



AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com



Tracking Colored Objects Using Kalman Filter

Dr. Anas Quteishat¹, Anwar Al-Mofleh², Anmar Abuhamdah³

¹Al-Balqaa Applied University, Faculty of Engineering Technology Computer Engineering Department

E-mail: Anas.Quteishat@fet.edu.jo

²Al-Balqaa Applied University, Faculty of Engineering Technology Electrical Engineering Department,

³Department Of Computer Science, College Of Computer Science And Engineering ,Taibah University, Madinah Munawwarah, KSA

ARTICLE INFO

Article history:

Received 25 January 2014

Received in revised form 8

March 2014

Accepted 10 March 2014

Available online 20 March 2014

Keywords:

Object tracking,
Image Processing
Kalman filter

ABSTRACT

Background: Tracking objects in a sequence of images is one of the challenging problems in computer vision today. The applications of tracking objects can be seen in different fields such as surveillance, traffic control and security sectors. **Objective:** In this paper, two different techniques are used to track colored objects. The first technique is based on moving object region, this method identifies and tracks a blob token or a bounding box, which is calculated for connected components of moving objects in 2D space. It's relies on properties of these blobs such as size, color, and centroid. The other technique is Kalman filters, introduced in the early 1960's by Rudolf Emil Kalman. In this paper a system in Matlab that is able to track and estimate the position of colored moving ball from the background in a video sequence based on background subtraction analysis in a sequence of frames with different resolution is proposed. **Results:** The presented system was tested using different background noises and different colored balls. The system was successful in the tracking and estimating process. **Conclusion:** The obtained results show that the proposed system was able to successfully track colored objects using the Kalman filter technique

© 2014 AENSI Publisher All rights reserved.

ToCite ThisArticle:Names of authors., Paper title. *Aust. J. Basic & Appl. Sci.*, 8(3): 532-540, 2014

INTRODUCTION

Tracking objects based on color is one of the quickest and easiest methods for tracking an object from one image frame to the next. The applications of tracking objects can be seen in various fields such as surveillance, security and traffic control. In this paper a tracking system for tracking colored objects is proposed. The system tracks a ball based on specific color and draw a boundary around it then use a Kalman filter to reduce the noise of estimation. The system will take the frames that are taken from a video stream then it will detect the object. The object will undergo pre-image processing stages as shown in Figure 1.

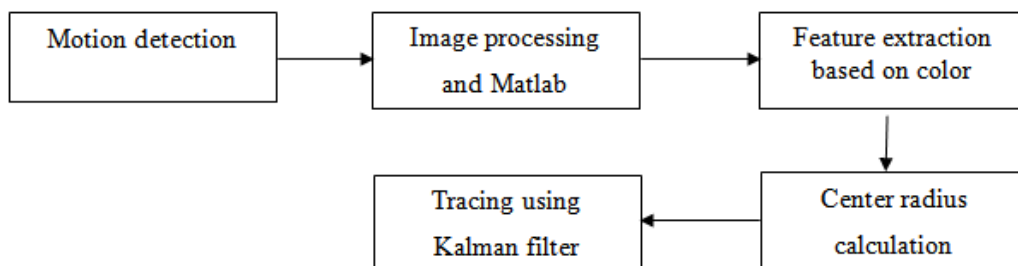


Fig. 1: System overview

The paper is organized as follows the next section will introduce the image processing techniques done on the tracked objects. Section 3 will introduce the Kalman filter, while section 4 introduces the experiment and results. Finally the conclusion is presented in section 5.

Image Processing:

In this section image processing is used to extract the traced object from the background. This process consists of several stages such as: Converting the color image to grayscale image, Converting the grayscale image to Binary image, Image opening and closing, removing object connected to boundary and finally feature

Corresponding Author: Dr. Anas Quteishat, Al-Balqaa Applied University, Faculty of Engineering Technology Computer Engineering Department
E-mail: Anas.Quteishat@fet.edu.jo

extraction. Before embarking in the image processing process motion detection is presented in the next subsection.

2.1. Motion Detection:

The most basic of detecting motion is the Differential Motion analysis which detects motion by considering the difference values of two consecutive frames. There also exists more complex methods such as the Lucas-Kanade-Tomasi optical flow method (Shi and Tomasi,1994, Angelo,et.al,2008). Both methods have advantages and disadvantages but for its simpler implementation and less complexity we chose to use the Differential Motion Analysis method. The image difference in absence of motion is easy to implement; it happened by subtracting the current frame from previous frame, if there is no motion in the second frame, the difference will be theoretically zero, but in the presence of motion the difference will not be zero and the motion will detection as shown in Figure.2 (Eduardo M. and Charlotte P.2007)

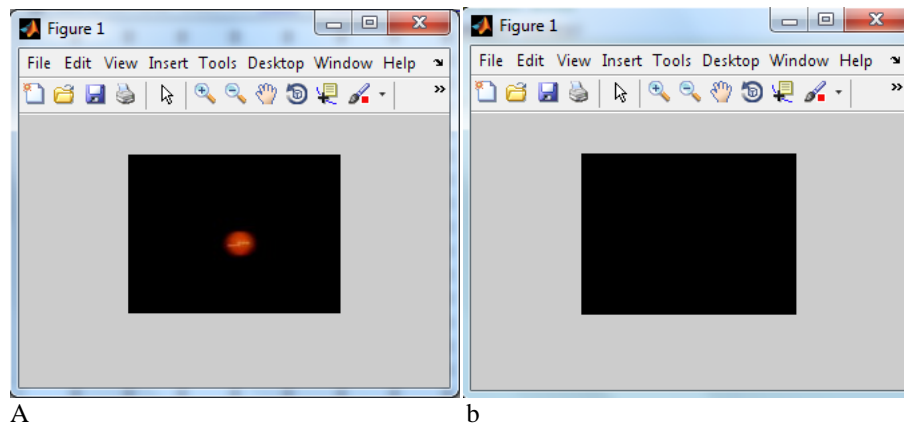


Fig. 2: subtract the background image, background, from the original image

Figure.2. Shows the subtract the background image. Figure 2.a shows the original image. Figure 2.b shows the background. After subtracting the images, maybe there is some value in the absence of motion in uncontrolled work space; this may be due to in the light source or the reflection of the background (Ganesh and Anthony,2007, Gary Bisho,2012). To solve this problem the image difference is compared to a certain threshold value to remove small changes than can be the result of changes in lighting conditions or noise. If the result of changing is a smaller than the threshold then there is no movement; on the other hand if it's equal to or larger than the threshold then there is movement in the object presence of the frame. These thresholds is obtained by making a lot of experiments obtained by checking for the highest value of the difference frames in the absence of motion, and after repeating this experiment several times we find that the good value is (10),and we see that this value depends on the environment the tracking system is around the object.

$$I_{threshold}(x, y) = \begin{cases} 1, & I_{difference}(x, y) > threshold \\ 0, & I_{difference}(x, y) \leq threshold \end{cases} \quad (1)$$

2.2 . Object Tracking Based On Color:

Tracking objects based on color is one of the quickest and easiest methods for tracking an object from one image frame to the next. The speed of this technique makes it very attractive for near real time applications but due to its simplicity many issues exist that can cause the tracking to fail. In this step we use the RGB subtraction values to remove all objects in the image except the predefined color ball, and then normalizes the remaining. As we know previously any colored image is consists of three basic colors RGB (Red, Green and Blue) and each color is one matrix, i.e. each colored image consists of three matrices. In our approach we use a simple subtraction method can be represent easily as follow: Such as if we want to tracking the red object in the absence of motion, we must subtract the red component from the grayscale image to extract the red components in the image, after that we have just the component of red color and the other colors (Green, Blue) will be disappear because they are subtracted from the red components. Figure 3 shows the explained method.

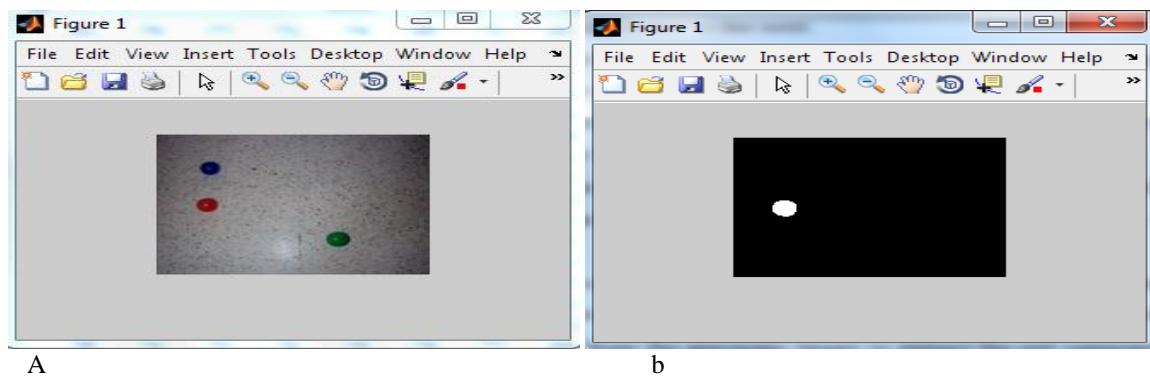


Fig. 3: Object tracking based on color

Figure 3 showstracking objects based on color by subtracting. Figure 3 a shows the original image, figure 3 b. show the tracking of red ball after subtracting and converting the original image into binary image. And so on for the other colors.

2.3. Object Tracking Based on Region:

In this step we do the image blob analysis to get a set of properties for each object region. After we filter the frames based on combinations of dilation and erosion to reduce the influence of noise, followed by a connected component analysis for labeling each moving object region, very small regions are discarded. First, at this stage we calculate the centroid for the moving object which is an approximation for the center of the moving object. For moving ball tracking we use a hybrid method based on bounding box and the center of mass of the ball, we select the largest object, then do bubble sort (large to small) on regions in case there are more than one object. During the initialization period a data record is generated for each object: a label for indexing, the centroid of the ball using two variables (cc,cr) and radius, After that we will use the largest object related to the initialization data record. But the radius will be defined by mathematical equations:

$$R = \sqrt{A / \pi} \quad (2)$$

$$A = \pi \times R^2 \quad (3)$$

where is R is the radius of the ball, A is the area of the ball.

The next step is predicting the position of the ball using Kalman filtering. For each new frame, the predicted position is searched to see whether it can find any match with the previous data record, if a matching object region is found, it is marked as successfully tracked and belongs to a normal move; if we cannot find any match, then the object may have changed lanes, or stopped, or exceeded the expected speed. If matched, then it is also marked as successfully tracked; if still not yet matched, it will be marked as a new object and added to the data record. If an existing object is not being tracking for our frames, it will be marked as stopped. According to the video capturing speed, we also define a threshold, which is used for marking tracking finished. Matching is performed within certain thresholds for the different feature vector elements. The three main elements used for matching are: same color, a linear change in size, and a constant angle between the line corner point-upper left point versus the line corner point-lower bottom point.

Occlusions are reported if bounding boxes are overlapping. In case of partial occlusions, calculated corners and further feature vector elements are tested for making a decision. Finally the data record will be updated using the results of the matching process. For object such as center of the ball using two variables (cc,cr) and radius.

3. Kalman Filter:

The purpose of the Kalman filter is to mitigate the noise and other inaccuracies in our measurements. In our case, the measurement is the x,y position of the object that has been segmented out of the frame, in our system we work with 2D images. What the Kalman Filter does is use a model to predict what the next position should be assuming the model holds true, and then compares that estimate to the actual measurement pass in (Greg W. and Gary B., 2006). The actual measurement is used in conjunction with the prediction and noise characteristics to form the final position estimate and update a characterization of the noise (measure of how much the measurements are differing from the model).

Our system can work as follow:

- Predict .
- Associate with a measurement .
- Correct using the measurement.

Using the last few readings, you can estimate the ball's velocity and estimate where it will be at any time in the future (this is the prediction part of the *Kalman filter*). Mathematically, Kalman filter is an estimator that predicts and corrects the states of wide range of linear processes. the optimal state is found with smallest possible variance error, recursively. However, an accurate model is an essential requirement. First of all, we have to initialize consists of constructing a model transition matrix A , the process noise covariance matrix Q , the measurement covariance matrix R , the measurement transition matrix H found in the formulas:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \quad (4)$$

$$z_k = Hx_k + v_k \quad (5)$$

Where x_k is a model vector, z_k is a measurement vector, w_{k-1} is model noise drawn from the distribution identified by Q and v_k is the measurement noise drawn from the distribution identified by R .

$$p(w) \sim N(0, Q) \quad (6)$$

$$p(v) \sim N(0, R) \quad (7)$$

In our case, all the parameters are independent and we pretend there is no transformation needed to convert measurement vector data into model vector data (e.g. rotation or scaling).

Therefore, the H matrix is initialized as an identity matrix, the R matrix is initialized as a one-valued diagonal with the value specified by the experiment, and the Q matrix is also initialized as a one-valued diagonal again with the value specified per experiment. Had there been any dependencies (say, between the velocities, we would have needed to encode those in the covariance matrices).

The A matrix is tightly coupled with our state vector and represents our expected transformation of a given state. In other words, it tells us how the parameters change with each iteration or time step. In algebraic form, using all parameters, it looks like:

$$\begin{aligned} 1*x + 0*y + 1*V_x + 0*V_y + 0*A_y &= x \\ 0*x + 1*y + 0*V_x + 1*V_y + 0*A_y &= y \\ 0*x + 0*y + 1*V_x + 0*V_y + 0*A_y &= V_x \end{aligned}$$

It's easy then to see that the matrix for A should be:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ \frac{dt}{dt} & 0 & 1 & 0 \\ 0 & dt & 0 & 1 \end{bmatrix}.$$

Next, whenever you receive a new reading, do a weighted average of the reading and of your estimate obtained in the previous step (this is the update part of the Kalman filter). The result of the weighted average is a new estimate that lies in between the predicted and measured position, and is more accurate than either by itself.

We use a simple "closest" algorithm to match the first set of three measurements with the next set of three measurements (since we don't know which measurement belongs to which ball having randomized it). From this pairing, we can make an estimate of the velocities ($V_x = x_1 - x_2$, etc.) and we set A_y to a "known" constant.

Figure 4 shows the results an estimation using Kalman filter where Red crosses represent the actual data, blue represent Kalman gain, green and light blue represent the noise measurement that happened in update time.

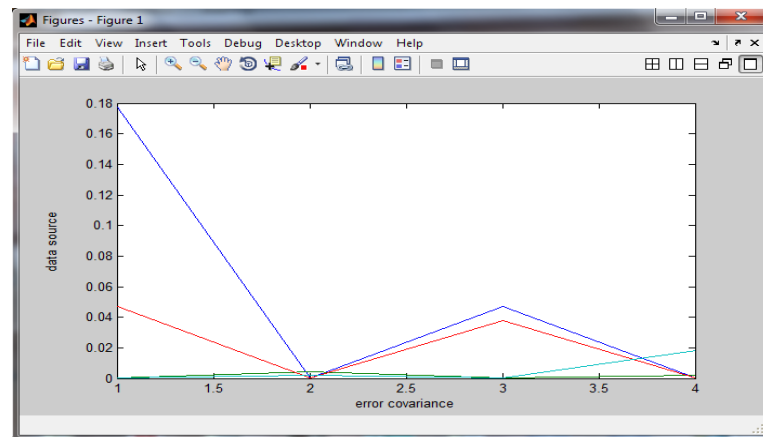


Fig. 4: Kalman filter correctio

Experiment and Results:

The experiments for the system were done on different number of object, different colored objects and different back grounds. The operating system used to run the software is Windows® 7, and the software was implemented using Matlab 10b. The image size used in the experiment was 240X320 - 120xX160 pixels, the system processing time is 0.2 seconds per frame.

4.1 Tracking a Single Ball:

The first experiment was conducted on a red ball with a black background. The black background presents a background without any noise, where the lighting does not affect the ball. Figure 5. a, shows the red moving in the black background, while figure 5. b shows the coordinates of the centroid of the ball.

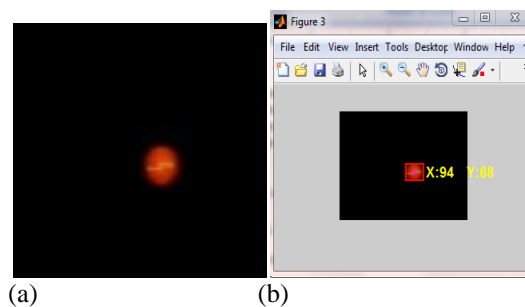
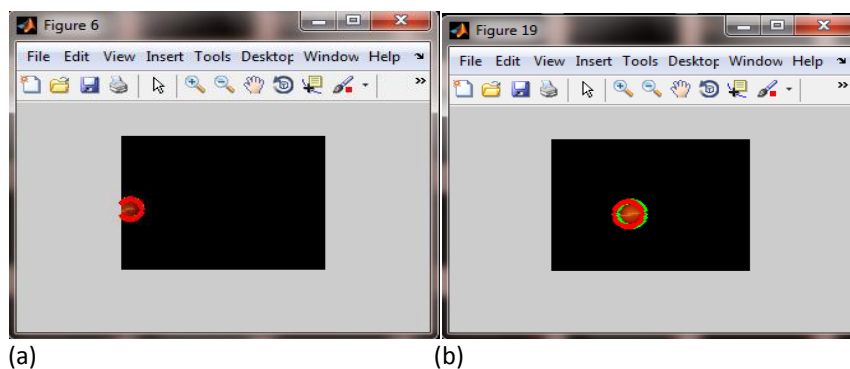


Fig. 5a: The red ball moving a black background. b: The coordinates of the centroid of the red ball.

The next experiment was to investigate the ability of the system to track the red ball using Kalman filter estimation. Figure 6 shows the ball moving from left to right. In the figure, the boundary of the ball is in green color while the estimated position is in red color. One can notice that the estimation for the ball position was very near to the real position for the boundary of the ball.



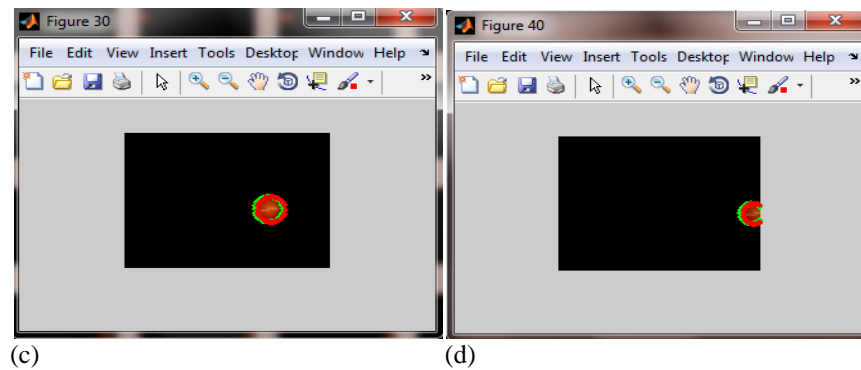


Fig. 6: Tracking the red ball using Kalman filter

Figure 6, shows the tracing of the red ball in the black background. (a) show the tracing at the beginning of the tracking process, (b) and (c) show the tracing at the middle of tracking process Finally (d) shows the end of tracking. Form the figure can notice that the Kalman filter Presented by the red circle and the tracking represented by green cycle. In this experiment, the frame size of 120x160, to trace the 40 frames.(b).

4.2.Tracking Two balls in a Black Background:

In this experiment another ball was presented to the tracking system along with the blue ball.

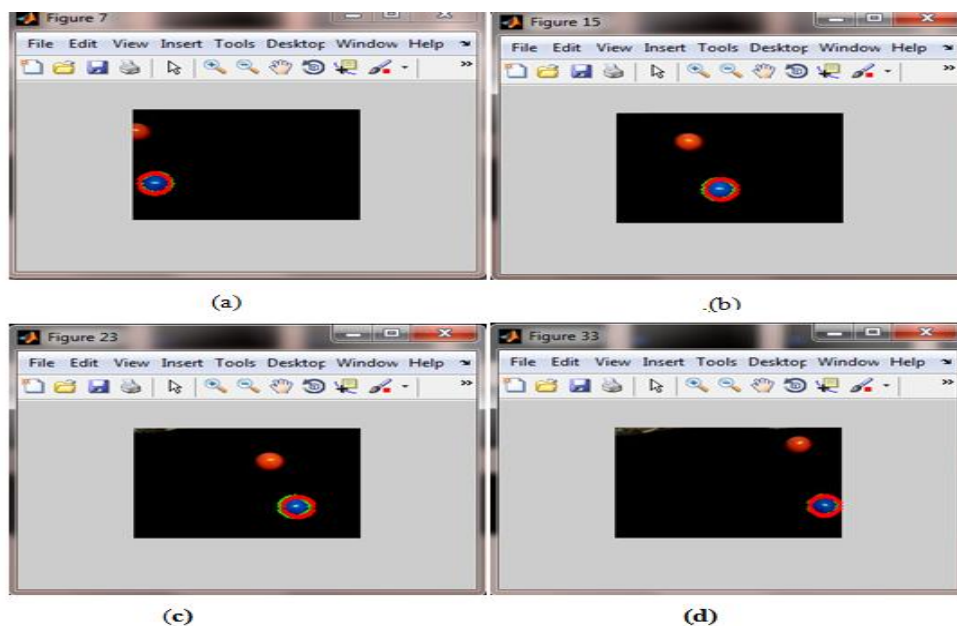


Fig. 7: Tracing the blue ball in the presence of the red ball.

Figure 7 shows the tracing of the blue ball in the presence of the red ball. Note that the green boundary shows the actual boundary of the ball while the red boundary is the Kalman filter estimation for the ball. shows the tracing of the blue ball in the same black background in figure 7. (a) show the tracing at the beginning of the tracking process, (b) and (c) show the tracing at the middle of tracking process ,Finally (d) shows the end of tracking. form the figure can notice that the Kalman filter presented by the red circle and the tracking represented by green cycle. In this experiment, the frame size of 120x160, to trace the 35 frames.

4.3.Tracing Balls In a Noisy Background:

In this subsection two experiments are presented. The two experiments where done to trace a red ball in the noisy backgrounds. The first background is a shinny will light background while the second background is a dark with no lighting back ground.

4.3.1. Tracing Balls In a Shiny Background:

In the first experiment the background was a shiny will light background; note that reflection from the light source is presented. The task was to track the blue ball in the presence of the red ball. Again, the green line presents the actual boundary while the red line presents the estimated boundary using Kalman filter

In this experiment, the tracing time was recorded to compare the frame size effect on the tracing process. For frame size of 120x160 it took the program 78.87 seconds to trace 90 frames, while for frame size of 240x360 it took the processor around 80.87 seconds to trace the same 90 frames, this show that these two frame sizes has almost the same tracing time.

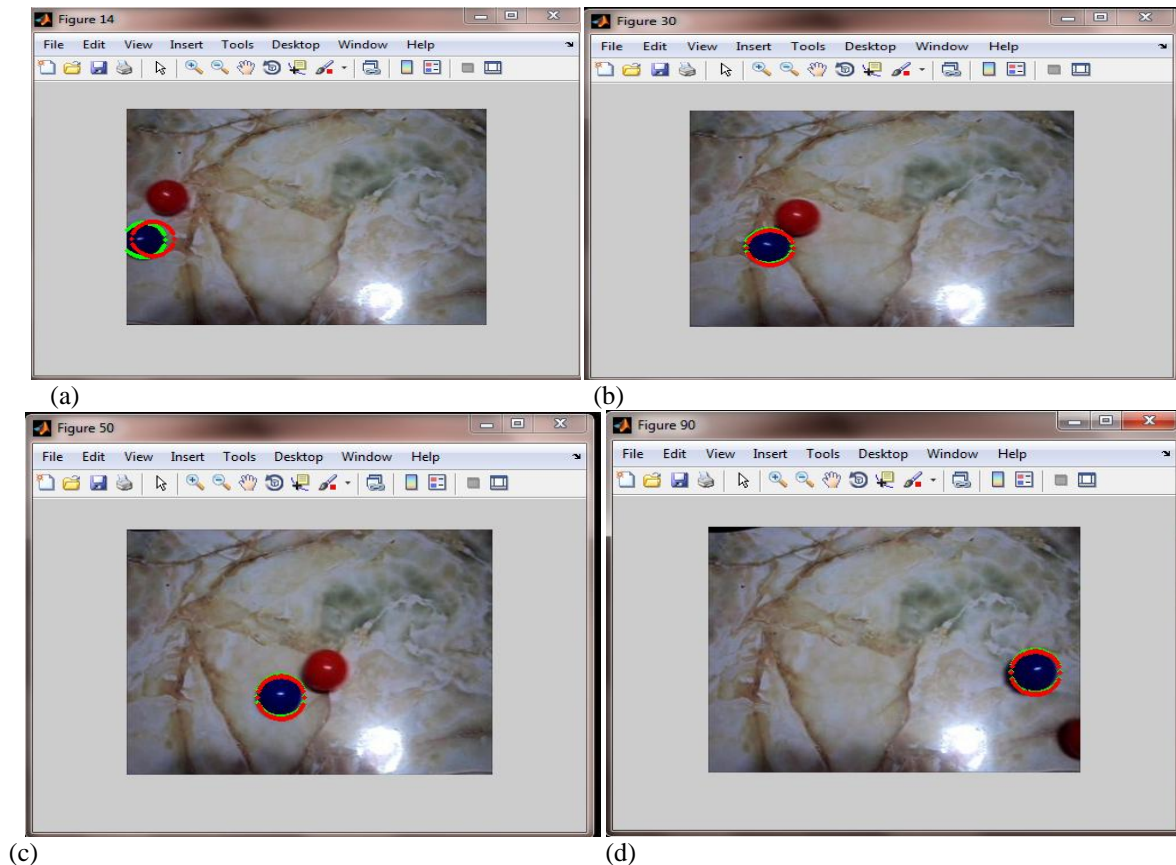


Fig. 8: Tracing a blue ball in a noisy back ground

Figure 8, shows the tracing of the blue ball in the noisy back ground. Comparing figures 7 and 8 one can notice that the Kalman filter traced the blue ball more efficiently in the black background, nevertheless the tracing in the noisy back ground was successful and the ball was traced throughout all the frames.

4.3.2. Tracing Balls In aDark Back Ground:

In this experiment three balls were presented to the system: red, blue and green balls. The task was to trace the blue ball and the green ball separately. So the first try was to trace the blue ball. In figure 10, the green lines present the actual boundary of the ball while the red line presents the estimated boundary of the ball. In this experiment, the three balls where collided together so their motion was not linear as graph below which presented the behavior of each ball:

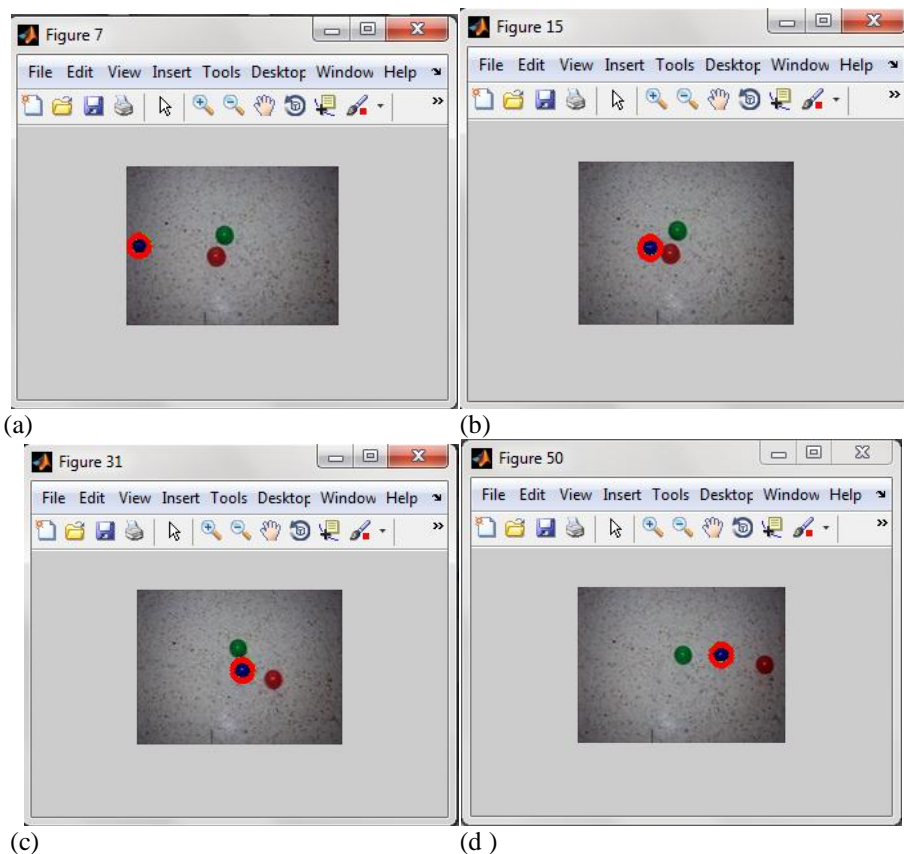


Fig. 10: Tracing a blue ball in a dark background

Figure 10 shows tracing a blue ball, (a) and (b) shows the tracing before collision, (c) shows the moment of collision when blue ball coiled with green ball and the green ball change its direction in (d) figure. The next task was to trace a green ball in the same dark background. In this case the green ball color was too near to the background and it was not detectable. Figure 11 shows the attempt to trace the green ball and how the system did not detect it.

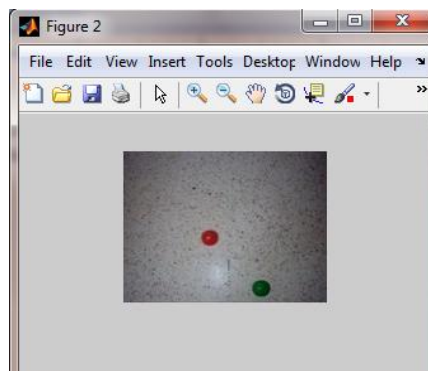


Fig. 11: tracing the green ball in a dark background

In the case of the green ball the thresholding level needed to be decreased from 0.09 to 0.01, this decreasing in the threshold level has its advantages and disadvantages. Its disadvantages are the noise that could be mistreated as moving objects. While it's advantaged is the fact that the system was able to trace dark objects. The parameters of the thresholding levels where taken by try and error method. Figure 12 shows the successful attempt to trace the green ball.

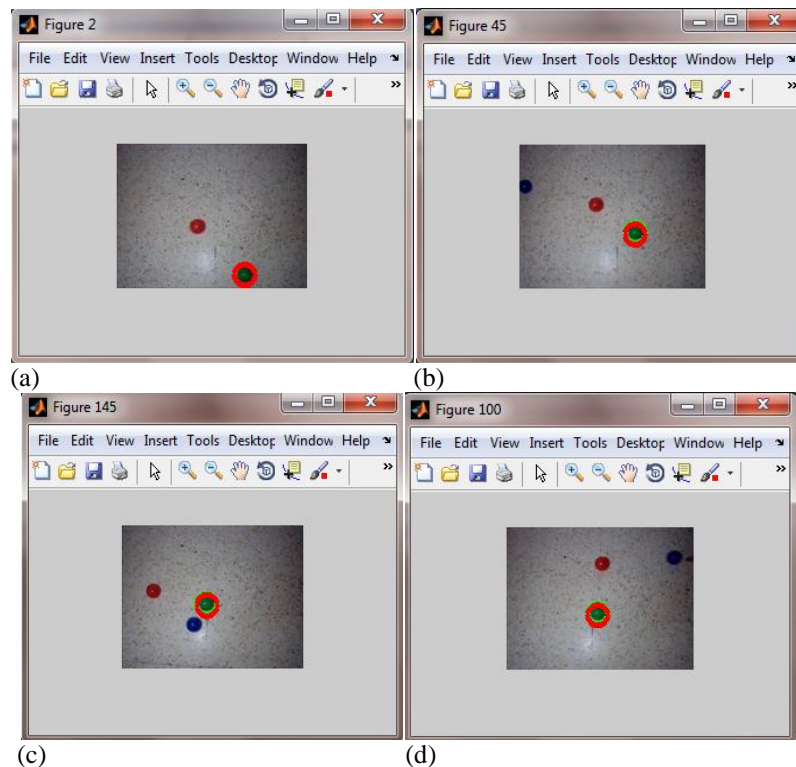


Fig. 12: successful attempt to trace the green ball

Figure 12, shows the tracing of the green ball in the same dark background. (a) Show the tracing at the beginning of the tracking process, (b) and (c) show the tracing at the middle of tracking process, finally (d) shows the end of tracking. From the figure we can notice that the Kalman filter presented by the red circle and the tracking represented by green cycle.

Conclusion:

In this work, a methodology for a tracking system using background subtraction and *Kalman filter*. The tracking system consists of a sequence of frames that took from low resolution camera with different background. the software that control the image process is *Matlab*. In the simplest form, tracking can be defined as the problem of estimating the trajectory of an object in image as it moves in a scene. In other word, a tracker assigned consistent labels to the tracked object in different frames; depending on the tracking domain; and it also can provides object-centric information ,such as area or shape of an object.

To improve the system, it might be possible to model the shape of the object and feed that to an adaptive filter, such as the *Kalman filter*. Introducing the Kalman filter would allow more complex constraints that are also adaptive during runtime.

REFERENCES

- Angelo Bosco, Arcangelo Bruna, Sebastiano Battiato, Giuseppe Bella, and Giovanni Puglisi, 2008. "Digital Video Stabilization through Curve Warping Techniques" IEEE Transactions on Consumer Electronics, 54: 2.
- Carlo Tomasi and Takeo Kanade, 1991. "Detection and Tracking of Point Features." Carnegie Mellon University Technical Report CMU-CS-91-132.
- Gary Bisho University of North Carolina at Chapel Hill Department of Computer Science Chapel Hill, NC 27599-3175 <http://www.cs.unc.edu/~{welch, gb}> Greg Welch {welch, gb}@cs.unc.edu, accessed on 15-April-2012 at 1:30 pm.
- Greg Welch and Gary Bishop, 2006. An Introduction to the Kalman Filter, TR 95-041 Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, Updated: Monday.
- Eduardo Monari, Charlotte Pasqual, 2007. "Fusion of Background Estimation Approaches for Motion Detection in Non-Static Background", IEEE Advanced Video and Signal Based Surveillance, pp: 347-352.