IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Surveillance Using CAMshift Tracking

Prof. S.O. Dahad[1], Rupali R Sonawane[2]

E &Tc Department, Government College of Engineering, Jalgaon, India[1]

E &TcDepartment, Government College of Engineering, Jalgaon, India[2]

Abstract: The aim of this system is to track a person of interest in real time based on scenes obtained from camera modules. This system avoids potential human error as we are using automatic monitoring. It is a histogram based real time tracking system. Moving objects are detected using background subtraction processing. Moving object detection, recognition and tracking are basic steps in processing.

Keywords: Tracking, surveillance, CAMshift, camera.

I. INTRODUCTION

Public security has become very important issue due to the increased terrorist attacks, burglaries etc. Electronic surveillance forms an integral part of today's security systems. CCTV modules are sold in huge numbers across the globe. Primitive surveillance systems [1] involved cameras. All these cameras would feed the live video to a dedicated monitor and in a control room many such monitors would be stacked one upon the other. These systems can be found in departmental stores, museum to high security areas like railway stations, airports and banks.

Fig.1. Applications of camera surveillance systems

The main objective of this project is to develop a system which reliably detects the person's presence or absence in a locality.The first part of the paper provides information about the algorithm selection and camera selection.

II. LITERATURE REVIEW

In [2] a method to track any object which is differ from the background was proposed. Multiple objects gets consistently tagged and hence it is a problem for selectingan object of interest.In [3] a method of tracking an aerial objects was discussed. The given image is divided into blocks and these blocks are adaptively changing their size according to template size. In [4,5] authors has discussed motion segmentation computation and also optical flow. It creates a difficulty when object is detected at start. In [6] a real time tracking and adaptive framework was done. In [7] partial filtering was discussed. In [8] author discussed mean shift algorithm which is suitable for multiple objects tracking. In [9] it showed that these algorithms are computationally complex. In [10] CAMshift algorithm was discussed which simple and efficient algorithm for tracking. In [11] location of the object could be tracked from probability distribution of an image. In [12] the author proposed an algorithm which processes all the pixels in a given frame.

III.SYSTEM ARCHITECTURE

The application scenario would include an enclosed building where people enter one at a time. The building is equipped with many interconnected camera modules. Each of the entrances is to be provided with a camera system such that each entrant is registered by taking his high resolution photograph. These images of people are stored in a central storage. Whenever a query for a person is requested in terms of photograph, the supplied image is compared with the ones in the database. After verifying the person's presence, the feature related data of the subject is communicated to all the camera modules for tracking. The status of the tracked person is conveyed to the security official via a display terminal as shown in Fig.1.

- Camera modules

The camera modules are located at various locations in an enclosed place or a building. As a high resolution image would require larger computations while a low resolution image would degrade the video quality. Hence, an optimum resolution of 640x480 is a good trade off. The camera must present the data in an easily readable RGB for further processing on the host machine.

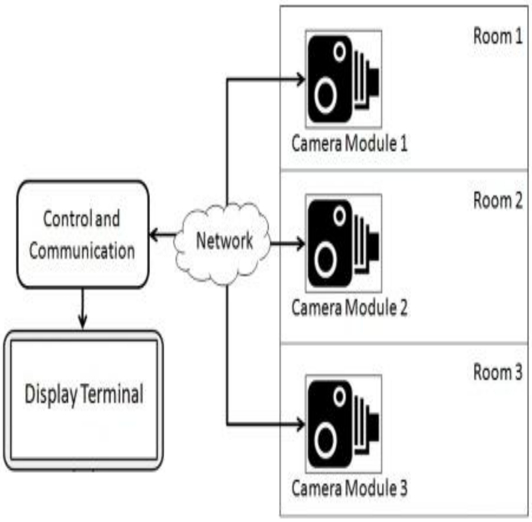

Fig.2. Block Diagram

IV. PROPOSED TRACKING ALGORITHM

The image captured from the camera is obtained in RGB (Red-Green-Blue) format. However, this format is not suitable for color based tracking as R, G and B components are highly correlated. Hence the RGB image is converted to HSV (Hue-Saturation-Value) color space. Object tracking using HSV is achieved by extracting H, S and V planes of the raw image and using all of them for further processing. The approach is expensive for an embedded platform. This problem can be overcome by considering only H plane. However for a grayscale image, H component does not convey any meaningful information and hence cannot be used to track a grayscale object. This issue can be addressed by treating grayscale object separately and using V component to process such images. This requires distinguishing between color and grayscale objects, which is done using the S component. The subsequent processing occurs on H for color targets and V for grayscale images. The remainder of this section contains the details of the propose method.

A. Division of image into blocks

Prevalent systems apply CAMShift on every pixel of the chosen region. To reduce the amount of computations, the frame is divided into certain number of blocks. The size of a single block is limited by the following constraints.

- It cannot be so large that the object size is insignificant
- It cannot be so small that the computational overhead exceeds the per pixel computations

For optimum performance, block size should be equal to the size of the object when the object is at the farthest end from the camera, where it can be practically tracked. Thus the block size is chosen depending upon the size of the object and its distance from the camera. In this implementation for a 640× 480 pixel frame, a block size of 32 ×24 pixels is chosen to track a person's shirt at a distance ranging from 1 m to 10 m.

B. Modified CAMshift Algorithm

Consider an image IRGB, the input frame in which the object is to be searched and tracked. Let TRGB represent the object image, that needs to be searched. Each pixel in the image IRGB is indexed as (x; y). As discussed in previous sections, processing needs to be done only on H image for color targets and V images for grayscale targets. For a color target, let T and I represent the H planes while for grayscale targets, T and I represent V planes.

1) Background Subtraction (BGS):

BGS is used to remove the effect of static background objects while tracking. This is the first step in the algorithm, where each camera module captures an image in the absence of the object to be tracked. This image forms the reference, that is subtracted from every subsequent frame. The subtraction is done only for H plane in case of color targets and V plane in case of grayscale targets. If B is the reference background frame and I is the newly acquired frame, then the difference image D is given as follows.

$$D(x,y) = \begin{cases} 1 & \text{if } I(x,y) - B(x,y) > Dth \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

whereDth is the threshold to detect if a pixel has changed. The difference image D obtained must have zero value at all pixels except where foreground objects are positioned. However, even with a static background, illumination and sensor noise may affect some of the pixel values.

The image D is divided into M×N blocks, each of size BW×BH and stored in a matrix DB.

$$DB(m,n) = \begin{cases} 1 \text{ if } \left(\sum_{xl}^{xu} \sum_{yl}^{yu} D(x,y)\right) > DBth1 \\ 0 \quad \text{otherwise} \end{cases} \quad (2)$$

Where,
xl=m*BW+1,xu=(m+1)*BW
yl=n*BH+1;yu=(n+1)*BH

The background may change significantly due to movement of objects and variation in illumination. Hence the background image needs to be updated periodically. The background cannot be updated on the camera module that is currently tracking the object, as the object may appear merged with the background in the subsequent frames. Therefore it is done only when the camera module is free from the tracking task.

2) Histogram Matching:

The histograms of these blocks are computed and compared with the target image histogram. Each block is assigned a value equal to the bin-wise Sum of Absolute Differences (SAD) of its histogram and the target histogram. This results in blocks that match closely with the target histogram having lower numerical values. Further, the reciprocal of SAD values is calculated and normalized. This is done to satisfy the requirement of the CAMShift algorithm that locates the region with higher numerical values. Let I_B be the block wise representation of the image I. The histogram of I_B is computed on per

DOI 10.17148/IJARCCE.2016.55226

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

block basis for Pbins. The histogram values $I_h(m;\ n)$ for a block (m; n) are as follows:

$$I_h(m,n)=HISTOGRAM(I_B(m,n)) \qquad (3)$$

Taking the reciprocal of the difference of target image histogram T_h and that of histogram of blocks of I_B result in a new image F of size M×N.

$$F(m,n)=\frac{1}{\sum_{i=0}^{hn} |Ihi\ (m,n)-Thi\ |} \quad (4)$$

Ih(i)(m,n) and Th(i) are the i^{th} bins of histograms
The matrix F is normalized by dividing each element by the maximum value among the elements of matrix F.

$$Fnorm(m,n)=\frac{F(m,n)}{Fmax} \qquad (5)$$

The end result is a matrix Fn containing the M×N normalized values of F, corresponding to each block in the image I.

3) CAMShift using block values:
Applying CAMShift algorithm to the region of interest in the frame involves computations on each pixel of the region. Hence, it is proposed to use a single value per block computed in the previous step to represent the whole block. This substitution drastically reduces the calculations in CAMShift algorithm without any significant change in results. CAMShift algorithm is applied to the matrix F_n as follows.
The 0^{th} moment and the 1^{st} moments of the probability distribution image F_n are calculated as follows

$$M00=\sum_{m=0}^{M}\ \sum_{n=0}^{N} Fn(m,n)*BW*BH$$

$$M01=\sum_{m=0}^{M}\ \sum_{0}^{N} m*Fn(m,n)*BW*BH$$

(6)

$$M10=\sum_{m=0}^{M}\ \sum_{0}^{N} n*Fn(m,n)*BW*BH$$

Using the results from Eqn.(6), the location and the size ofthe object are calculated as follows:

$$xcen=\frac{M01}{M00}$$

$$ycen=\frac{M10}{M00} \qquad (7)$$

$$xsize=\sqrt{M00}$$

The values obtained from Eqn.(7) are used to mark the object in the frame.By adopting block division based approach, the valuescomputed in Eqn.(6) and Eqn.(7) vary as compared to thestandard CAMShift algorithm. This slight variation in thecalculated size and position of the object is the compromise for the reduction in the number of computations.

V. RESULTS

The Fig.3 shows the green color tracking and tracking mark move towards the area where green color density is more.

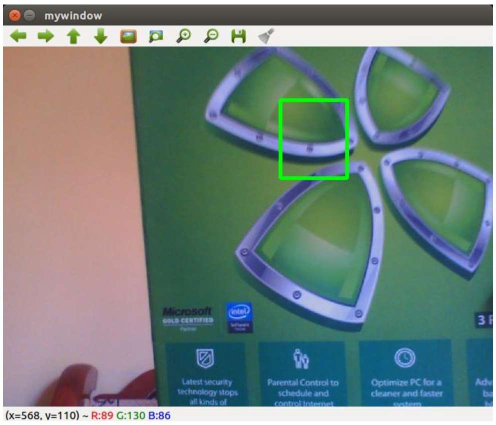

Fig..3Green Color Tracking

VI. CONCLUSION

In this paper, I have proposed a modified CAMShift algorithm. The blockdivision approach adopted here reduces the computations to agreat extent without significant loss in the quality of tracking,as validated by experiments. The results show that these solutions hold a great potential in real-time object tracking applications.

REFERENCES

[1] T.D. Raty. Survey on contemporary remote surveillance systems for public safety. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40(5):493–515, 2010."
[2] Y. Wang, M. Casares, and S. Velipasalar, "Cooperative object tracking and event detection with wireless smart cameras," in Advanced Video and Signal Based Surveillance, 2009.AVSS '09. Sixth IEEE International Conference on, 2009, pp. 394–399.
[3] S. Sahani, G. Adhikari, and B. Das, "A fast template matching algorithm for aerial object tracking," in Image Information Processing (ICIIP), 2011 International Conference on, 2011, pp. 1–6.
[4] Jung Uk Cho, Seung-Hun Jin, Xuan Dai Pham, and Jae WookJeon. Object tracking circuit using particle filter with multiple features. In SICE-ICASE, 2006. International Joint Conference, pages 1431–1436, 2006.
[5] S. Denman, C. Fookes, and S. Sridharan.Improved simultaneous computation of motion detection and optical flow for object tracking. In Digital Image Computing: Techniques and Applicat ions, 2009. DICTA '09., pages 175–182, 2009.
[6] X. Chen, X. Li, H. Wu, and T. Qiu, "Real-time object tracking via CAMshift-based robust framework," in Informat ion Science and Technology (ICIST), 2012 International Conference on, 2012, pp. 527–530.
[7] J. U. Cho, S.-H. Jin, X. D. Pham, and J. W. Jeon, "Object tracking circuit using particle filter with multiple features," in SICE-ICASE, 2006. International Joint Conference, 2006, pp. 1431 1436.
[8] B. Li, Z. yuanZeng, and Z. ru Wu, "Multi-object tracking based on improved mean-shift algorithm," in Image and Signal Processing, 2009. CISP '09. 2nd International Congress on, 2009, pp. 1–5.
[9] J. U. Cho, S.-H. Jin, X. D. Pham, J. W. Jeon, J.-E.Byun, and H. Kang, "A real-timeobject tracking system using a part icle filter," in Intelligent Robots and Systems, 2006IEEE/RSJ International Conference on, 2006, pp. 2822–2827.
[10] G. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in Applications of Computer Vision, 1998.WACV '98.Proceedings. FourthIEEE Workshop on, 1998, pp. 214–219.
[11] X. Qin-lan, "A real-time motion target detecting and tracking system based on tms320dm642," in Intelligent Systems, 2009. GCIS '09. WRI Global Congress on, vol. 3, 2009, pp. 83–87.
[12] X. Gang, C. Yong, C. Jiu-jun, and G. Fei, "Automatic CAMshift tracking algorithm based on fuzzy inference background difference combining with twice searching," in EHealth Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on, vol. 1, 2010,pp. 1–4.