IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. V (Sep – Oct. 2014), PP 01-08
www.iosrjournals.org

Occlusion detection in video sequences

Kapil A. Chavan[1], P. P. Halkarnikar[2]
[1]*(Department of Technology, Shivaji University,Kolhapur,India)*
[2]*(Dr.D.Y.Patil College of Engg. & Technology, Pune, India)*

Abstract: *An occlusion is the region between two overlapping objects with disparate motion. Detecting these occluded objects is crucial for many of the video processing. The Occlusion detection is decomposed into two independent sub problems. The First is to detect foreground objects on a frame-wise basis, by labeling each pixel in an image frame as either foreground or background. The second is to couple object observations at different points in a sequence to yield the object's motion trajectory and Occlusion. The motion segmentation is based on an adaptive background subtraction method that models each pixel as mixture of Gaussians. The Gaussian distributions are then evaluated to determine which are most likely to result from a background process. This is useful to track moving objects and detect occlusion in lighting changes, repetitive motions from cluster, and long term scene changes.*

Keywords: *Occlusion detection, Adaptive background estimation, Gaussians model*

I. Introduction

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images. Applications of computer vision include systems for:

- Controlling processes, e.g., an industrial robot
- Navigation, e.g., an autonomous vehicle or mobile robot
- Detecting events, e.g., visual surveillance or people counting
- Organizing information, e.g., indexing databases of images and image sequences
- Modeling objects or environments, e.g., medical image analysis or topographical modeling
- Interaction, e.g., the input to a device for computer-human interaction
- Automatic inspection, e.g., manufacturing applications

Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, learning, indexing, motion estimation, and image restoration.

- **Object recognition** –

This is the task of finding a given object in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary in different view. Object detection has applications in areas of computer vision, including image retrieval, video surveillance, and face detection. An image retrieval system is a computer system for browsing, searching and retrieving images from a large database of digital images. Video surveillance system can be used in home, shops for security. Face detection includes detecting and recognizing human faces via image processing algorithms.

Occlusion:-

Occlusions are omnipresent and are crucial for visual understanding in a 3D world. An occlusion event occurs whenever one object is covered or uncovered by another object that is spatially closer to the observer. The observer will consider here as video camera. The figures of occluded objects shown below:-

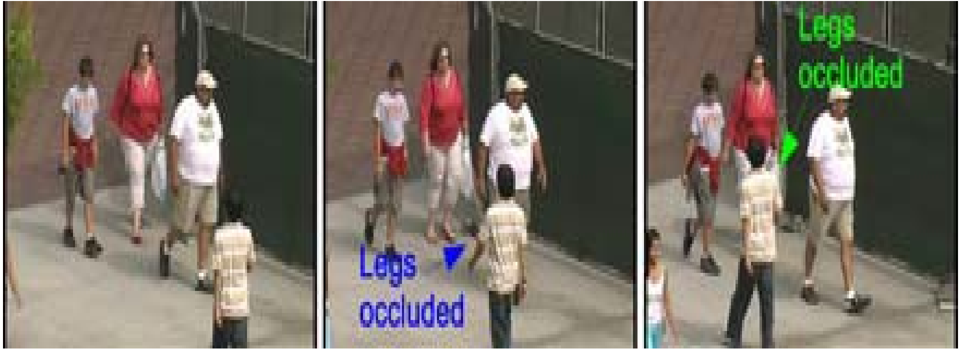


Fig 1. Example of changing occlusion relations.

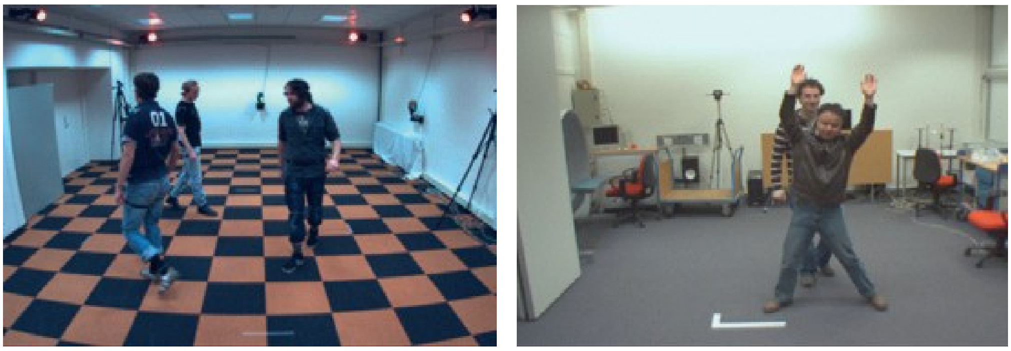


Fig 2. Example of self occlusion & inter person occlusion.

There are two type of occlusion can be present as self occlusion & inter person occlusion.

Self occlusion can be handled within each individual by using edge and color features. Inter-person occlusion handling might needs to take into account physical relationships between persons, such as distance and depth information [1].

II. Literature Survey

A number of methods have been applied to the problem of explicitly determining occlusion boundaries.

1) One early approach of J. Ullmann [2] determines occlusion boundaries for a set of known object types on a small pixel grid. This method showed promising performance assuming no noise and a set of a priori known objects. Later, occlusion detection was combined with motion estimation to classify occluded areas based on a photometric mismatch between frames. The drawback of this method is that any errors in the motion vector field (MVF) are likely to cause false detection of occlusion boundaries.
2) Learning-based research was conducted by R. Depommier and E. Dubois [3] , which used two separate models to describe scene motion, i.e., a two-parameter translational model for regular motion and a six-parameter generative model for occlusion boundaries.
3) V. Kolmogorov and R. Zabih [4] given, a graph cuts approach has been also considered in which the uniqueness constraint is utilized to guarantee proper occlusion handling.
4) N. Apostoloff and A. Fitzgibbon [5] present separate approach, occlusion events are determined by the presence of T-junctions in a spatiotemporal volume created from a video sequence.
5) S. Ince and J. Konrad [6] given, Geometric approaches have been also considered, analyzing the motion field alone to determine the presence of occlusions.
6) A more recent direction of A. Stein and M. Hebert [7], [8] makes use of local appearance cues as well as motion information to detect occlusion boundaries. The drawback of these learning-based methods is that, in order to obtain good detection performance, significant training data must be available.
7) Andrew stain & Derek hoiem provide strong cue for object detection. The appearance & motion cue are use together for detection of occluded object in moving sequences [1].
8) Feifei Huo & Emile A. Hendriks presents simultaneously tracking poses of multiple people is a difficult problem because of inter-person occlusions and self occlusions [9].
9) HeungKyu Lee presents the multiple image objects detection, tracking, and classification method using human articulated visual perception capability in consecutive image sequences [10].
10) Jiyan Pan , Bo Hu & Jian Qui Zhang propose a complete solution to robust and accurate object tracking in face of various types of occlusions which pose many challenges to correct judgment of occlusion situation and proper update of target template. In order to tackle those challenges, they first propose a content-adaptive progressive occlusion analysis (CAPOA) algorithm. Accurate tracking of an occluded target is achieved by rectifying the target location using the variant-mask template matching (VMTM) [11].
11) Hieu T. Nguyen and Arnold W.M. Smeulders propose a new method for object tracking in image sequences using template matching. To update the template, appearance features are smoothed temporally by robust Kalman filters, one to each pixel. The resistance of the resulting template to partial occlusions enables the accurate detection and handling of more severe occlusions. Abrupt changes of lighting conditions can also be handled, especially when photometric invariant color features are used. The method has only a few parameters and is computationally fast enough to track objects in real time [12].
12) Loris Bazzani, Domenico Bloisi, Vittorio Murino gives Visual tracking of multiple targets is a key step in surveillance scenarios, far from being solved due to its intrinsic ill-posed nature. A comparison of Multi-Hypothesis Kalman Filter and Particle Filter-based tracking is presented. Both methods receive input from a novel online background subtraction algorithm. The aim of this work is to highlight advantages and

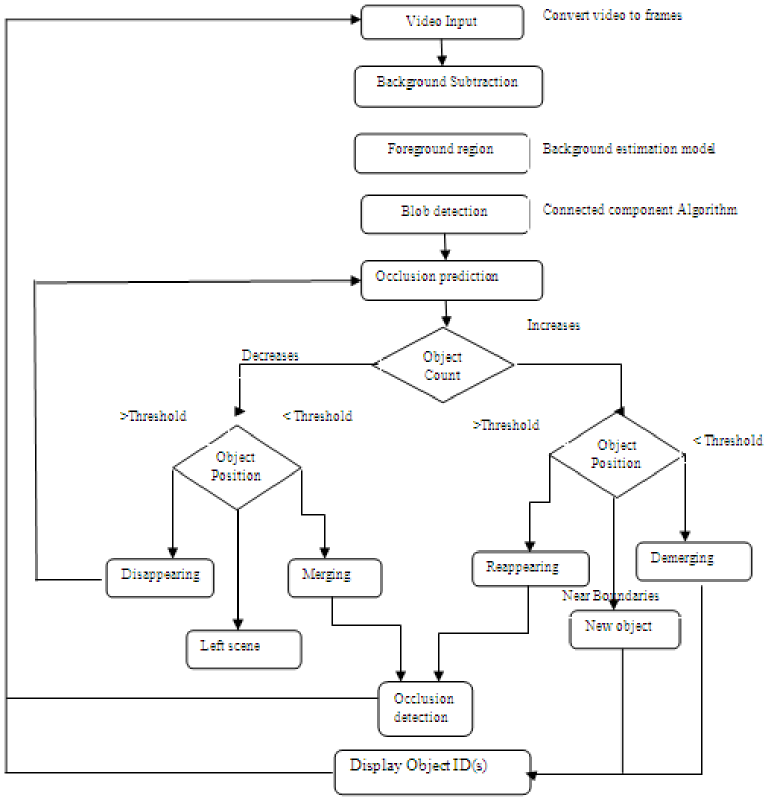
disadvantages of such tracking techniques. Results are performed using public challenging data set (PETS 2009), in order to evaluate the approaches on significant benchmark data [13].

13) Anil M. Cheriyadat, Budhendra L. Bhaduri, and Richard J. Radke propose an object detection system that uses the locations of tracked low-level feature points as input, and produces a set of independent coherent motion regions as output. As an object moves, tracked feature points on it span a coherent 3D region in the space-time volume defined by the video. In the case of multi-object motion, many possible coherent motion regions can be constructed around the set of all feature point tracks [14].

14) Afef Salhi and Ameni Yengui Jammoussi presents a implementation of an object tracking system in a video sequence. This object tracking is an important task in many vision applications. The main steps in video analysis are two: detection of interesting moving objects and tracking of such objects from frame to frame. In a similar vein, most tracking algorithms use pre-specified methods for preprocessing. In this work, several object tracking algorithms (Meanshift, Camshift, Kalman filter) with different preprocessing methods. Then evaluated the performance of these algorithms for different video sequences. The obtained results have shown good performances according to the degree of applicability and evaluation criteria [15].

III. Proposed Method

The proposed method specified below which consist of steps as Convert video to frames, Background estimation model, Blob detection, Occlusion prediction & detection.



IV. Adaptive Background for Detection

If each pixel resulted from a single surface under fixed lighting, a single Gaussian would be sufficient to model the pixel value while accounting for acquisition noise. If only lighting changed over time, a single, adaptive Gaussian per pixel would be sufficient. In practice, multiple surfaces often appear in the view frustum of a particular pixel and the lighting conditions change. Thus, multiple, adaptive Gaussians are required. We use an adaptive mixture of Gaussians to approximate this process. Each time their parameters are updated, the Gaussians are evaluated using a simple heuristic to hypothesize which are most likely to be part of the “background process". Pixel values that do not match one of the pixel's “background" Gaussians are grouped using connected components. Finally, the connected components are tracked across frames using a multiple hypothesis tracker.

V. Mixture Model

We consider the values of a particular pixel over time as a “pixel process", i.e. a time series of scalars for gray values or vectors for color pixel values. At any time, t, what is known about a particular pixel, $\{x_0; y_0\}$, is its history

$$\{X_1, \ldots., X_t\} = \{I(X_0, Y_0, i): 1 \leq i \leq t\}$$

where I is the image sequence.

We chose to model the recent history of each pixel, $\{X_1, \ldots., X_t\}$, as a mixture of K Gaussian distributions. The probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} W_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where K is the number of distributions, $W_{i,t}$ is an estimate of the weight (the portion of the data accounted for by this Gaussian) of the ith Gaussian in the mixture at time t, $\mu_{i,t}$ and $\sum_{i,t}$are the mean value and covariance matrix of the ith Gaussian in the mixture at time t, and where η is a Gaussian probability density function

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)|\Sigma|} e^{-1/2(Xt - \mu t)\Sigma(Xt - \mu t)}$$

K is determined by the available memory and computational power. Currently, from 3 to 5 are used. Also, for computational reasons, the covariance matrix is assumed to be of the form:

$$\sum\nolimits_{k,t} = \sigma_k^2 I$$

This assumes that the red, green, and blue pixel values are independent and have the same variances. While this is certainly not the case, the assumption allows us to avoid a costly matrix inversion at the expense of some accuracy.

Thus, the distribution of recently observed values of each pixel in the scene is characterized by a mixture of Gaussians. A new pixel value will, in general, be represented by one of the major components of the mixture model and used to update the model.

If the pixel process could be considered a stationary process, a standard method for maximizing the likelihood of the observed data is expectation maximization [16]. Because there is a mixture model for every pixel in the image, implementing an exact EM algorithm on a window of recent data would be costly. Also, lighting changes and the introduction or removal of static objects suggest a decreased dependence on observations further in the past. These two factors led us to use the following on-line K-means approximation to update the mixture model.

Every new pixel value, Xt, is checked against the existing K Gaussian distributions, until a match is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution3. This threshold can be perturbed with little effect on performance. This is effectively a per pixel/per distribution threshold. This is extremely useful when different regions have different lighting, because objects which appear in shaded regions do not generally exhibit as much noise as objects in lighted regions. A uniform threshold often results in objects disappearing when they enter shaded regions.

If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t})$$

Where α is the learning rate and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized. 1/α defines the time constant which determines the speed at which the distribution's parameters change. $w_{k,t}$ is effectively a causal low-pass filtered average of the (threshold) posterior probability that pixel values have matched model k given observations from time 1 through t. This is equivalent to the expectation of this value with an exponential window on the past values.

The μ and σ parameters for unmatched distributions remain the same. The parameters of the distribution which matches the new observation are updated as follows

$$\mu_t = (1 - p)\mu_{t-1} + pX_t$$

$$\sigma\frac{2}{t}=(1-p)\sigma\frac{2}{t-1}+p\,(Xt-\mu t)T\,(Xt-\mu t)$$

Where

$$p = \alpha\eta(Xt|\square k, \sigma k)$$

is the learning factor for adapting current distributions5. This is effectively the same type of causal low-pass filter as mentioned above, except that only the data which matches the model is included in the estimation.

One of the significant advantages of this method is that when something is allowed to become part of the background, it doesn't destroy the existing model of the background. The original background color remains in the mixture until it becomes the Kth most probable and a new color is observed. Therefore, if an object is stationary just long enough to become part of the background and then it moves, the distribution describing the previous background still exists with the same μ and σ^2, but a lower w, and will be quickly reincorporated into the background.

VI. Background Model Estimation

As the parameters of the mixture model of each pixel change, we would like to determine which of the Gaussians of the mixture are most likely produced by background processes. Heuristically, we are interested in the Gaussian distributions which have the most supporting evidence and the least variance.

To understand this choice, consider the accumulation of supporting evidence and the relatively low variance for the "background" distributions when a static, persistent object is visible. In contrast, when a new object occludes the background object, it will not, in general, match one of the existing distributions which will result in either the creation of a new distribution or the increase in the variance of an existing distribution. Also, the variance of the moving object is expected to remain larger than a background pixel until the moving object stops. To model this, we need a method for deciding what portion of the mixture model best represents background processes.

First, the Gaussians are ordered by the value of w/σ. This value increases both as a distribution gains more evidence and as the variance decreases. After re-estimating the parameters of the mixture, it is sufficient to sort from the matched distribution towards the most probable background distribution, because only the matched models relative value will have changed. This ordering of the model is effectively an ordered, open ended list, where the most likely background distributions remain on top and the less probable transient background distributions gravitate towards the bottom and are eventually replaced by new distributions.

Then the first B distributions are chosen as the background model, where

$$B = \text{argmin}_b(\sum_{k=1}^{b} wk > T)$$

Where T is a measure of the minimum portion of the data that should be accounted for by the background. This takes the "best" distributions until a certain portion, T, of the recent data have been accounted for. If a small value for T is chosen, the background model is usually unimodal. If this is the case, using only the most probable distribution will save processing.

If T is higher, a multi-modal distribution caused by a repetitive background motion (e.g. leaves on a tree, a flag in the wind, a construction flasher, etc.) could result in more than one color being included in the background model. This results in a transparency effect which allows the background to accept two or more separate colors.

VII. Connected components

The method described above allows us to identify foreground pixels in each new frame while updating the description of each pixel's process. These labeled foreground pixels can then be segmented into regions by a two-pass, connected components algorithm [17].Because this procedure is effective in determining the whole moving object, moving regions can be characterized not only by their position, but size, moments, and other shape information. Not only can these characteristics be useful for later processing and classification, but they can aid in the tracking process.

VIII. Occlusion detection

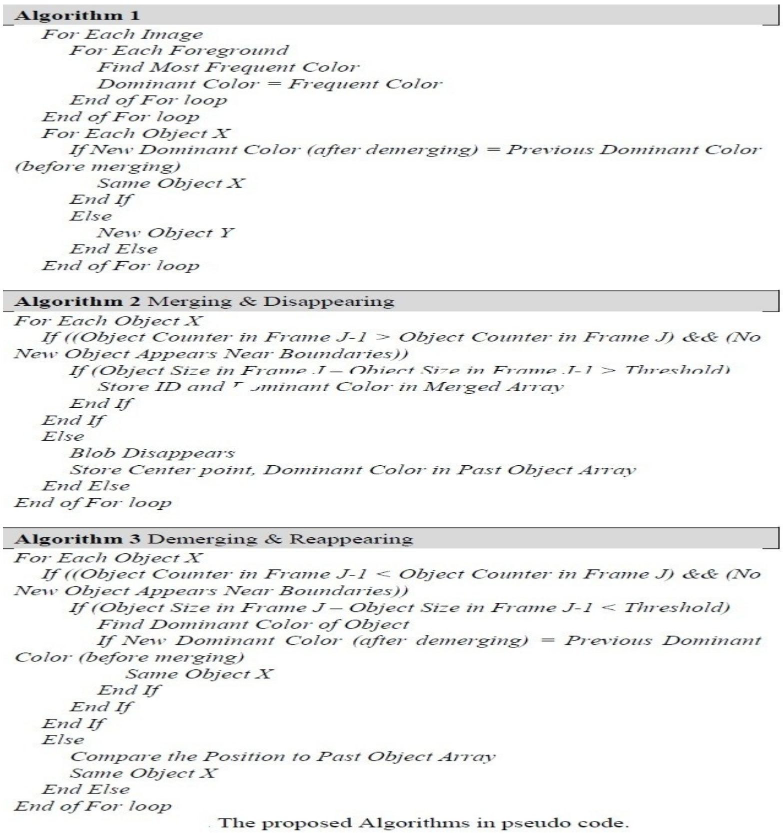
Establishing correspondence of connected components between frames is accomplished using a linearly predictive multiple hypotheses tracking algorithm which incorporates both position and size. We have implemented method for seeding and maintaining sets of Kalman filters.

At each frame, we have an available pool of Kalman models and a new available pool of connected components that they could explain. First, the models are probabilistically matched to the connected regions that they could explain. Second, the connected regions which could not be sufficiently explained are checked to and new Kalman models. Finally, models whose fitness (as determined by the inverse of the variance of its prediction error) falls below a threshold are removed.

Matching the models to the connected components involves checking each existing model against the available pool of connected components which are larger than a pixel or two. All matches with relatively small error are used to update the corresponding model. If the updated models have sufficient fitness, they will be used in the following frame. If no match is found a “null" match can be hypothesized which propagates the model as expected and decreases its fitness by a constant factor. If the object reappears in a predictable region of uncertainty shortly after being lost, the model will regain the object. Because our classification system requires tracking sequences which consist of representations of a single object, our system generally breaks tracks when objects interact rather than guessing at the true correspondence.

The unmatched models from the current frame and the previous two frames are then used to hypothesize new models. Using pairs of unmatched connected components from the previous two frames, a model is hypothesized. If the current frame contains a match with sufficient fitness, the updated model is added to the existing models. To avoid possible combinatorial explosions in noisy situations, it may be desirable to limit the maximum number of existing models by removing the least probable models when excessive models exist. In noisy situations (e.g. ccd cameras in low-light conditions), it is often useful to remove the short tracks that may result from random correspondences.

The algorithm for detect occlusion counters in frames for Merging, Demerging, Disappearing & Reappearing as follows:

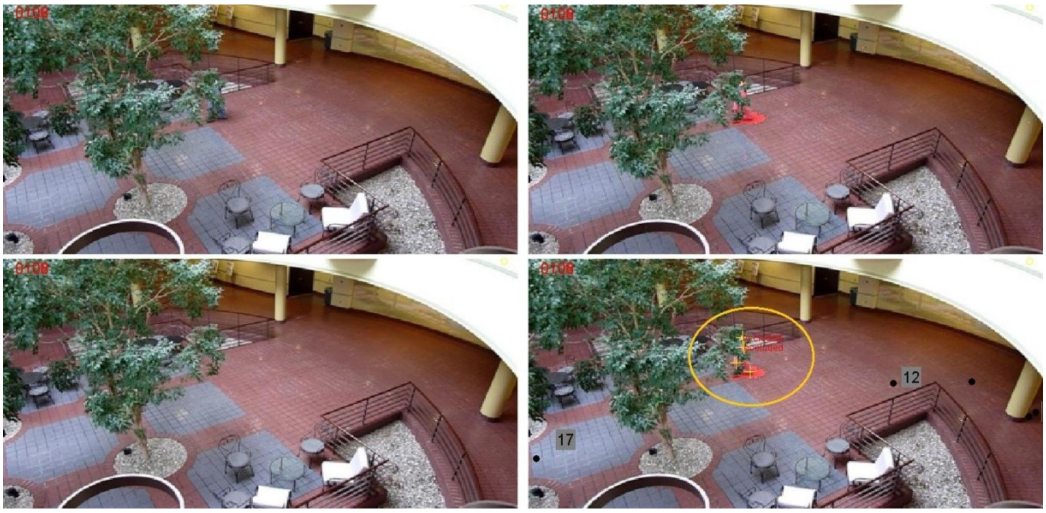


```
Algorithm 1
    For Each Image
        For Each Foreground
            Find Most Frequent Color
            Dominant Color = Frequent Color
        End of For loop
    End of For loop
    For Each Object X
        If New Dominant Color (after demerging) = Previous Dominant Color
(before merging)
            Same Object X
        End If
        Else
            New Object Y
        End Else
    End of For loop

Algorithm 2 Merging & Disappearing
For Each Object X
    If ((Object Counter in Frame J-1 > Object Counter in Frame J) && (No
New Object Appears Near Boundaries))
        If (Object Size in Frame J – Object Size in Frame J-1 > Threshold)
            Store ID and Dominant Color in Merged Array
        End If
    End If
    Else
        Blob Disappears
        Store Center point, Dominant Color in Past Object Array
    End Else
End of For loop

Algorithm 3 Demerging & Reappearing
For Each Object X
    If ((Object Counter in Frame J-1 < Object Counter in Frame J) && (No
New Object Appears Near Boundaries))
        If (Object Size in Frame J – Object Size in Frame J-1 < Threshold)
            Find Dominant Color of Object
            If New Dominant Color (after demerging) = Previous Dominant
Color (before merging)
                Same Object X
            End If
        End If
    End If
    Else
        Compare the Position to Past Object Array
        Same Object X
    End Else
End of For loop
```

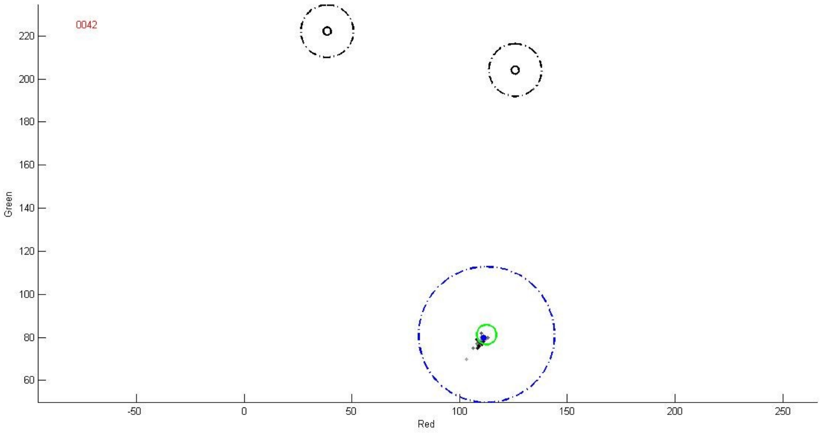
. The proposed Algorithms in pseudo code.

IX. Experiments & Results

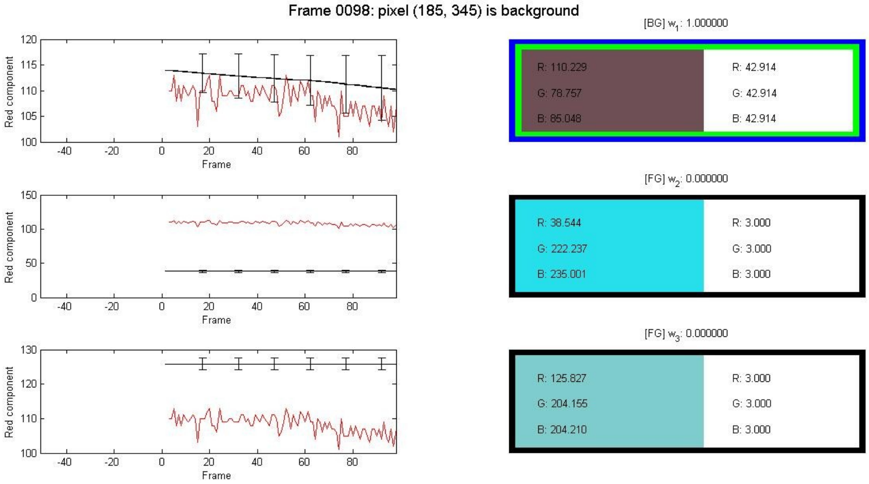
[1] The result shows person is occluded under tree. Sequentially Background model, Object in the frame, marked object & object occluded under is shown & marked.



[2] Scatter plot for debugging pixel shown out in the result.



[3] The pixel statistics for debugging pixel is shown in the results.



X. Conclusion

This paper has shown a novel, probabilistic method for background subtraction. It involves modeling each pixel as a separate mixture model. We implemented approximate method which is stable and robust. This method deals with slow lighting changes by slowly adapting the values of the Gaussians. It also deals with multi-modal distributions caused by shadows, specularities, swaying branches, computer monitors, and other troublesome features of the real world which are not often mentioned in computer vision. It recovers quickly when background reappears and has a automatic pixel-wise threshold. All these factors have made this tracker an essential part of our activity and object classification research.

This system has been successfully used to detect occlusion in indoor environments, people and cars in outdoor environments. All these situations involved different cameras, different lighting, and different objects being tracked. This system achieves our goals of performance over extended periods of time without human intervention.

References

Journal Papers:

[1]. Andrew Stein , Derek Hoiem ," Learning to Find Object Boundaries Using Motion Cues", 2005.

[2]. J. Ullmann, "Analysis of 2-D occlusion by subtracting out," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 4, pp. 485–489, Apr.1992.

[3]. R. Depommier and E. Dubois, "Motion estimation with detection of occlusion areas," in Proc. IEEE Conf. Acoust., Speech, Signal Process., vol. 3, pp. 269–272, Mar. 1992.

[4]. M. J. Black and D. J. Fleet, "Probabilistic detection and tracking of motion boundaries," Int. J. Comput. Vis., vol. 38, no. 3, pp. 231–245, Jul./Aug. 2000.

[5]. V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in Proc. Int. Conf. Comput. Vis., pp. 508–515, Sep. 2007.

[6]. N. Apostoloff and A. Fitzgibbon, "Learning spatiotemporal T-junctions for occlusion detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, pp. 553–559, 2005.

[7]. S. Ince and J. Konrad, "Geometry-based estimation of occlusions from video frame pairs," in Proc. IEEE Int. Conf. Accoust. Speech Signal Process., vol. 2, pp. ii/933–ii/936.

[8]. A. Stein and M. Hebert, "Combining local appearance and motion cues for occlusion boundary detection," in Proc. Brit. Mach. Vision Conf., pp. 1–10, Sep. 2007.

[9]. Feifei Huo , Emile A. Hendriks ," Multiple people tracking and pose estimation with occlusion estimation", 2012.

[10]. HeungKyu Lee,"Multiple Image Objects Detection, Tracking and Classification Using Human Articulated Visual Perception Capability".

[11]. Jiyan Pan, BoHu and Jian Qiu Zhang.," Robust and Accurate Object Tracking under Various Types of Occlusions", IEEE Transactions on circuits and systems for video technology, vol. 18, no. 2, February 2008.

[12]. Hieu T. Nguyen and Arnold W.M. Smeulders.," Fast Occluded Object Tracking by a Robust Appearance Filter", IEEE Transactions on pattern analysis & template matching, vol.28, no 8,Augest 2004.

[13]. Loris Bazzani, Domenico Bloisi, Vittorio Murino.," A Comparison of Multi Hypothesis Kalman Filter and Particle Filter for Multi-target Tracking".

[14]. Anil M. Cheriyadat, Budhendra L. Bhaduri, and Richard J. Radke.," Detecting Multiple Moving Objects in Crowded Environments with Coherent Motion Regions".

[15]. Afef Salhi and Ameni Yengui Jammoussi.," Object tracking system using Camshift, Meanshift and Kalman filter", World Academy of Science, Engineering and Technology 64 2012.

[16]. A Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society, 39 (Series B):1-38, 1977.

[17]. B. K. P. Horn. Robot Vision, pp. 66-69, 299-333. The MIT Press, 1986.