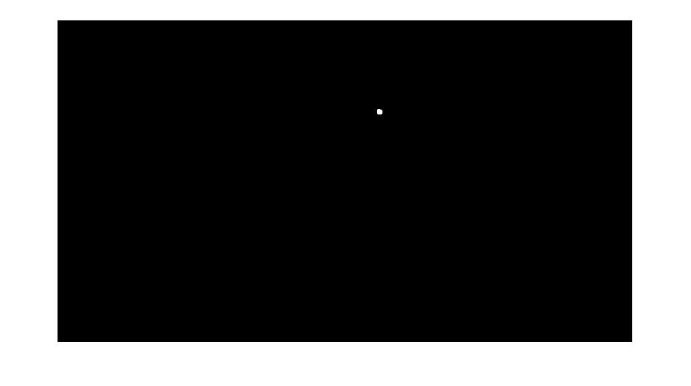
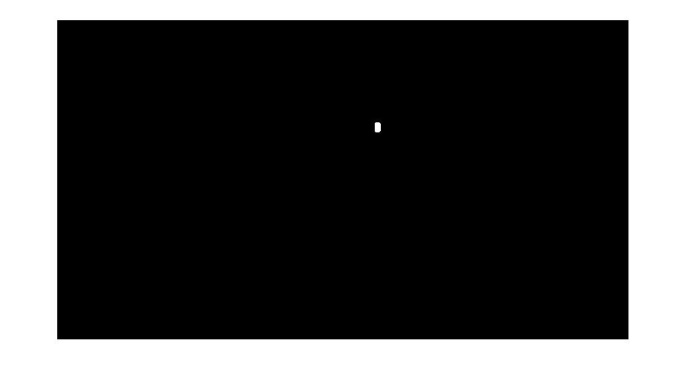
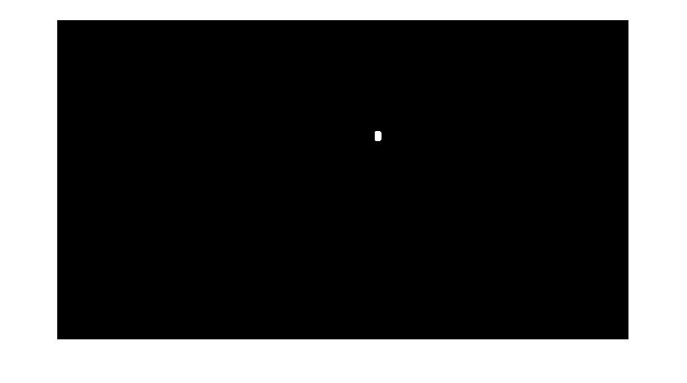
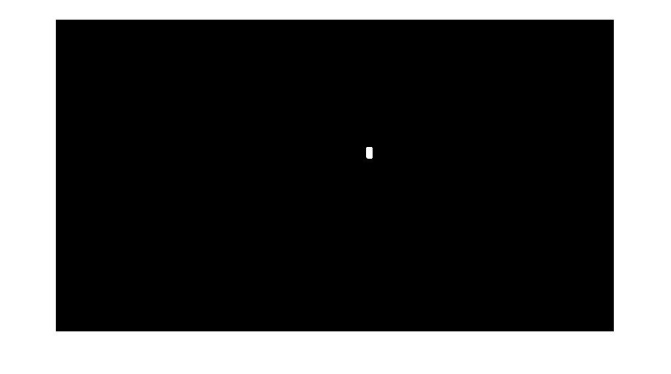
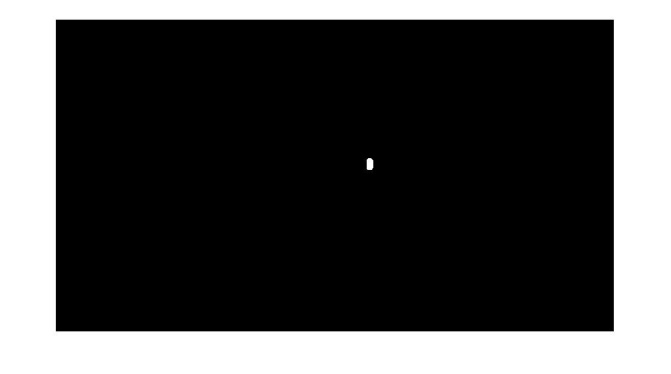
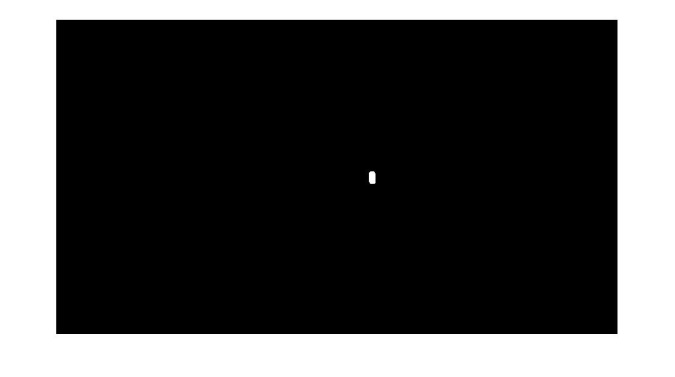
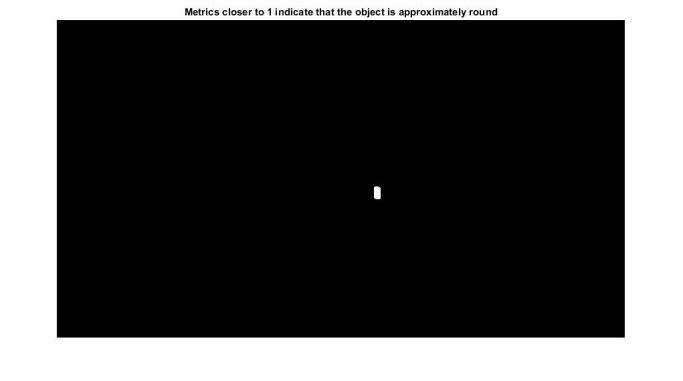
**obtain the area calculation corresponding to label 'k'**

**area = stats (k).Area;**

**compute the roundness metric**

**metric = 4\*pi\*area/perimeter^2;**

After applying the equation and condition the following result will be shown. 



* 1. Kalman Filter

**Kalman filtering**, also known as **linear quadratic estimation** (**LQE**), is an algorithm that uses a series of measurements observed over time

The Kalman filter has numerous applications in technology. A common application is for guidance, navigation, and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filters also are one of the main topics in the field of robotic motion planning and control, and they are sometimes included in [trajectory optimization](https://en.wikipedia.org/wiki/Trajectory_optimization). The Kalman filter also works for modeling the central nervous system's control of movement. Due to the time delay between issuing motor commands and receiving sensory feedback, use of the Kalman filter supports a realistic model for making estimates of the current state of the motor system and issuing updated commands.



In the diagram shown above shows that all the calculations which are done to calculate the next state of the object is based on the calculation of Kalman gain. That is the original value is given by the data input that is measure value. And the previous value calculated by the Kalman is given to the equation to calculate the next state of the object. And Kalman gain is calculated through the error is the measurement and error in the estimate and will be used in the equation to calculate next state.

The above diagram will be explained below in detail.

* + 1. Kalman Filter Calculating Kalman Gain



KG

|  |
| --- |
| 1 |
| 0.9 |
| 0.8 |
| 0.7 |
| 0.6 |
| 0.5 |
| 0.4 |
| 0.3 |
| 0.2 |
| 0.1 |
| 0 |

Accurate

Estimates Are Inaccurate

Measurements are accurate

Kalman Gain =KG

Error in Measurement = EMEA

Error in Estimate = EEST

Measurements Are Inaccurate

Estimates are accurate

KG= 0 ≤ KG ≥ 1

In above equation of Kalman Gain shows that if the error in measurement is greater than the Kalman will select its measured value for the next state instead of original measured value, and if the measured value has very small error then it gives a large weight to the original measured value and selects the measured value as the next state of the object.

* + 1. Estimating Next State



Kalman Gain =KG

Error in Measurement = EMEA

Error in Estimate = EEST

Current Estimate = EST t

Previous Estimate = EST t-1

KG = 0 ≤ KG ≥ 1 equation (1)

EST t = EST t-1 + KG [MEA - EST t-1 ] equation (2)

EEST = = > EEST = [1 – KG] () equation (3)

In above equation the equation (1) is used to calculate the Kalman Gain. That is used to give the weight to the value to original measurement and measurement calculated by the Kalman. Then in equation (2) the next value estimate is calculated by the equation using previous estimate and the value which is calculated by the Kalman Gain weight is given to original measurement and if measurement has small error then it will be added to the previous state and if it has large error then Kalman measured calculation will be added to the previous state.

* + 1. Kalman Filter Equations



As shown in above diagram that the above boxes show the equations that are used in Kalman filter to calculate the next state.

The initial state box contains two matrixes X0 is the state matrix and the P0 is the covariance matrix.

The previous state box contains two matrixes Xk-1 is the previous state matrix and the Pk-1 is the previous covariance matrix.

The new state box contains two matrixes Xkp is the predicted state matrix and the Pkp is the predicted covariance matrix. Xkp is calculated using previous state and its motion model to calculate the next state. Pkp is calculated using previous covariance.

The Update box contains equation one is used to calculate Kalman gain and second uses the previous state and the Kalman gain and the measured state and calculated state and Kalman gain assign weight to the values and the next state will be calculated.

Measured input box contains an equation which only converts the measurement value to a matrix so then it can be used in the equation.

The current box contains the equation where values are been assigned to calculate the covariance matrix and the state matrix.

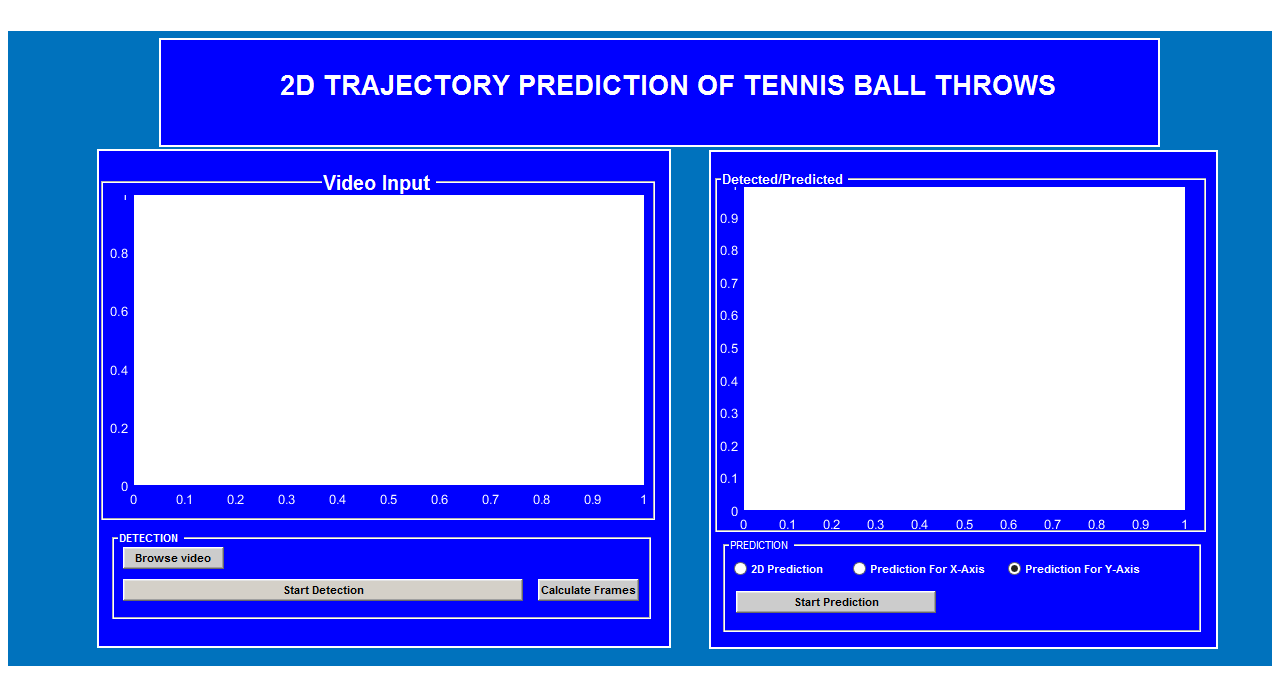
The output box contains the matrixes of state matrix and covariance matrix.

The above process will be done for every input value given by the data input.

CHAPTER 4: TRANSITION

# Screen Snapshots

**App Main Screen**

****

CHAPTER 5:

Future Work and Conclusion

# Future Work / Conclusion

## Future Work

In the above scenario of 2D Trajectory Prediction of Tennis Ball Throws. We have done detection using frame differencing and using equation to find objects which are close to circle and done prediction using Kalman filter.

In future we want to apply neural network techniques in our project. In neural networks the change will be that we have predicted the next state in this using Kalman. But in neural networks we will feed the trajectories data of thrown ball from one position to the algorithm and train it until then, we will throw the ball from that position and in starting position of ball the trained algorithm will check its position and using the train data it will show the predicted trajectory. So we have known that using Kalman we will only predict next state but using neural networks we can predict the next trajectory.

## Conclusion

In this proposed model, an improved ball tracking for BTV, besides varying camera motion causes as the presence of many balls-like objects and the small size of the tennis ball. It is not only increases the accuracy in identifying the ball, but also improves the accuracy in determining the ball projection position. In addition, it detects the ball landing frames and landing positions based on the accurate ball tracking, the result shows, model that this is able to precisely classify the ball tracking in Australian Open match between Nadal and Federer. After detection the prediction results are gathered using datasets and applying Kalman filter on the dataset and 2D prediction results are shown with very accurate results. As discussed in future work we will work on this model to apply neural networks.

# References

1. Object Detection and Tracking based on Trajectory in Broadcast Tennis Video by

M. Archana and M. Kalaisevi Geetha / Procedia Computer Science 58 ( 2015 ) 225 – 232.

1. Ball Tracking for Tennis Video Annotation by F. Yan (B )·W. Christmas·J. Kittler Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK
2. Automated Ball Tracking in Tennis Videos by Tayeba Qazi1, Prerana Mukherjee2, Siddharth Srivastava3, Brejesh Lall4, Nathi Ram Chauhan5 1, 5Department of Mechanical and Automation Engineering Indira Gandhi Delhi Technical University for Women Delhi, India 2, 3, 4Department of Electrical Engineering, Indian Institute of Technology, Delhi, India 1,5 qazitayeba,nnramchauhan@gmail.com, 2,3,4 {eez 13 8300, eez 127 506, brejesh} ee.iitd.ac.in
3. Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video by Xinguo Yu, Member, IEEE, Hon Wai Leong, Member, IEEE, Changsheng Xu, Senior Member, IEEE, and Qi Tian, Senior Member, IEEE
4. Players Tracking and Ball Detection for an Automatic Tennis Video Annotation by Kosit Teachabarikiti 1 , Thanarat H. Chalidabhongse 2 , Arit Thammano 1 1 Faculty of Information Technology, King Mongkut’s Institute of Technology Ladkrabang Bangkok, Thailand 10520 2 Faculty of Engineering, Chulalongkorn University Bangkok, Thailand 10330
5. A Trajectory-Based Ball Detection And Tracking Algorithm In Broadcast Tennis Video by

Xinguo Yua, Chern-Horng Simb, Jenny Ran Wangc, and Loong Fah Cheongb,

Electrical and Computer Eng, National University of Singapore, Singapore 117543 ‘Institute for Information Research, 2 I Heng Mui Keng Terrace, Singapore 119613 , E School of Computer Science & Eng, The University of New South Wale, Sydney 2052, Australia

## Additional References

1. https://www.youtube.com/watch?v=q7AiwWwiF\_k