Object Tracking System Using Approximate Median Filter, Kalman Filter and Dynamic Template Matching

G. Mallikarjuna Rao, Scientist
Control Systems Laboratory, Research Centre Imarat, DRDO, Hyderabad, India
E-mail: mallikarjun.guttikonda@gmail.com

Dr.**Ch.Satyanarayana**, Professor
Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Kakinada. Andhra
Pradesh, India
E-mail: chsatyanarayana@yahoo.com

Abstract— In this work, we dealt with the tracking of single object in a sequence of frames either from a live camera or a previously saved video. A moving object is detected frame-by-frame with high accuracy and efficiency using Median approximation technique. As soon as the object has been detected, the same is tracked by kalman filter estimation technique along with a more accurate Template Matching algorithm. The templates are dynamically generated for this purpose. This guarantees any change in object pose which does not be hindered from tracking procedure. The system is capable of handling entry and exit of an object. Such a tracking scheme is cost effective and it can be used as an automated video conferencing system and also has application as a surveillance tool. Several trials of the tracking show that the approach is correct and extremely fast, and it's a more robust performance throughout the experiments.

Index Terms— Kalman Filter, Approximate Median Filter, Target tracking, Dynamic Template Matching

I. Introduction

Detecting of people has become a very active research area in computer vision community due to the vast number of applications with many applications traffic monitoring, human activity surveillance, people counting and other financial applications. In all these applications extraction of moving objects from the video sequence is the key procedure. Since the camera is not in motion here, the inherent ambiguities of ego-motion is not considered and the scene structure can be rejected at the very beginning [1], [2].

In this work we considered the single object tracking. The motivation behind this work is to develop software for tracking which has the major application in security, surveillance and vision analysis [3]. The developed software must be capable of tracking any single object moving in the frame and to implement on a hardware which is capable of on board calculations with high performance and low power consumption.

This work mainly focussed on developing an algorithm for tracking of an object and determining the velocity of moving object in sequence of frames. Video surveillance of human activity generally needs persons to be detected and background subtraction is a mighty mechanism for recognizing changes in the video sequence [4]. All the evaluation has been performed on a windows PC running MATLAB R2010a. MATLAB has an Image Processing Toolbox for handling images and videos efficiently. Algorithms can be extended for real time applications.

The implementation of this object tracking is explained step by step as follows. In section II, few approaches implemented in real-time for object tracking are explained. In section III, algorithms that are developed to accomplish this work are explained. In section IV, implementation part which includes the logical approach of tracking done for the work is described, evaluation part describes the interface between the hardware and workstation, converting the video into images and loading the images. Finally results are visualized. In section V conclusion and future scope are mentioned.

II. Approaches Of Object Detection And Tracking

The enlargement of high-powered computers, the accessibility of high worth, low-cost video cameras, and the machine-driven video analysis has generated an outsized deal of interest in object detection and tracking algorithms [5].

Single Object Tracking

Object tracking is the process of locating and following the moving object in sequence of video frames. The recorded video may have some noise due to bad weather (light, wind, etc. or due to hardware problems). There are two key steps implementation of our object tracking system:

• Detection of moving objects,

• Tracking of these objects from frame to frame.

2.1 Pre-Processing

The median filter is a classical noise removal filter. Noise is removed by calculating the median from all its box elements and stores the value to the central element. If we consider an example of 3x3 matrix

$$\begin{bmatrix} 1 & 2 & 9 \\ 4 & 3 & 8 \\ 5 & 6 & 7 \end{bmatrix}$$

The median filter sorts the elements in a given matrix and median value is assigned to the central pixel. Sorted elements 1, 2, 3, 4, 5, 6, 7, 8, 9 and median 5 will assign to the central element. Similar box scan is performed over the whole image and reduces noise. Execution time is more compared to mean filter, since the algorithm involves with sorting techniques. But it removes the small pixel noise.

2.2 Frame Difference

The system first analyses the images, being grabbed by the camera, for detection of any moving object. The Median Approximation algorithm is used for this purpose, which gives us output the position of the moving object in the image. This information is then used to extract a square image template (of fixed size) from that region of the image. The templates are generated as and when the appearance of the object changes significantly.

2.3 Dynamic Template Matching

The newly generated template is then passed on to tracking module, which starts tracking the object taking the template as the reference input. The module uses template-matching to search for the input template in the scene grabbed by the camera. If the object is lost while tracking (signifying that the object has changed its appearance) a new template is generated and used. Since the image templates, being used for matching, are generated dynamically the process is called Dynamic Template Matching [6], [7].
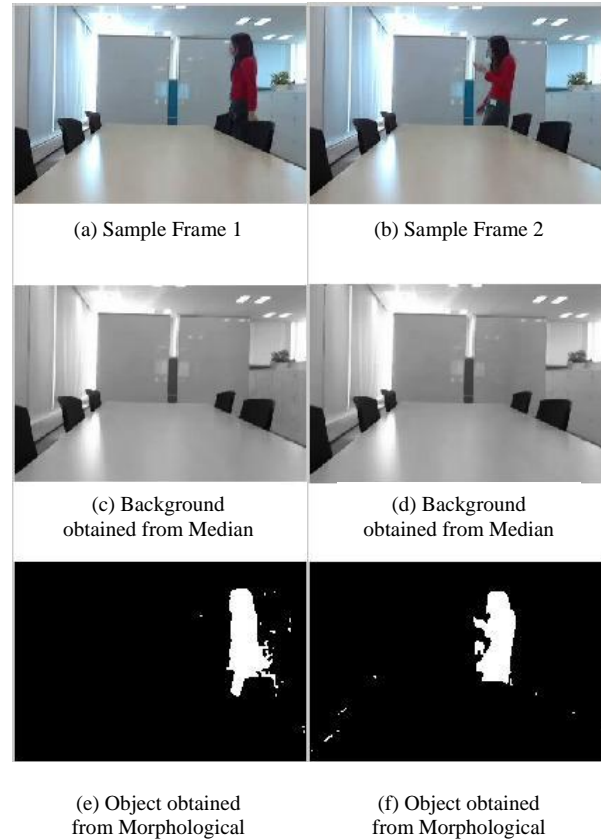

(a) Sample Frame 1   (b) Sample Frame 2

(c) Background obtained from Median   (d) Background obtained from Median

(e) Object obtained from Morphological   (f) Object obtained from Morphological

Fig. 1: Feature Extraction

2.4 Bounding Box

If the segmentation is performed using frame difference, the residual image is visualized with rectangular bounding box with the dimensions of the object produced from residual image. For a given image, a scan is performed where the intensity values of the image are more than limit (depends on the assigned value, for accurate assign maximum). In this Features is extracted by colour and here the intensity value describes the colour. The pixel values from the first hit of the intensity values from top, bottom, left and right are stored. By using this dimension values a rectangular bounding box is plotted within the limits of the values produced.

III. Algorithms

This section explains the algorithms used to track the single object.

3.1 Object Detection using Approximate Median Filtering

Median filtering has been shown to be very robust and to have performance comparable to higher complexity methods. However, storing and processing many frames of video (as is often required to track slower moving

objects) requires an often prohibitively large amount of memory. This can be alleviated somewhat by storing and processing frames at a rate lower than the frame rate, thereby lowering storage and computation requirements at the expense of a slower adapting background.

In median filtering the previous N frames of the video are buffered, and the background is calculated as the median of buffered frames [8], [9]. If a pixel in the current frame has a value larger than the corresponding background pixel, the background pixel is incremented by 1.Likewise, if the current pixel is less than the background pixel, the background is decremented by one. In this way, the background eventually converges to an estimate where half the input pixels are greater than the background, and half are less than the background approximately the median.

The Algorithm for Object Detection

1. Initially the first frame is considered as the background (bg).

2. For each pixel of the next input frame (fr)

Subtract the pixel intensity value from the background image.

Difference = fr-bg

IF(Difference> Threshold)

fg = bg

ELSE

fg = 0

3. For each pixel of the background

IF (fg > bg)

bg=bg+1

> *ELSE*

> *bg=bg-1*

4. Perform certain Morphological operations on the extracted image 'fg' to improve the image quality.

5. Calculate the Centroid (c1,c2) of the binary image fg. The result of this operation is a set of two integers which determine the position of the moving object in the given frame.

6. Use Kalman Filter to improve the accuracy of the obtained centroid values.

7. Get the next input frame and Goto (Step 2).

3.2 Kalman Filter

A Kalman filter is an optimal estimator i.e. infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive [10], [11],

[12]. If all noise is Gaussian, the Kalman filter minimises the mean square error of the estimated parameters. Kalman Filter is useful because

- Good results in practice due to optimality and structure.

- Convenient form for *online real time* processing.

- Easy to formulate and implement given a basic understanding.

- Measurement equations need not be inverted.

Equations used in Kalman Filter

1) STATE ESTIMATION:

Known are x(k|k), u(k), P(k|k) and the new measurement z(k+1).

1. State Prediction x(k +1|k):

$$F(k) x(k| k) + G(k) u(k)$$

2. Measurement Prediction:

$$z(k +1|k)= H(k)x(k +1|k)$$

3. Measurement Residual:

$$v(k +1)= z(k +1) - z(k +1|k)$$

4. Updated State Estimate:

$$x(k +1|k +1)= x(k +1|k) +W(k +1)v(k +1)$$

Where W(k+1) is called the Kalman Gain defined next in the state covariance estimation.

2) STATE COVARIANCE ESTIMATION:

1. State prediction covariance:

$$\mathbf{P}(k +1|k) = \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}(k)'+\mathbf{Q}(k)$$

2. Measurement prediction covariance:

$$\mathbf{S}(k +1) = \mathbf{H}(k +1)\mathbf{P}(k +1|k)\mathbf{H}(k +1)'+\mathbf{R}(k +1)$$

3. Filter Gain:

$$\mathbf{W}(k +1) = \mathbf{P}(k +1|k)\mathbf{H}(k +1)' \mathbf{S}(k +1)\text{-}\mathbf{1}$$

4. Updated state covariance:

$$\mathbf{P}(k +1|k +1) = \mathbf{P}(k +1|k) -\mathbf{W}(k +1)$$
$$\mathbf{S}(k +1)\mathbf{W}(k +1)'$$

### 3.3 Object Tracking using Dynamic Template Matching

The aim of our object tracker module is to generate the trajectory of an object over time by locating its position in every frame of the video. The possible object region in every frame is obtained by means of our object

detection module, and then our tracker module corresponds the object across frames. In our tracking approach the object is represented using the Primitive Geometric Shape Appearance Model (i.e. the object is represented as a rectangle). The limitation of the approximate median filter algorithm in not being able to detect stationary objects is the reason why the template matching process must be coupled with it to create a good tracking system.

The object tracking module takes the positional information C(x1, x2) of the moving object as an input from the object detection module. This information is then used to extract a rectangular image template (whenever required) from the last acquired frame. The module keeps on searching it in the frames captured from that point. Whenever found it displays a Yellow overlaid rectangle over the detected object. If the template matching doesn't yield any result (signifying that the object has changed its appearance) a new template is generated and used. As the templates for searching are being generated dynamically the object tracking module is said to be employing the dynamic template matching algorithm.

Algorithm for the Template Matching

1. Get the positional information C (x1, x2) from the Object Detection Module.

2. Generate an image template T by extracting a rectangular image from the last frame input from the camera. The Template is extracted using the coordinates given by

(x1-30,x2-40)   (x1+30,x2-40)

(x1+30,x2-40)   (x1+30,x2+40)
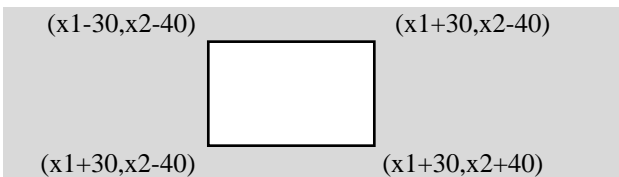
3. Search the generated template Ti in the last frame grabbed by the camera. This is done by using an efficient template matching algorithm.

4. IF   the template matching is successful

          THEN

IF the tracker has NOT detected motion of the object

AND the detector has

          THEN goto STEP 1 (get a new template)

      ELSE goto STEP 5 (get the x, y position)

    ELSE goto STEP 1 (get a new template)
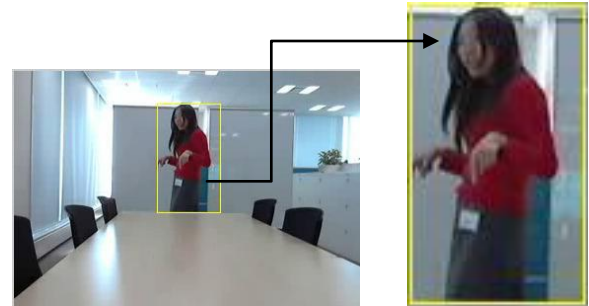
5. Goto (Step 3).

Fig. 2: Template generated dynamically based on the Centroid information of the moving object

3.4 Visualization of the Features Extracted

A Bounding Box is used to visualize the features extracted.

Algorithm for the bounding box is as followed

1. Read the image difference

2. For (present position=initial value: final value) of Y resolution

3. For (present position=initial value: final value) of X resolution

    (a) Calculate the sharp change in intensity of image from top and bottom

     (b) Store the values in an array

4. Height of the bounding box is = bottom value- top value

5. For (present position=initial value: final value) of X resolution

6. For (present position=initial value: final value) of Y resolution

(a) Calculate the sharp change in intensity of image from left and right

(b) Store the values in an array

7. Width of the bound box = right value - left value

8. Using the dimensions, draw boundary to the image

    Initial value: the starting position of the pixel in an image. Final value: the ending position of the pixel in an image.

    Height = (bottom value –top value)/2

    Width = (rightvalue –leftvalue)/2

9. Add the height value with the top value and store it in a variable like mid top

10. Add the width value to the left value and store it in a variable like mid left

11. Assign the max intensity to the pixel at pixel value at (mid top, mid left)

IV. Implementation and Evaluation

This section explains the implementation and evaluation of algorithms for single object tracking. The step by step process is shown graphically in Fig. 3. Algorithms for the single object tracking are developed in MATLAB. The task is performed by recording a video using digital camera with 30fps. The implementation is initially performed on MATLAB and various methods for object tracking are tested. Among the tested methods Approximate Median Filter is used for further calculations.

4.1 Image Processing Toolbox

The image processing toolbox [13] in MATLAB has several useful functions for processing of images and videos. Some of the important Image Processing functions used in the algorithm are:

MEDFILT2()   2-D median filtering

B = MEDFILT2(A,[M N]) performs median filtering of the matrix A in two dimensions. Each output pixel contains the median value in the M-by-N neighbourhood around the corresponding pixel in the input image. MEDFILT2 pads the image with zeros on the edges, so the median values for the points within [M N]/2 of the edges may appear distorted.

RGB2GRAY()

I = rgb2gray(RGB) converts the true-colour image RGB to the grayscale intensity image I. rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

BWMORPH()

BW2 = bwmorph(BW,operation) applies a specific morphological operation to the binary image BW.

REGIONPROPS()

STATS = regionprops(BW, properties) measures a set of properties for each connected component (object) in the binary image, BW. The image BW is a logical array; it can have any dimension. This is used to extract Area, Centroid, Bounding Box values of the extracted binary images.

4.2 Conversion

Conversion explains the functions that are performed to convert the recorded video into input format of image for MATLAB. The steps are:

- Video to images (true colour)
- Images to grayscale intensity images

1) *Video to Images*

The task of the object tracking is performed by a video recorded on a webcam at 160x120 frame size. The lower frame size of the video is to

- Less memory can be used for calculations
- Change in position of object can be easily observed in each frame

The converted video frames is now spitted into individual frames and saved as true colour RGB format.

2) *RGB Image to Grayscale Intensity Images*

The images are converted into grayscale intensity images. This is performed using RGB2GRAY function available in the image processing toolbox of MATLAB.

4.3 Noise Removal

Noise removal is used to improve the image quality by reducing noise. Noise is reduced by filter technique and by adjusting the threshold value. Median filter algorithm is applied as it removes the salt pepper noise. The median filter is very good at removing short noise (but not perfect).

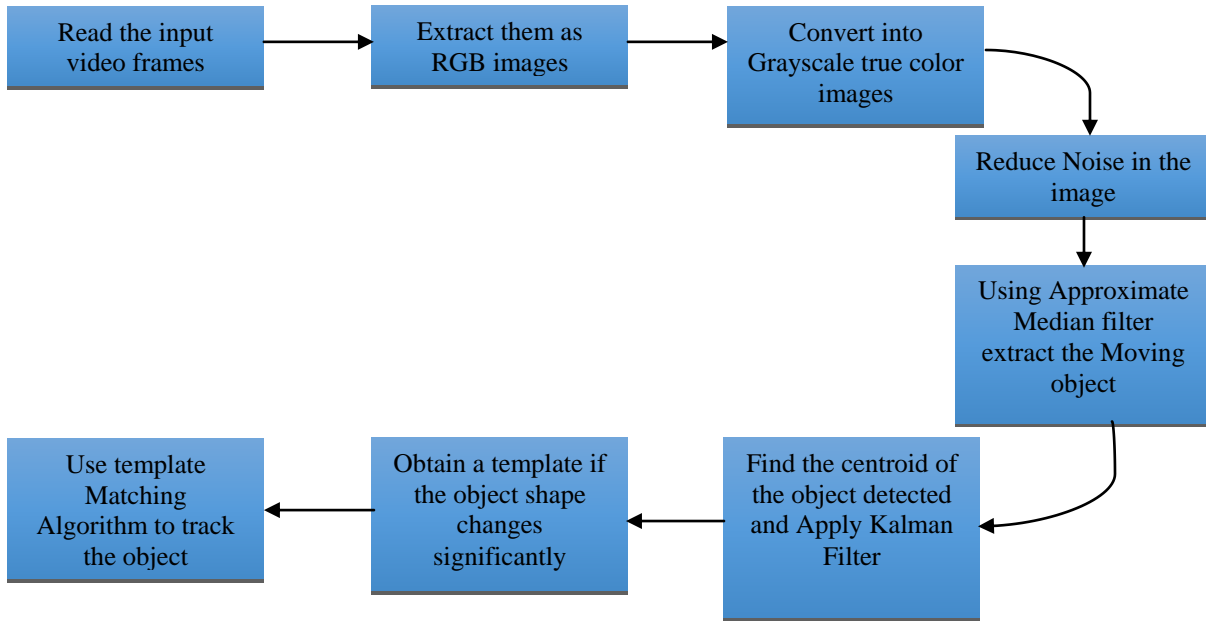Fig. 4: Tracking of a Person walking in a room

```
┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│ Read the input│───▶│ Extract them as│──▶│ Convert into  │
│ video frames  │     │  RGB images   │     │Grayscale true color│
└──────────────┘     └──────────────┘     │    images     │
                                           └──────────────┘
                                                   │
                                                   ▼
                                           ┌──────────────┐
                                           │Reduce Noise in the│
                                           │    image      │
                                           └──────────────┘
                                                   │
                                                   ▼
                                           ┌──────────────┐
                                           │Using Approximate│
                                           │ Median filter │
                                           │extract the Moving│
                                           │    object     │
                                           └──────────────┘
┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│ Use template │◀───│Obtain a template if│◀─│Find the centroid of│
│  Matching    │     │ the object shape│   │the object detected│
│Algorithm to track│ │   changes     │   │and Apply Kalman│
│  the object  │     │ significantly │   │    Filter     │
└──────────────┘     └──────────────┘     └──────────────┘
```

Fig. 3: Block Diagram of the implementation and evaluation

4.4 Object Detection

The experiment involved the tracking of a single person walking inside a room. The results of the experiment are presented in Fig. 4, the yellow square represents the result of the template matching module and has been superimposed by the tracking module on the original frame.

4.5 Feature Extraction

Feature Extraction visualizes the moving object in sequence of frames and a rectangle bounding box is

drawn using its dimensions. Section II explained a clear view on bounding box. Bounding box algorithm described in section III plots the bounding box to the residual image from the frame difference. When the algorithm is implemented for the image, dimensions of the image are produced and a rectangular boundary is plotted. By using this bounding box dimensions, we plot the centroid of the box and used for tracking of objects.
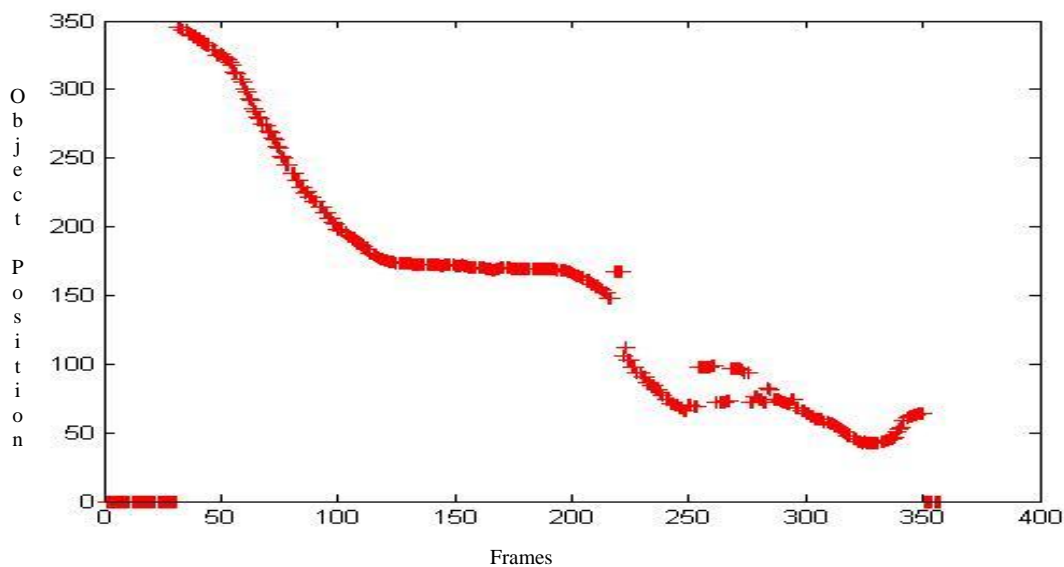

Fig. 5: Plot of the centroid of the person walking in the video observed

From the above Fig. 5, it can be observed that the video tracker is able to track every aspect of the object

of interest. Even slight changes in the movement of hands can be tracked. The plot of the tracking shown in

Fig.5 indicates that the tracking is done with a great deal of accuracy. From frames 100 through 200 the video tracker has been able to track the object even when it is stationary. This has been achieved by incorporating the template matching algorithm.

V. Conclusion and Future Scope

A robust and efficient automated single object tracking system is presented in this paper. The system has been implemented using an algorithm based on Approximate Median Filter, Kalman filter and dynamic template matching. The algorithm has experimentally been shown to be quite accurate and effective in detecting a single moving object even under bad lighting conditions or occlusions. Such an automated object tracking system can be used in applications where accurate tracking is required but good lighting conditions cannot be provided. The video tracking system is furthermore very much applicable to areas like video conferencing and surveillance. Future work focuses on tracking multiple objects at the same time as well as on improving tracker accuracy during camera motion. The algorithms can be implemented on hardware equipment. These algorithms can be further extended for the use in real-time applications and object classifications.

Acknowledgments

References

[1] Karan Gupta and Anjali V. Kulkarni, Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching. SCSS (1) 2007: 245-250

[2] Alsaqre F.E., Baomng Y., "Multiple Moving Objects Tracking For Video Surveillance Systems",ICSP'O4 Proceedings, Volume 13 no.04, pp.1301-1305, 2004.

[3] F. Jean, R. Bergevin, A.B. Albu, "Body tracking in human walk from monocular video sequences," Proc. of the 2nd Canadian Conf. on Computer and Robot Vision, pp. 144-151, May 2005.

[4] Haritaoglu I., Harwood D. and Davis L.S., "W4: Real-Time Surveillance of people and their Activities", Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22, no. 8, pp. 809-830, 2000

[5] Young Min Kim," Object Tracking in a Video Sequence",2007

[6] Aree Ali Mohammed, Astrid Laubenheimer Yusra and Ahmed Salih, Efficient Face Tracking and Detection in Video: Based on Template Matching, ht tp://elrond.informatik.tufreiberg.de/papers/WorldComp2012/IPC2078.pdf

[7] P. Sanyal et al., International Journal of Computer Science and Mobile Computing Vol.1, December-2012, pg. 1-5

[8] Sivabalakrishnan.M and Dr.D.Manjula," An Efficient Foreground Detection Algorithm for Visual Surveillance System "IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.5, May 2009

[9] N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications 8(3), pp. 187-193, 1995.

[10] G. Welch and G. Bishop. An Introduction to the Kalman Filter, Proceedings of SIGGRAPH 2001, pp 19-24.

[11] LIU Zhong-pu, Shu-guang Zhao and PAN Xiang-he, "A New Method of Motion Object Tracking Based on Kalman Prediction and Mean-shift Search "Optoelectronic Technology, Vol.29, pp. 30-33,2009(in Chinese).

[12] WENG Shiuh-ku, KUO Chung-ming and TU Shu-kang, "Video Object Tracking Using Adaptive Kalman Filter," Journal of Visual Communication and Image Representation(S1190-1208),Vol.17, pp. 1194-1197, 2006

[13] http://www.mathworks.in/products/image/

Authors' Profiles

G.Mallikarjuna Rao is currently working as scientist in Research Centre Imarat (RCI), Defense Research Development Organisation, India. He has 14 years of experience. His research interests include Target Tracking, Image processing, Pattern recognition and Signal Processing. He guided several M.Tech and B.Tech Projects.

Dr. Ch. Satyanarayana is Professor in Computer science and Engineering Department at Jawaharlal Nehru Technological University Kakinada, Kakinada. He has 13 years of experience. His area of interest is on Image processing, Database Management Systems, Speech Recognition, Pattern recognition and network security. He published more than 25 research papers in international journals. He published more than 30 research papers in international conferences.