International Journal of Control and Automation
Vol. 13, No. 2, (2020), pp. 868 - 876

Proximity Approach for Object Detection in Video

Nilesh Uke, Member IEEE
Professor, Department of Computer Engineering
Trinity Academy of Engineering, Pune, India ,

Shailaja Uke
Assistant Professor, Department of Computer Engineering
SKN SITS, Lonavala, India ,

Abstract

Last decade we are experiencing more applications in video surveillance to address issues related to social needs. As public concern about crime and terrorist activity increases, the importance of public security is growing, and video surveillance systems are increasingly widespread tools for monitoring, management, and law enforcement in public areas. Object detection is a primary concern about all of these applications domains.

In this paper, we exploit computer vision methods to detect moving object from video to track in real time as objects encountered in the indoor and outdoor environment. Proximity is a fact of being near to other and justifies closeness. These concepts of object being close to each other is checked while the process of object tracking. System tracks assorted objects against an environment consisting of objects of varying sizes, shapes and colors. Initially background modeling is performed using the function which accumulated the background frames from mean and standard deviation of first N frames. Each significant change in the object appearance thereafter, due to new object, old object disappearance is tracked based on the proximity of the target object. The visual resemblance is determined with respect to the detected object in the previous video frame and the last frame captures.

Keywords: Moving Object, Object Tracking, Proximity, Gestalt Law,

1. Introduction

Methods for extracting moving objects from videos are being studied extensively by many researchers due to its wide verity of applications. Once the moving object detected; it is being used in many application which includes measuring vehicle traffic [1], motion tracking [2],[3], traffic sign recognition [4]–[6], pedestrian detection [7], [8] , face and logo detection [9]–[11], and drivers drowsiness detection [12]. But in recent years, due to increased demand of intelligent systems and more challenging real world scenes made systems to be more robust to noise in data, abrupt motion or illumination variation, non-rigid or articulated movement of objects, background variation etc. The main difficulty to solve tracking problem is to find correspondence of the same moving objects in different frames of the video. This problem may solved by looking at several aspects of the scene, such as the density and proximity of objects, variable shapes, presence of occlusions etc. The problem is further complicated by several factors such as camera quivering, flawed calibration of the on-board cameras, complex environments, and so on [13].

2. Background Study

Most recently, many research related to visual tracking is being carried out. Stereo vision-based model for multi-object detection and tracking is proposed for surveillance systems [14]. Computer Vision methods and deep convolutional neural networks (CNNs) are seemingly combined in DEP-SEE framework [15] to exploits to detect, track and recognize in real time moving objects observed during moving in the outdoor environment. Earlier we proposed hybrid method of object detection using motion estimation and tracking by parallel Kalman filter [16]. A system [2] is proposes with a unique object detection and tracking system where video segmentation, feature extraction, object detection and tracking are combined perfectly using various features.

2.1 Visual Perception and Proximity

ISSN: 2005-4297 IJCA
Copyright ⓒ 2020 SERSC

A lot of work has been done on investigation and implementation of Gestalt principles for visual perception. Perceptual grouping uses Gestalt principles [17] to group visual features together to meaningful parts (objects). The Gestalt Laws are descriptive principles in Gestalt psychology that specify the way in which the human brain performs perceptual grouping as shown in Fig 1. They have found extensive application, not only in computer vision, but as guiding principles in visual interface design and design of education material.

Figure 1: Gestalt principles of proximity, good continuation, and similarity

Once the image has been grouped into similar elements according to the above grouping principles, the human process of object recognition and classification begins. One common way to determine proximity is to measure absolute locations and compute distances. Proximity is a fact of being near to other and justifies closeness. These concepts of object being close to each other is checked while the process of object tracking. The ability to identify an object as the same individual across a period of occlusion can rely on several perceptual and cognitive processes, especially when only a single object is involved [18]. Spatial proximity and motion coherence is used in [19] where residual feature tracks are clustered into independently moving entities. Earlier work in [20] also describes the use of proximity for detecting of moving object in real time environments.

3. System Description

This systems aim at achieving real-time object detection and tracking of multiple objects, operational in any unknown background. The basic flow of the tracking system is illustrated in Fig 2. The input to the system is real-time video captured from a standard web-cam or any camera installed at fixed location (stationary camera). Two main processes are implemented in this system are extracting background model and updating incoming frame.

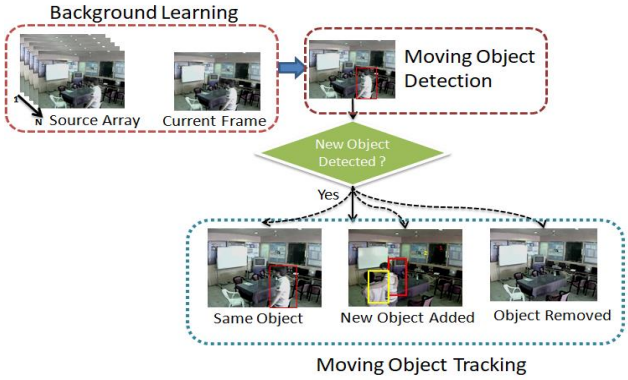

Figure 2: Basic Flow of Object Detection and Tracking

The first process used a simple technique to extract initial background model from first *N* sequence taken from the camera sensors. The second process continuously updates the incoming frame for moving object depending upon the proximity analyzes is to check whether he object is old, new or removed from the frame sequence with respect to be background. A search strategy for finding the most likely location of the object in the current frame depends on the proximity analysis.

3.1 Developing a Background Model (Background Learning)

Background subtraction is a most common method for discriminating moving objects where camera is fixed [21]. Basically, a pixel-wise reference model for the stationary part of the scene is estimated. Then, the observed image is compared with this reference to obtain a foreground mask. For a single fixed camera no extra processing is necessary. In case of multiple cameras, we get inputs from more than one source.

ISSN: 2005-4297 IJCA
Copyright ⓒ 2020 SERSC

In order to develop a background model in this work, we consider first N frames (100 frames) of the video. We have assumed that SourceArr is an array of k consecutive frames and accFrames is a structure of a 32-bit Image used to store pixel level. Therefore computing mean deviation by providing x and y coordinates of the image is simpler. If 'c' is the channel of the RGB image then accumulated frames represents the mean of the 'n' images by Equation 1.

$$\forall x, y, c \; accFrames[x, y, c] = \left(\sum_{i=1}^{n} sourceArr[i][x, y, c] \right)/n \quad \text{..... Eq 1}$$

Similarly, lets describe represents the standard deviation of the images from their mean by accDiff by following Equation 2.

$$\forall x, y, c \; accDiff[x, y, c] = \left(\sum_{i=1}^{n} |(sourceArr[i][x, y, c] - accFrames[x, y, c])| \right)/n \quad \text{..... Eq 2}$$

Algorithm 1 – createAccumulatedImage

Input: int ICount
Initialize: SourceArr[100], accFrames - 32bit IplImage, accDiff - 32bit image, stdDev - 8bit image, Icount=100
For i = 0 to Icount **Do** //learnt the mean of the 100 images
- Create a 32 bit image to store the captured image
- Query for a new image 8 bit
- Convert the queried 8 bit image to 32 bit and store it in sourceArr[i]
- Add the converted image to the accumulator

End for //At the end of this loop the image accFrames holds the sum of data for 100 frames
- to divide every pixel value by Icount(100) to get the mean value from the accumulated sum
- finding the average of all frames / mean

For i= 1 to Icount Do // loop subtracts the mean (accFrames) of the 100 images
- from each image and then adds the resulting difference onto a difference accumulator
- cvAbsDiff puts the difference between the mean
- Adding the difference on the difference accumulator accDiff
- accDiff now holds the sum of the absolute difference of all the images from their mean

End for
- to divide every pixel value by Icount(which is 100) to get the mean of the deviation
- now convert the accDiff back to an 8 bit image

Output: Learn background from past 100 frames

The procedure for maintaining the background for object detection process is given in the following algorithm. Once we know the background, extracting the foreground is simple matter.

3.2 Object Detection

Main aim of background modeling is to segment regions corresponding to moving objects such as vehicles and humans from the rest of an image. Detecting moving regions provides a focus of attention for later processes such as object tracking. Object detection is the task of localization of objects in an input image. The definition of an "object" varies. It can be a single instance or a whole class of the object.

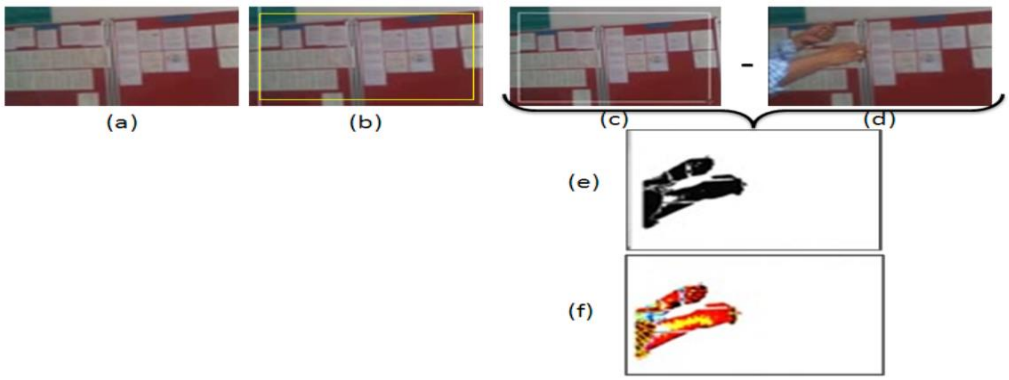

Figure 3: The result of background subtraction (a) Original Image (b) Selecting Region (c) Learned Background (d) New object in frame (e) Result of c-d (f) object detected

Copyright ⓒ 2020 SERSC

We use technique based on the background subtraction, since the camera is static and fixed at certain place, as in bank, parking or ATMs. Object detection can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Foreground pixels are detected by using the background model and the current queried image from video. Then it subtracts the intensity value of each pixel in the current frame from the corresponding value in the reference background model as shown in Fig 3. Any significant change in an image region from the background model signifies a moving object.

3.3 Object Tracking

The object of interest once detected and recognized can be easily tracked. Tracking involves tracing the path followed by the object of interest. Object tracking is the process of locating an object in the image plane, where it moves around the scene and is still a challenging task. The object-tracking algorithm keeps an internal list of tracked objects. For each frame, these objects are compared with the detected blobs and if they correspond, they are updated with information of these blobs. For each frame difference image is calculated using Eqn. 3. Threshold in this equation ranges from 0 to 100, which helps us to detect large or small object's contours. Ideally if differenceThresh is chosen in between 30 to 40, too small moving objects are neglected

$$diffImage[x,y,c] = \begin{cases} 255 & (|source[x,y,c] - meanImg[x,y,c]| - stdDiff[x,y,c]) > differenceThresh \\ 0 & otherwise \end{cases} \quad \text{..... Eq 3.}$$

3.3.1 Bounding box for tracking

Commonly used method to draw attention of tracked objects in surveillance videos for display purposes is bounding box. This bounding box of a foreground object can be defined as the smallest rectangle that can contain the object. The height and width of this rectangle are found by determining the pixels in the foreground object that are at the upper, lower, left, and right extremes of the object's contour. The offset between extremity pixels coordinates in the x and y directions can be used to determine the smallest rectangle that contains the object as seen in Fig 4. The height and width of this rectangle are then extracted as features used for object classification. Using the area of the foreground object as well as the height and width of the bounding box containing the object, it is easy for further tracking.

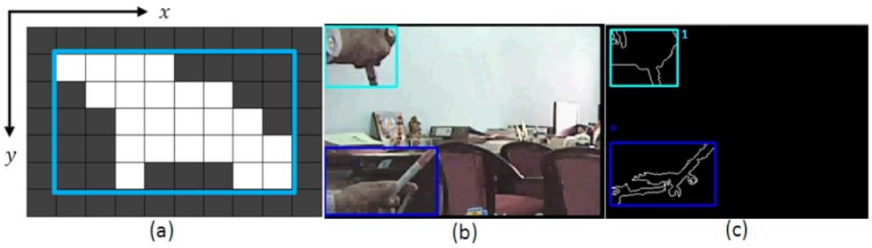

Figure 4: (a) Bounding box of foreground object (b) Original frames with objects (c) Box with active contours

Every new frame is checked for any new objects appearance or disappearance. Following three conditions are observed:-

Old Object Detected: Once the first foreground moving objects is detected, there is a constant observation on the new frame for new objects. First scenario when number of objects are equal in previous and current frame i.e., shapeCountCurrent = shapeCountPrevious. For each object in shapeArrayCurrent find the index of the object with closest proximity from previous shape array. This object is found by calling a findNearest

Copyright ⓒ 2020 SERSC

function, which returns the closest object. Swapping of previous array with current shape array is performed and shape count is set to current count.

New object Detected: In the frame, the new object can only appear at the border region of image. Scenario when number of objects in current frame are greater than in previous frame i.e., shapeCountCurrent >shapeCountPrevious. For each object in previous shape array, find the object with closest proximity in current shape array by calling findNearest function. New shape index and color index is provided to the newly added object. Shape count is updated for further processing.

Old Object Removed: The disappearance of the old object only appears at the image border region. Assuming that one object moves near the border in the previous frame, and will disappear in the next frame, and then the similar approach is used. Scenario when number of objects in current frame are lesser than in previous frame i.e., shapeCountCurrent < shapeCountPrevious,

3.3.2 Finding nearest Object

Every object appearing in input frame is defined by a structure. A Shape Index points to a particular shape in the previous array; where as findNearest() loops over all the shapes structures in the current array and searches for the one that is closest to the shape pointed by shapeIndex. In the current example in Fig 5 the function would return the index value of 4 which is a shape in the current array that is closest to the one in the previous array.

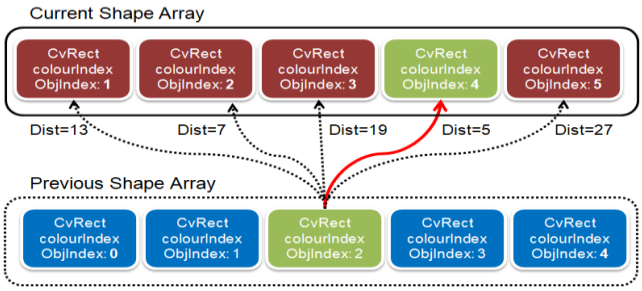

Figure 5: Finding Nearest Object (Similarity)

To match the stored and new objects every new object was compared to every stored object. Although this method is relatively slow, it produced the most consistent results when compared to other explored methods.

4.0 Experimental Results and Analysis

To evaluate the tracking performance for real images, we performed experiments with a sequence grabbed from the stationary cameras in the desktop environment are categorized in single-object indoor, multi-object indoor, single-object outdoor and multi-object outdoor environment. The proposed system for object detection and tracking in unknown environment was tested to operate in complex, real world indoor and outdoor environments. The OpenCV library was employed for the realization of the proposed system, whilst the required hardware is restricted to a single stationary camera. Images of resolution 640 × 480 were used and the target objects considered includes not just human and moving vehicles, but also includes book, animal, ball, etc. for both indoor and outdoor environments.

4.1 Single Object - Indoor Environment

Our tracked object varies in size from frame to frame and from video to video, in which the smallest size is 30 × 6, and the largest size is 341 × 365. Generally, a bigger tracked object

ISSN: 2005-4297 IJCA
Copyright ⓒ 2020 SERSC

will tend to perform better due to the smaller number of object removed after thresholding. Single Object Tracking in indoor environment is shown in 6(a), where original image captured from camera after background learning, with three frames numbers 160, 172 and 185 is shown. Fig 6(b) shows the image received and normalized process performed on incoming frames. This normalization is necessary to check whether the current image is representing foreground or background pixels. This is achieved by XORing the R,G,B values of both images.

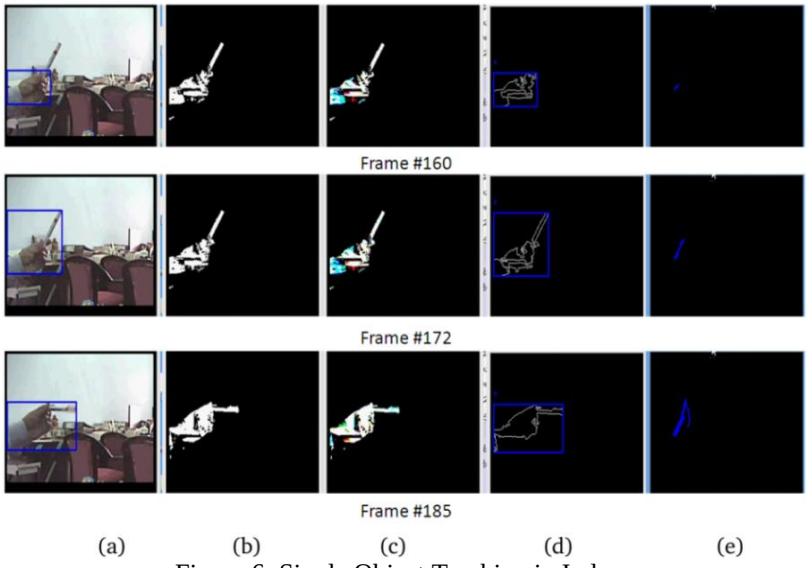

Figure 6: Single Object Tracking in Indoor
(a) New frame (b) Normalized image (c) Difference image (d) Preserved biggest contour (d) Trace of the object

Fig 6(d) shows in object being tracked in three frames and represented by rectangular bounded box in blue color surrounding the object of interest. It is also numbered with the same color of the bounding box depicting that; 1st object detected and tracked successfully. Finally, the trace or trajectory of the object movement is shown in Fig 6(d). Single object is successfully tracked in indoor environment with varying illumination, provided the camera is stationary.

4.2 Multiple Objects - Indoor Environment

Multiple objects are shown with different bounding boxes with different color while tracking as they appear in the frame. Fig 7 shows an example of tracking two objects in an indoor environment. The blue and green bounding

ISSN: 2005-4297 IJCA
Copyright ⓒ 2020 SERSC

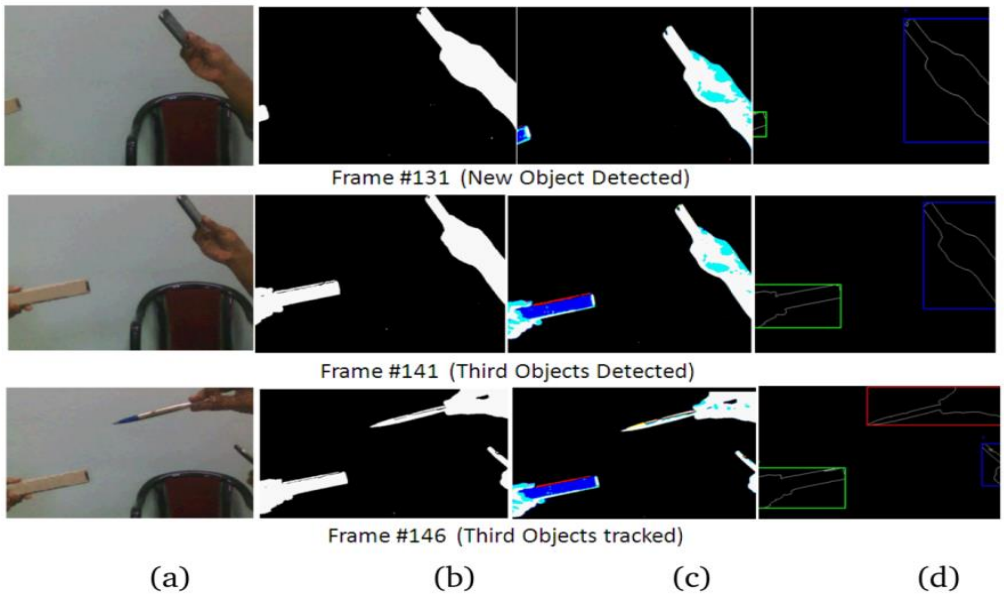

Figure 7: Multiple Objects Tracking in Indoor Environment
(a) original image, with four frames no 131, 141 and 146, (b) normalized image (c) temporal difference result of original image (d) object tracked box shows the position of the both the objects without occlusion.

In Fig. 7, three objects are detected and tracked and they are numbered as 1 and 2 as shown in all three frames number 131, 141 and 146. Successful moving object detection in outdoor environment is difficult task, since there are many kinds of problems such as illumination change, fake motion, and noise in background. Here the principles of proximity are used to check whether a new frame arriving from camera has same number of objects in the previous frame or not. We can see in the Fig. 7; as second object makes entry in field of view, it is appear as new object and new colorIndex is allocated to it. As object moves, it is covered by bounding box. If objects get closer to each other they get occluded and single color is shown for both the objects. Whereas, if objects separates out after a brief time, color and numbers are retained to show that they are older objects in the video sequence.

4.3 Single Object - Outdoor Environment

Following Fig 8 shows single object tracking in outdoor environment with human as an object 8(a) original image, with three different frames no 76, 111 and 118, 8(b) normalized image 8(c) temporal difference result.

We can see the first object is labeled with green color in frame 76. Whereas second person arrives in 111th frame is shown with red bounding box. It can be noticed in frame 111 of Figure 8(d), a human object has its own body parts as individuals objects and sudden movements of such object is detected as separate entity. Blue rectangle depicts the issue of such deformable object tracking.

Copyright ⓒ 2020 SERSC

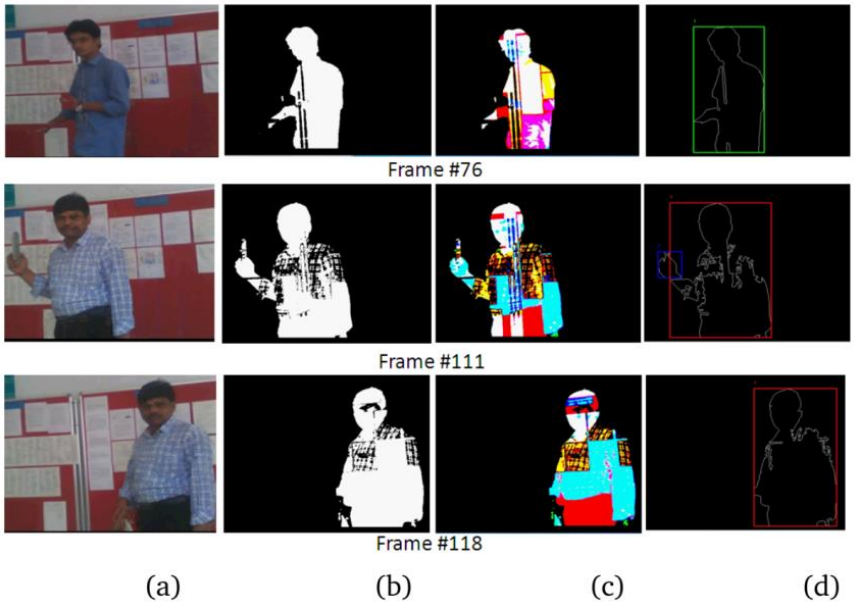

Figure 8: Single Objects tracking in Outdoor environment
(a) Original image no 76, 111, 118 (b) Normalized image (c) Object Detection (d) Object tracked

4.4 Multiple Objects - Outdoor Environment

Task of multiple object tracking is a challenging when there are complex interactions between target objects. It is important to be able to track multiple objects simultaneously to obtain good results. Many tracking algorithms have better performance under static background but get worse results under background with fake motions. Tracking multiple objects (human) in outdoor environment is also similarly tested. Tracking multiple objects in outdoor scenes (some time indoor environment) necessarily leads to the problem of occlusion; which needs to be handelned separately.

The proposed system is able to distinguish transitory and stopped foreground objects from static background objects in dynamic scenes; detect and distinguish left and removed objects; and can be feed to classifier to detect objects into different groups such as human, human group and vehicle. Therefore, this method is highly memory and time efficient. Moreover, our method can effectively deal with various scenes such as the indoor scene, the outdoor scene, and the cluttered scene.

5.0 Conclusion

We proposed a real-time object tracking system using stationary camera in different environmental conditions. Moving objects were detected and tracked against an environment consisting of objects of varying sizes, shapes and colors. At the beginning we modeled the background by using accumulated images of background frames from mean and standard deviation of first N frames. Each significant change in the object appearance thereafter, due to new object, old object disappearance was tracked based on the proximity of the target object. The visual resemblance was determined with respect to the detected object in the previous video frame and the last frame captures. Experimental result showed the effectiveness of the proposed method in object tracking under indoor and outdoor environment and partial occlusion. The results demonstrate a high performance of the proposed algorithm in the cluttered outdoor scene. Method does not handle full occlusion and consider it as single object.

References

[1] N. J. Uke and R. C. Thool, “Moving Vehicle Detection for Measuring Traffic Count Using OpenCV,” *J. Autom.*

Copyright ⓒ 2020 SERSC

Control Eng., vol. 1, no. 4, pp. 349–352, 2015.
[2] N. J. Uke and R. C. Thool, "Motion tracking system in video based on extensive feature set," *Imaging Sci. J.*, vol. 62, no. 2, 2014.
[3] A. Barth, "Vehicle Tracking and Motion Estimation Based on Stereo Vision Sequences," 2010.
[4] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems Perspectives and Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, 2012.
[5] C.-H. Lai and C.-C. Yu, "An Efficient Real-Time Traffic Sign Recognition System for Intelligent Vehicles with Smart Phones," *2010 Int. Conf. Technol. Appl. Artif. Intell.*, pp. 195–202, Nov. 2010.
[6] C. Fang, A. Member, S. Chen, and S. Member, "Road-Sign Detection and Tracking," *IEEE Trans. Veh. Technol.*, vol. 52, no. 5, pp. 1329–1341, 2003.
[7] Y. Xu, X. Cao, and H. Qiao, "An efficient tree classifier ensemble-based approach for pedestrian detection," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 41, no. 1, pp. 107–117, 2011.
[8] M. Wang and X. Wang, "Transferring a Generic Pedestrian Detector Towards Specific Scenes," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 12, p. 3274,3281, 2012.
[9] N. Mohan and H. Sharma, "An Improved Passive Tracking System for Automated Person of Interest (POI) Localization with SVM based Face detection," *Int. J. Control Autom.*, vol. 12, no. 6, pp. 190–199, 2019.
[10] A. K. Bhardwaj and A. K. Pandit, "Landmark Facial Detection by using Gaussian Regression Guided Network," *Int. J. Control Autom.*, vol. 12, no. 5, pp. 431–436, 2019.
[11] J. P. Kiran Kumar and M. C. Supriya, "Towards real time logo detection and classification using deep learning," *Int. J. Control Autom.*, vol. 13, no. 2, pp. 63–73, 2020.
[12] N. J. Uke, R. C. Thool, and P. S. Dhotre, "Drowsiness Detection – A Visual System for Driver Support," *Int. J. Electron. Commun. Comput. Eng.*, vol. 3, no. 2, pp. 29–33, 2012.
[13] N. M. and F. P. J. Fernández, R. Guerrero, "Multi-Level Paralelism in Image Identification," *Mec. Comput.*, vol. 28, no. 3, pp. 227–240, 2009.
[14] L. Cai, L. He, Y. Xu, Y. Zhao, and X. Yang, "Multi-object detection and tracking by stereo vision," *Pattern Recognit.*, vol. 43, no. 12, pp. 4028–4041, 2010.
[15] R. Tapu, B. Mocanu, and T. Zaharia, "DEEP-SEE: Joint object detection, tracking and recognition with application to visually impaired navigational assistance," *Sensors (Switzerland)*, vol. 17, no. 11, 2017.
[16] N. J. Uke and P. R. Futane, "Efficient method for detecting and tracking moving objects in video," in *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology, ICAECCT 2016*, 2017.
[17] K. Koffka, *Principles of Gestalt psychology*. Routledge, 1999.
[18] B. J. Scholl, "Object Persistence in Philosophy and Psychology," *Mind Lang.*, vol. 22, no. 5, pp. 563–591, Nov. 2007.
[19] A. Kundu, C. V. Jawahar, and K. M. Krishna, "Realtime moving object detection from a freely moving monocular camera," *2010 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2010*, pp. 1635–1640, 2010.
[20] N. Uke, "Real-Time Tracking of Multiple Moving Objects in Video Using Proximity - IJETT," *Int. J. Emerg. Trends Technol.*, vol. 4, no. 1, 2017.
[21] F. Porikli, "Achieving real-time object detection and tracking under extreme conditions," *J. Real-Time Image Process.*, vol. 1, no. 1, pp. 33–40, 2006.

Authors

Dr. Nilesh J. Uke, He received the B.E. degree in Computer Science and Engineering from Amaravati University, India, in 1995, and the M.E. from Bharathi Vidhyapeeth in 2005 and Ph.D. degrees in Computer Science, from SRTM University India, in 2014. He is currently a Principal and Professor at Trinity Academy of Engineering, Pune, Maharashtra, Pune. His current research interest includes Visual Computing, Artificial Intelligence, Human Computer Interface and Multimedia. He is a member of IEEE, ACM and Life Member of the Indian Society for Technical Education (ISTE) and Computer Society of India (CSI). nilesh.uke@gmail.com

Mrs. Shailaja Uke, She received the B.E. degree in Information Technology from Savitribai Phule University, India, in 2003, and the M.Tech (Information Technology) from Bharathi Vidhyapeeth in 2008 and pursuing her Ph.D. degrees in Computer Science from SRTM University. She is currently working as assistant professor at SKN SITS, Lonavala, Maharashtra. Her current research interest includes WSN, Object Oriented Modeling, Human Computer Interface. She is a member of Indian Society for Technical Education (ISTE). snuke@sinhgad.edu

Copyright ⓒ 2020 SERSC