International Journal of Electronics Communication and Computer Engineering
Volume 5, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Simulink Model Design for Real-Time Moving Object Detection using Background Subtraction Algorithm

Prof. R. P. Patil
E & TC Department, SKNCOE, Vadgaon Bk, Pune, India
Email: rohita_jagdale@yahoo.com

Ms. P. D. Mahamuni
E & TC Department, SKNCOE, Vadgaon Bk, Pune, India
Email: pdmahamuni@gmail.com

Abstract **– Detecting regions of change in multiple images of the same scene taken at different times is of widespread interest due to a large number of applications in diverse disciplines, including remote sensing, surveillance, medical diagnosis and treatment, civil infrastructure, and underwater sensing. Background subtraction methods are widely exploited for moving object detection in videos in many applications. In recent years extensive investigations and analyses have been done in the domain of moving object detection. Detection of moving objects in video processing plays a very important role in many vision applications. The vision systems that include image processing methods are widely implemented in many areas. This background subtraction algorithm is implemented using MATLAB/ Simulink. It provides number of blocks related to the image processing and it is very easy to implement the models.**

Keywords **– Motion Detection, Background Subtraction Algorithm, FPGA, VGA Monitor.**

I. INTRODUCTION

Motion detection is a very important part for many computer vision applications, like human or vehicle tracking. These applications require segmenting the motion region out of the scene. Various motion detection methods have been extensively investigated in the literature. However, background subtraction methods are typically used when dealing with fixed cameras. Each application that benefit from smart video processing has different needs, thus requires different treatment. However, they have something in common: moving objects. Thus, detecting regions that correspond to moving objects such as people and vehicles in video is the first basic step of almost every vision system since it provides a focus of attention and simplifies the processing on subsequent analysis steps. Due to dynamic changes in natural scenes such as sudden illumination and weather changes, repetitive motions that cause clutter (tree leaves moving in blowing wind), motion detection is a difficult problem to process reliably. Frequently used techniques for moving object detection are background subtraction, statistical methods, temporal differencing and optical flow whose descriptions are given in second chapter.

Complex traffic surveillance systems are often used for controlling traffic and detecting unusual situations, such as traffic congestion or accidents. Most such systems are built using high resolution cameras connected via a high-bandwidth link to the processing center. The need for automated processing of video data is obvious and many solutions of systems for motion analysis is should be real-time. For this approach uses a single, stationary, constant zoom, monochrome camera to detect moving and recently stopped vehicles. The result of the algorithm is a binary mask image of blobs representing the detected objects and a table with the parameters of the objects. A low resolution camera can be used, since the detected objects (vehicles) are large enough and their details are not important for this application.

II. LITERATURE SURVEY

Any image has two models [1] background model, foreground model which are explained as follow:

1) Background Model: In background subtraction, the general assumption is that a background model can be obtained from a training sequence that does not contain foreground objects. Moreover, it usually assumes that the images are captured by a static camera. Thus, foreground objects can be detected by checking the difference between the testing frame and the background model built previously.

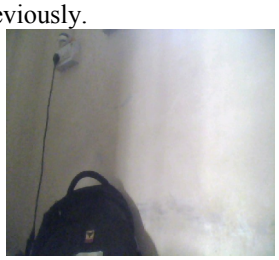


Fig.1. Background Model

Fig.2. Foreground Model

Fig.1 shows the background model in which the object is not present only the background is present. The scene representation is called the background model. The background intensity should be unchanged over the sequence except for variations arising from illumination change or periodical motion of dynamic textures. Thus, background images are linearly correlated with each other [1-4].

2) Foreground model: The foreground is defined as any object that moves differently from the background. Foreground motion gives intensity changes that cannot be fitted into the low-rank model of background. Thus, they can be detected as outliers in the low-rank representation. Generally, the foreground objects should be contiguous pieces with relatively small size. Fig.2 shows the foreground model in which the moving object is present. This is the main difference between background and foreground model. And this difference will be identified by applying it to background subtraction algorithm [1-4].

A) Generic Flow of Algorithm

Each application that benefits from the smart video processing has different needs, thus require different treatments. However, they have something in common that

is moving objects. In generic flow of algorithm, as shown in Fig. 3 detecting regions that correspond to moving objects such as people and vehicles in video is the first basic step of almost every vision system since it provides a focus of attention and simplifies the processing on subsequent analysis steps. Due to dynamic changes in natural scenes such as sudden illumination and weather changes, repetitive motions that cause clutter (tree leaves moving in blowing wind), motion detection is a difficult problem to process reliably. Frequently used techniques for moving object detection are background subtraction, statistical methods, temporal differencing and optical flow whose descriptions are given below [5-7].

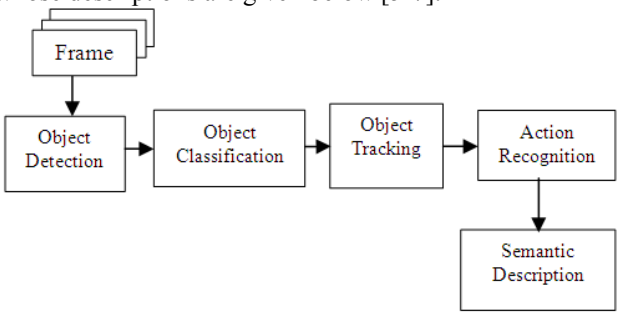


Fig.3. A Generic Frameworks for Smart Video Processing Algorithms

In object detection scenario, an object can be defined as anything that is of interest for further analysis. For instance, boats on the sea, fish inside an aquarium, vehicles on a road, planes in the air, people walking on a road, or bubbles in the water are a set of objects that may be important to track in a specific domain. Objects can be represented by their shapes and appearances. In this section, first describe the object shape representations commonly employed for moving object detection and then address the joint shape and appearance representations. Different object representation methods are articulated shape models, primitive geometric shapes, object silhouette & contour, points & templates, skeletal models, active & multi-view appearance models.

B) Motion Detection Methods

Some algorithms for the motion detection are available which are described as follows:

1) Temporal Difference: Temporal differencing attempts to detect moving regions by making use of the pixel-by-pixel difference of consecutive frames (two or three) in a video sequence. This method is highly adaptive to dynamic scene changes; however, it generally fails in detecting whole relevant pixels of some types of moving objects. The mono colored region of the human on the left hand side makes the Temporal Differencing Algorithm to fail in extracting all pixels of the human's moving region. Also, this method fails to detect stopped objects in the scene.

Additional methods need to be adopted in order to detect stopped objects for the success of higher level processing. Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels, e.g., there may be holes left inside moving entities. As an example of this method, detect moving targets in real video streams using temporal differencing. After the absolute difference between the current and the previous frame is obtained, a threshold function is used to determine changes. By using a connected component analysis, the extracted moving sections are clustered into motion regions. An improved version uses three-frame instead of two-frame differencing.

2) Optical Flow: Optical flow methods make use of the flow vectors of moving objects over time to detect moving regions in an image. They can detect motion in video sequences even from a moving camera. However, most of the optical flow methods are computationally complex and cannot be used real-time without specialized hardware [8]. Optical-flow-based motion segmentation uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. For example, compute the displacement vector field to initialize a contour based tracking algorithm, called active rays, for the extraction of articulated objects. Optical-flow-based methods can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex, very sensitive to noise and cannot be applied to video streams in real time without specialized hardware.

The optical flow algorithm is fails to detect the moving object in the frames. It required large number of calculations, sensitive to noise and calculations are complex. So it is not suitable for the real-time applications. The background subtraction and temporal difference both is low cost algorithm but the temporal difference algorithm is fails to detect the color image object shape. So the Background Subtraction Algorithm is used for the real-time moving object detection.

III. System Design

In this paper, real-time moving object detection is described using background subtraction algorithm. Motion detection methods are basically a process which detects the object in the surveillance area [1]. This algorithm is design in the Simulink. In Simulink library it gives large number of the block sets which are related to the image processing system. The direct blocks are available for the image read and operations performed on that image. The Fig.4 shows the system block diagram.

1) Input from Web Camera: The input images or video for the system are taken from the web camera of the laptop for real-time results. The background is same for overall video because the web camera is stable. So first capture the background image in which only the background is present. Then the real-time video is captured.

2) Background Subtraction Algorithm: Both the inputs, background image and real-time video is given to the background subtraction algorithm. The process algorithm is described as follow:

1. Sequences of video frames
2. Preprocessing on frames
3. Subtraction of Background frame and video frame
4. Segmentation using thresholding

5. Morphological filtering
6. Detection of moving object

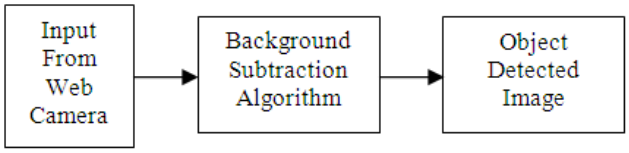


Fig.4. System Block Diagram

a) Sequences of Video Frames: Input to the background subtraction algorithm is the sequence of the real-time video frames. The video is taken from the web camera of the laptop. Video is nothing but the sequence of frames.

b) Pre-processing on Frames: This is the first step in background subtraction algorithm and is performed on each frame which is to be subtracted. Pre-processing is done to improve the image data that suppresses unwanted distortions or enhances some image features important for further processing. Due to impurities in the background, the discrepancy image obtained contains the motion region as well as noise. This noise might be included in the image due to environmental factors, illumination changes, during transmission of video from the camera to further processing. Therefore, noise needs to be removed. For this system median filter is used in Simulink model. In Simulink library the 'median filter' block is directly available for pre-processing. This in turn blurs the image frames which help in shadow removal.

c) Subtraction of Background and Video Frames: Pre-processing reduces the noses from the each frame. The real-time video frames are subtracted from the background frame continuously. It removes the background from the image to detect the object in that image. This happens because backgrounds of the both input images are same so the pixel value of the background in both input images are same. In this the subtraction is done pixel by pixel.

d) Segmentation using Thresholding: Thresholding is a procedure that eliminates an unwanted range of pixels in the scene with respect to certain threshold values. Data validation is involved with the collection of techniques to reduce the misclassification of pixels. The threshold value is very important for the clarity of the image. Deciding the threshold value gives the pixel information, when a pixel is supposed to be considered as either a background pixel, or a foreground pixel.

After the subtraction of two images, the pixel value is compared with the threshold value and then tagged to '1' or '0'. After the segmentation the object detected image is in binary form.

$| f(x,y) - B(x,y) | < T_d$;

pixel tagged to 0= background (black) (1)

$| f(x,y) - B(x,y) | > T_d$;

pixel tagged to 1= moving object (white) (2)

Where, $f(x,y)$ is the pixel value of background image.

$B(x,y)$ is the pixel value of object image.

e) Morphological filtering: The segmented frame is now given to the morphological filtering for reducing the noise. The function of the morphological filtering is removing the small regions probably created by noise; fill up unnecessary cavities, smoothing boundaries, extracting edges. It will give pixel level operations.

f) Motion detection: After the segmentation and morphological filtering the moving object is clearly seen in the frame and that will be the output of the system which is display on the VGA monitor.

C. Simulink model for Real-time Moving Object Detection

The Simulink model shown in Fig. 5 is prepared in MTALAB with the help of Simulink library browser which contains the Image and Video processing block set. The software model for object detection consists of Image from file, Video Viewer, Embedded function block, parameter blocks. Both the input images are displayed using the video viewer block. Each time run the Simulink model, the different image dataset has to be selected. For real-time moving object detection this model is run for two times one time for 2-3 seconds for taking background frame and second time for real-time video of 20-30 seconds. Both the frames are passing through the pre-processing. In pre-processing median filtering of frame is done, also the color images are converted into gray scale.

1) from Video Device: The From Video Device block is used to acquire image and video data streams from image acquisition devices, such as camera and frame grabbers, in order to bring the image data into a Simulink model. The block can be configured and previewed the acquisition directly from Simulink. The From Video Device block opens, initializes, configures, and controls an acquisition device. The opening, initializing, and configuring occur once, at the start of the model's execution. During the model's run time, the block buffers image data, delivering one image frame for each simulation time step. The block has no input ports. The block can be configured as to have either one output port, or three output ports corresponding to the uncompressed color bands, such as red, green, and blue, or Y, Cb, Cr but for this system gray images are required so one port device is used and other ports are terminated. It will capture the images of size 240 x 320. This block's sample rate is set to 1/30. Specify the sample time of the block during the simulation. This is the rate at which the block is executed during simulation. The default is 1/30. The block sample time does not set the frame rate on the device that is used in simulation. Frame rate is determined by the video format specified (standard format or from a camera file). Some devices even list frame rate as a device-specific source property. Frame rate is not related to the block sample time option in the dialog. Block sample time defines the rate at which the block executes during simulation time.

2) Image from Workspace: This block is used for taking the background image from workspace. Use the Value parameter to specify the MATLAB workspace variable that contains an expression that specifies the image to import into model. Use the Sample time parameter to set the sample period of the block. The image signal type parameter is available in property and for this image it is selected to one multidimensional signal; data type of output image is set to double.

3) To Frame: The Frame Conversion block passes the input to the output and sets the output sampling mode to the value of sampling mode of output signal parameter, which can be either Frame-based or Sample-based. The output sampling mode can also be inherited from the signal at the Ref (reference) input port, which can be made visible by selecting the Inherit output sampling mode from <Ref> input port check box. The Frame Conversion block does not make any changes to the input signal other than the sampling mode. In particular, the block does not re-buffer or resize 2-D inputs. Because 1-D vectors cannot be frame based, when the input is a length-M 1-D vector and the block is in Frame-based mode, the output is a frame-based M-by-1 matrix i.e. a single channel.

4) Median Filter: It performs median filtering of input matrix I. Use the Neighborhood size parameter to specify the size of the neighborhood over which the block computes the median. Use the output size parameter to specify the dimensions of the output. If select same as input port I, the block outputs an intensity image that is the same size as the image input to the I port. If select valid, the block only computes the median where the neighborhood fits entirely within the input image, so no padding is required. For this image 'same as input' is selected and padding is set to zero.

5) Embedded Function Block: This block is also called as S-function block; in this block code for the designed algorithm is embedded. This block is programmed using Background Subtraction algorithm for subtraction of the two input images, code for thresholding of the subtracted image and the software model is prepared.

6) Unit8: The Data Type Conversion block converts an input signal of any Simulink data type to the data type specified for the output data type parameter. The input can be any real- or complex-valued signal. If input is real, the output is real.

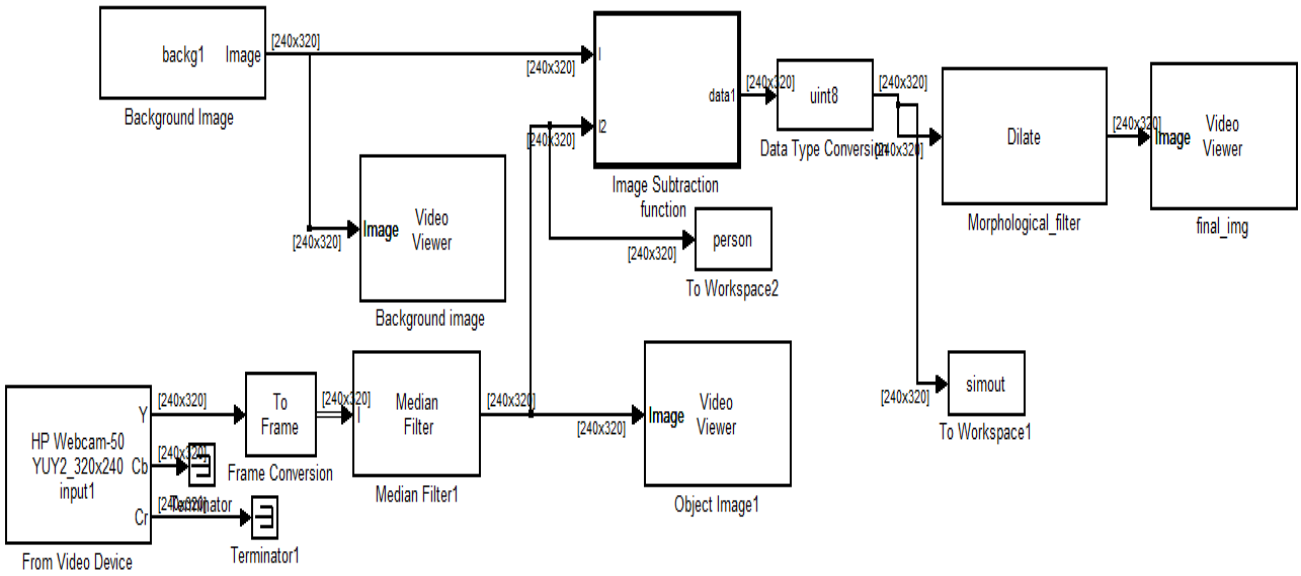


Fig.5. Simulink Software for Motion Detection

If the input is complex, the output is complex. This block requires specifying the data type and scaling for the conversion. If to inherit the data type or scaling from an input signal, use the Data Type Conversion Inherited block. The input and output needs to have equal parameter to know how the block handles the input.

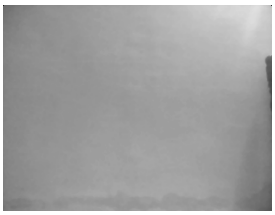
7) Simout: This block used to write input to specified Timeseries, array, or structure in a workspace.

For menu-based simulation, data is written in the MATLAB base workspace. Data is not available until the simulation is stopped or paused. For command-line simulation using the sim command, the workspace is specified using DstWorkspace field in the option structure. This block is used to log a bus signal, and uses "Timeseries" save format.

8) Dilate: This is one of the functions of morphological filtering. The Dilation block rotates the neighborhood or structuring element by 180 degrees. Then it slides it over an image, finds the local maxima, and creates the output matrix from these maximum values. If the neighborhood or structuring element has a center element, the block places the maxima there. It does not have an exact center; the block has a bias toward the lower-right corner, as a result of the rotation. The block places the maxima there.

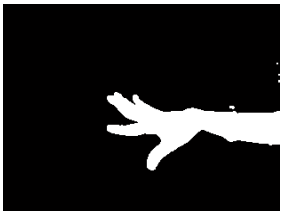
IV. Results And Discussion

The system model is designed using Simulink block sets. The real-time video is subtracted from the background frame and gives the shape of the object in the output frame with respect to the motion of the object. For experimentation of system the moving object is hand the motion of the hand is captured by the web camera of the laptop. The resultant frame continuously shows the shape of the hand motion. To observe the result the some frames of input video with different motion of hand and t subtracted frames are shown in Fig. 6.



(a) Background Frame


(b) Frame 1

(c) Frame 2


(d) Frame 3

Fig.6. Motion Detected Results

A) Parameter Evaluation

For parameter evaluation the ground truth model is used. Parameter evaluation for real-time video is not possible so the still images are used. In that model the resulting subtracted image after the segmentation and morphological filtering is compared with the actual result which is expected from the model. The Fig. 7 shows the ground truth models which is used to compare the images pixel by pixel and decides the pixel values are true or false. Pixel values are categorized into four terms true positive, true negative, false positive, false negative. These TP, TN, FP, FN are related to the foreground and background pixel. Using this model TP, TN, FP, FN terms are calculated and using these values accuracy, precision, recall, MCC this four parameters are calculated.

Background subtraction gives the output of this system and the ground truth gives the output which is actually required from the system. This ground truth output image is prepared in paint by editing the system output image. Generally by looking at the object the shape of the object is recognized and by that reference the changes in that image is done in paint for testing the accuracy of the image. For above boll object, both the images of background subtraction and ground truth output are given to the code which is in MATLAB. It uses 'for loop' for the comparison of the pixel values of both images. True Positive (TP) which represents the number of foreground pixels correctly detected by the algorithm and will compare the pixel value. If it is same then it will check for value of pixel '1' or '0', one represent the foreground pixel.

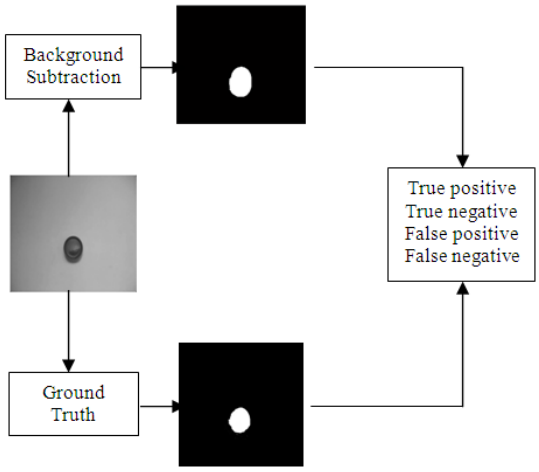


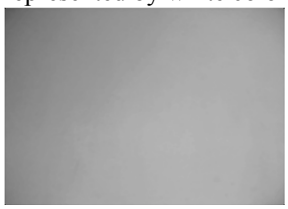
Fig.7. Ground Truth Model

If that pixel value is equal to '1' then TP is increment by one. False Positive (FP) is responsible for the number of pixels which are incorrectly classified as foreground objects.

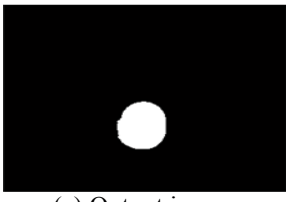
True Negative (TN) indicates the number of background pixels which are correctly detected as background scene by the algorithm. False Negative (FN) stands for the number of pixels corresponding to foreground objects which are misclassified as part of background image.

The Fig. 8(a) shows the background image which the object is not present and this image of size 400 x 400. Fig. 8(b) shows the object image in which the background is same as in background image and the object is ball. So when the subtraction of the background image and object image is done the object is detected in the final image.

The Fig. 8(c) shows the output image in which the object is detected. The subtraction of the image is done pixel by pixel so the background of the current image and the reference image has the same pixel value due to this subtraction of the background pixel is zero so it will be represented in the black color and the moving object is represented by white color.




(a) Background image (b) Object image



(c) Output image

Fig.8. Background Subtraction of Ball Object

So at the end the output image is the binary image. After the binary image getting the morphological filtering is applied to that image and it will perform the operation like opening, closing, sharpening the edges and it will also remove the noise from that frame.

The MATLAB code is used to calculate the four terms. Image manual is the image which is manually edited in paint and another is the image which is obtained at the output of the system. After running that code, it will give the values of TP, TN, FP, FN. TP = 13, TN = 154603, FN = 199, FP = 44 by putting these values in the following formulas it will gives the parameter values.

1) Recall: Recall is the measure of completeness and is defined as number of true positives divided by the total number of elements that actually belong to the foreground objects. (i.e. sum of both, true positives and false negatives). For ball object the values are given above and by putting that values in recall equation 3 value is '0.0613'.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{13}{13 + 199} = \frac{13}{212} = 0.061320 \qquad (3)$$

2) Precision: Precision can be considered as a measure of exactness or fidelity and is evaluated by dividing the number of items (foreground objects) correctly detected by the total number of pixels classified as foreground by algorithm. For ball object the precision value is '0.2281' this is calculated by putting the values of TP, FP values in the equation 4

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{13}{13 + 44} = \frac{13}{57} = 0.2280701 \qquad (4)$$

A good background algorithm is one producing simultaneously a small number of false positives and false negatives, i.e. both a high Precision and Recall value.

3) MCC: MCC is calculated to check whether the segmentation is properly working or not. Given TP the True Positive, TN the True Negative, FP the False Positive and FN the False Negative, to compute the Matthews Correlation Coefficient, because the two classes (motion and background) are of different size. It returns a value between −1 (perfect inverse segmentation) and +1 (perfect segmentation) while 0 signifies a wrong segmentation.

$$MCC = \frac{TP * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \qquad (5)$$

$$MCC = \frac{13 * 154603 - 44 * 199}{\sqrt{(13 + 44)(13 + 199)(154603 + 44)(154603 + 199)}}$$

$$MCC = \frac{2001083}{17008436.45} = 0.1177$$

By putting all the values in above equation the MCC parameter value is '0.1177' which in between '-1' and '1' which tells that the segmentation is properly done by the system.

4) Accuracy: The accuracy is calculated to understand how much the system gives accurate output. The accuracy is calculated for the object detection. It is calculate as follow:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (6)$$

$$\text{Accuracy} = \frac{13 + 154603}{13 + 154063 + 44 + 199} * 100 = 99.8431$$

The accuracy for the object ball in percentage is 99.8431%.

V. Conclusion

The result of background subtraction algorithm is a new image that retains the most desirable information and characteristics of each input image. The subtracted image contains greater information about the object only which is in foreground model. It provides an effective way of reducing the increasing volume of information while at the same time extracting all the useful information from the source images. Designing of Simulink model is easy for the image processing system because it gives direct function blocks for the processing on the image. Using this model the result can be observe at each step immediately and by observing that results the changes in the system can done due to this it gives high accuracy of result. It required less time for designing and observing the system result. Parameters are calculated for the resultant image to perform the analysis of the system. Observing the performance matrices it can be concluded that the background subtraction algorithm gives better results.

References

[1] M. Kalpana Chowdary , S. Suparshya Babu , S. Susrutha Babu, Dr. Habibulla Khan"*FPGA Implementation of Moving Object Detection in Frames by Using Background Subtraction Algorithm"* International conference on Communication and Signal Processing, April 3-5, 2013.

[2] Can Yang, Weichuan Yu "*Moving Object Detection by Detection Contiguous Outliers in the Low Rank Representation*" IEEE transaction on pattern analysis and intelligence, vol. 35, no.3, 2013.

[3] C. S´anchez-Ferreira, J. Y. Mori, C. H. Llanos "*Background Subtraction Algorithm for Moving Object Detection on FPGA* " Department Mechanical Engineering University of Brasilia 2012.

[4] *"Motion and Feature Based Person Tracing In Surveillance Videos"* transactions on computer vision 2011.

[5] Ying-Hao Yu, Q. P. Ha, and N. M. Kwok, "*Chip-based Design for Real-time Moving Object Detection using a Digital Camera Module", The* International Congress on Image and Signal Processing, CISP, 2009

[6] Bahadır Karasulu "*Review and Evaluation of Well-known Methods for Moving Object Detection and Tracking in Videos"* Journal of aeronautics and space technologies july 2010 volume 4 number 4 (11-22)

[7] Thomas B. Moeslund, Adrian Hiton, *"A survey of advances in vision-based human motion capture and analysis",* Computer Vision and Image Understanding 104 (2006) 90–126

[8] Michael J. Black "*The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields"* Computer vision and image understanding Vol. 63, No. 1, January, pp. 75–104, 1996

AUTHOR'S PROFILE

Rohita Patil

Born in September 1981, she earned her Bachelor of Engineering Degree in Electronics Engineering from Shivaji University, Kolhapur, Maharashtra, India in the year 2003, followed by Masters of Engineering in Electronics and Telecommunication Engineering from Dr. Babasaheb Ambedkar Technological University, Lonere in the year 2006.

She has worked as LECTURER in I^2IT, Hinjewadi, Pune Maharashtra, India form 2006 to 2010 . Currently she is working as ASSISTANT PROFESSOR at Smt. Kashibai Navale College of Engineering, Pune, Maharashtra, India. She has numerous papers published in various journals, publications etc.

Prof. Patil is member of the Indian Society of Technical Education.

Prajakta D. Mahamuni

Born in October 1989 she earned her Bachelor of Engineering Degree in Electronics and Telecommunication Engineering from Pune University, Maharashtra, India in the year 2011. She is currently pursuing Masters of Engineering in Electronics and Telecommunication Engineering with VLSI and Embedded System as special subject from the University of Pune, Maharashtra, India.

Currently she is working on Virtex-5 board of FPGA and exploring its functions for image and video based applications.