The 8th International Conference on
Computer Science & Education (ICCSE 2013)
April 26-28, 2013. Colombo, Sri Lanka

Appearance Based Tracking with Background Subtraction

Dileepa Joseph Jayamanne
Electronic and
Telecommunication Engineering
University of Moratuwa-Sri Lanka
Email:dileepajj@gmail.com

Jayathu Samarawickrama
Electronic and
Telecommunication Engineering
University of Moratuwa-Sri Lanka
Email:jayathu@ent.mrt.ac.lk

Ranga Rodrigo
Electronic and
Telecommunication Engineering
University of Moratuwa-Sri Lanka
Email:ranga@ent.mrt.ac.lk

***Abstract*—Grouping the detected feature points traditionally requires the storage of long corner tracks. The traditional method does not permit to arrive at a decision to cluster the feature points based on a frame by frame basis. This paper presents a method to group the feature points directly into objects using the most recent 20 frames. The detected corner features are validated and clustered based on two approaches. When objects move in isolation, an EM algorithm is used to cluster and every object is detected and tracked. When objects move under partial occlusion, the corner features are clustered based on an agglomerative hierarchical clustering approach. A probabilistic framework has also been applied to determine the object level membership of the candidate corner features. A novel foreground estimation algorithm with an accuracy of 98% based on color information, background subtraction result and detected corner features is also presented.**

I. Introduction

Monitoring vehicular traffic using surveillance cameras still need manual intervention. There have been work undertaken to detect, localize, and classify vehicles and to analyze vehicle behaviour like estimation of average speed, trajectory, flow rate and density using the video footage obtained through cameras [1]–[4]. The purpose of developing such automatic traffic surveillance systems is to track vehicles, monitor the vehicle traffic and extract traffic parameters. This system will enable to reduce manual intervention in monitoring traffic and identify regular road users and traffic violators. Traffic surveillance systems rely on accurate object detection and tracking.

A. *Related Work*

Extracting and tracking individual corner features or interest points and grouping them based on their trajectories is a method used in object tracking [1]. Although the segmentation of occluded objects is easy to perform, tracking the same corner feature for a long period of time is challenging. Furthermore a set of long corner trajectories will have to be processed and kept, resulting in heavy use of memory. The corner features are tracked from point of entry to exit and grouped directly into objects after obtaining long trajectories of the corner tracks using the proximity and motion history [1]. Points that rigidly move forward are grouped allowing occlusion to be handled at the cost of computational memory. This is a typical example of a single level clustering and feature grouping approach.

Kim [2] considers a dynamic multi-level feature grouping approach to obtain refined trajectories in real time in contrast to Coifman *et al.* [1]. Emerging feature points are initially grouped into small clusters using a Normalized-cut algorithm [5] and further a variation of Expectation Maximization algorithm is applied to serve two purposes namely; i) continue clustering the same clusters previously detected by N-cut in the next set of frames and ii) group the clusters that are detected to achieve object level grouping.

Background subtraction provides the base for most of the object tracking algorithms. Initially, it extracts a background hypothesis from a sequence of frames. The difference of the background hypothesis and the current frame separates the foreground. Although the computational time it requires is relatively small, it is unable to deal with occlusions, shadows, and sudden illumination changes. Kim [2] has combined the background subtraction and the feature tracking and grouping algorithms to produce high quality object trajectories from fragmented feature tracks. Kim's augmentation to the background subtraction algorithm uses a low-level feature tracking as a cue to validate the estimated foreground region; however, Kim has estimated the silhouettes based on conventional morphological operations. The main drawback of this estimation is considering the excess regions outside the boundaries of the objects as the foreground. This results in classifying such actual background pixels as a part of the foreground.

Tracking applications can be conducted using a fixed feature space. However, Collins *et al.* [3] have proposed a method for evaluating on-line, adaptive selection of appropriate feature spaces for tracking and for adjusting the set of features used to improve tracking performance. They have claimed that the features that best discriminate the object and the background are best for tracking the object. Although a wide range of features can be used for tracking like color, texture, shape, and motion, they have only considered a linear combinations of camera R, G, B pixel values to compose the set of seed candidate features.

Selecting the right features plays an important role not only in tracking but also in detection. In general, the most desirable property of a visual feature is its uniqueness in order for the

978-1-4673-4463-0/13/$31.00 ©2013 IEEE

objects to be easily distinguished from feature space [6]. Buch *et al.* [4] present another method, 3DHOG, for detection and classification of road users in urban scenes. This system works even if the appearance of vehicles varies substantially with the viewing angle. This is an extension to HOG feature extraction [7] by applying 3D spatial modeling to operate on still images. This overcomes the reliability limitations of motion silhouettes. This is an example of using a complete different feature space to detect the objects.

In this paper we directly group detected corner features into objects without waiting till long corner trajectories are available to cluster. Our method takes into consideration both ideas of Kim [2] and Malik *et al.* [1]. Our focus is to use a single clustering algorithm to achieve object level clustering that could be applied on a frame by frame basis using Kim's work as a base. We also propose a method to incorporate color information to threshold the background subtraction result to accurately estimate the foreground pixels. Although we use conventional morphological operations to preserve the shape of the objects, we apply dilation with a small structuring element on our background subtraction result. This reduces estimating a larger excess region around the boundary of the actual object as a part of the foreground. In our work, we propose methods

- to incorporate color information together with conventional morphological operations on background subtraction result, to preserve the shapes of silhouettes that correspond to different sized objects and arrive at a better foreground estimate,
- to achieve single-level clustering that directly correspond to objects without using long corner tracks,
- to validate cluster membership of corner features based on a probabilistic framework using Bayesian reasoning.

II. Methodology

The blobs that are detected using background subtraction are validated through the KLT point tracks. These tracks are intern assigned to clusters using several mechanisms. We have applied an EM algorithm to group the detected corner tracks for instances where vehicles move in isolation and an agglomerative hierarchical clustering algorithm for instances where vehicles move under partial occlusion. The outline of the proposed methodology is as follows:

1) Modeling the background and updating it every 15 frames.
2) Estimating the foreground region: Generating object blobs using color information and conventional morphological operations.
3) Detecting corner features using the KLT tracker [8].
4) Validating corner features and validating the estimated foreground region.
5) Applying a single level hierarchical clustering algorithm when vehicles move under partial occlusion.
6) Applying single level clustering based on an EM algorithm for vehicles moving in isolation.
7) Applying Bayesian reasoning to determine the membership of a corner feature for each cluster.
8) Validating the clustered feature points based on their membership.
9) Tracking clusters and obtaining the trajectories of the corresponding objects.

A. *The Background Model*

Our approach is mainly based on Kim's work [2]. We have implemented the suggested background model with a modification to the background subtraction algorithm when estimating the actual foreground region. We update the background every 15 frames and the frame rate of the videos considered is 30 fps. The background model used is as follows [2]:

$$B_{t+1} = I_c(B_t) \qquad \text{when } M_t = 1 \qquad (1)$$

$$B_{t+1} = I_c((1-\alpha)B_t + \alpha N_t) \qquad \text{when } M_t = 0 \qquad (2)$$

where B_t represents the background model at time t, B_{t+1} is the next background update, N_t is the temporal median of the recent 15 frames, M_t is the binary moving object hypothesis mask, α is a value in [0,1] and $I_c(.)$ is an illumination-correction function which is applied to each of the R, G, B values as follows:

$$I_c(R,G,B) = (k_R R, k_G G, k_B B) \qquad (3)$$

where k_R, k_G and k_B are determined by voting on R_C/R, G_C/G, and B_C/B over all the pixels in the images and (R_C, G_C, B_C) are the pixel values of the current frame.

M_t is a state that each pixel could occupy. $M_t = 1$ indicates that the considered pixel is estimated to be a foreground pixel and if $M_t = 0$ the considered pixel is estimated to be a non foreground pixel. M_t is generated from the resultant difference image obtained from subtracting the background from the current frame. How we compute M_t based on color information and morphological operations will be introduced in the next subsection. The computation of k_R, k_G and k_B for each R, G, B values of a frame used to update the background in our implementation is as follows:

$$\frac{\text{Average of the mid 50\% of the current frame}}{\text{Average of the mid 50\% of the previous background}}$$

*1) **Silhouette generation based on color to preserve shape of the foreground:*** Converting the conventional background subtraction result to its binary format by using a threshold would result in a loss of regions that correspond to the actual foreground. Even if the fragmented pieces that appear could be dilated to determine the boundary and be filled in to obtain the object blobs/silhouettes, it is difficult to define a single structuring element that is capable of dilating and preserving the shapes of different sizes of objects. Therefore we use color together with the conventional morphological operations to preserve the shapes of silhouettes that correspond to different sizes of objects. The proposed mechanism used to determine the value of M_t and to improve generating silhouettes is

$$PC_x = \frac{(I_x - B_t)}{B_t} \times 100\% \qquad (4)$$

where x represents R, G, B colour values of each pixel, I_x is the current frame, B_t is the recent background update and PC_x is the percentage colour change compared to the recent background update.

For each pixel when we obtain the percentage color change compared to the recent background update, we select the maximum of the three percentages and apply a threshold to determine whether the pixel belongs to the estimated foreground or non foreground region. Then we dilate the resultant binary image using small structuring elements. Next we fill holes, remove small regions, and apply connected component analysis to label the regions and obtain the blobs. In this manner we validate each blob by detecting corner features [8], [9]. If we do not detect corner features in a blob, it is considered as a false foreground region and M_t is set to 0 for all the pixels within the region otherwise M_t is set to 1 for all the pixels within the validated blobs considered as the final estimated foreground region.

B. Single Level Clustering

Grouping feature points into objects either could be direct [1] or it could be arriving at an intermediate stage of grouping before clustering into objects [2]. Our approach attempts to group the feature points directly into object level clusters immediately after being detected without monitoring how closely a group of points appear and rigidly move forward.

Our method mainly consists of two algorithms that could be applied based on the scenario. If all the objects move in isolation, object level clustering could be directly achieved by applying a two dimensional EM Algorithm [10] on the (X, Y) position coordinates of the feature points. EM requires not only the number of initialization points, but also their corresponding values close to the expected positions where clusters need to appear so that the desired clustering could be achieved. Thus we have used blob centers as the initialization points since each object corresponds to a single blob when objects move in isolation. When vehicles are partially occluded, blobs corresponding to actual objects are merged and therefore this approach fails in this scenario particularly because of not having the correct number of initialization points and their initial values.

In order to achieve object level clustering under partial occlusion, the second algorithm we have used is an agglomerative hierarchical clustering approach. Each feature point is considered as a different cluster and pairs of clusters are merged when moving up the hierarchy [11]. We have considered the x and y position coordinates, speed and the trajectory of each of the feature points when generating the feature matrix. Every instance of the feature matrix has been used to generate the matrix that encodes a tree of hierarchical clusters.

In both clustering algorithms, for a given frame once the feature points are detected, the decision to cluster will be made after the feature points have been tracked for next 20 frames. For each frame, feature points will be re-detected. Within the next 20 frames the speed of a feature point is computed using each of the two recent consecutive frames. Thus the speed of a feature point at a given instance is considered as the median speed of these set of speeds. But each point's trajectory is obtained considering only its next appearance. In hierarchical clustering, the absolute gradient is used as the feature "trajectory" assuming each feature point's path follows a straight-line.

1) ***Corner Feature Detection and Validation****:* In our work, we have used corner features as our feature points. The corner features are detected and tracked using the KLT tracker [8], [9]. Our work is based on Birchfield's KLT implementation [8] and the detected corner features are validated using the criteria mentioned in [2]. If a detected corner feature could be tracked thrice and if it does not have a match in the corresponding background image, such a point is termed as a valid feature point. Such validated corner points are then used to validate object blobs that results in the removal of false foreground regions facilitating a more accurate estimation of M_t.

2) ***Membership of corner features****:* Once the detected corner features are clustered, the cluster membership of each feature point is determined based on a probabilistic framework. As in [2], for each clustered feature point, given the parameters, a Bayesian reasoning is applied to compute the posterior probability of the feature point to find out whether it belongs to the cluster or not. The extracted parameters are

- the ratio of being in the same blob: r
- the proximity: p
- the history of motion: m

as in Kim's work [2]. The computation of the posterior probability of the cluster membership of a feature point depends on the generated prior distributions. For each feature point, the prior probability of being a member or not is 0.5 and thus it indicates the two possible states. Therefore for each parameter, we estimate two individual probability distributions. One distribution to represent a feature point being a member and the other to represent a feature point not being a member.

We used a semi-supervised procedure to extract these parameters to generate the probability distributions. The parameter r—ratio of being in the same blob— indicates that within the next 20 frames and out of 20, the number of times a particular feature point appearing on the actual object. For each feature point, r is actually the ratio of being in the actual object. When generating r—member distribution— all such values of the feature points for a set of frames are considered. In order to obtain $\sim r$—not a member distribution— as in before, each feature point is tracked for 20 times and out of 20, the number of times a feature point appearing outside the object is considered. The Fig. 1 shows the histograms obtained after normalizing r to be in [0,1] with 20 bins.

The parameter p—the proximity— refers to the minimum distance from the ellipse boundary to each clustered feature point. In order to extract this parameter, ellipses are drawn around the candidate clusters and the minimum distance from

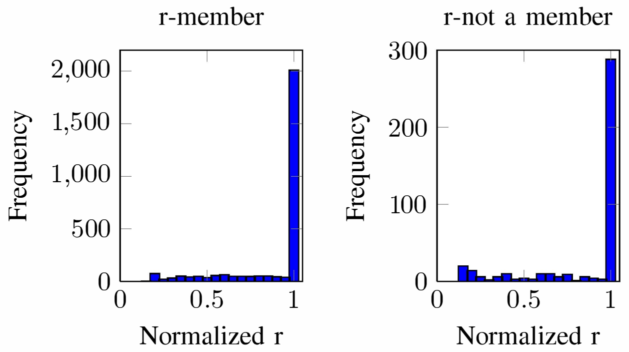

Fig. 1: left-probability distribution obtained for r—ratio of being in the same blob— for a feature point appearing on the actual object, right-probability distribution obtained for $\sim r$—ratio of not being in the same blob— for a feature point appearing outside the actual object.

the ellipse boundary for each of these points is obtained. The distance of a point that lies inside the ellipse is used to generate the p—member distribution— and the distance of a point that lies outside the ellipse is used to generate the $\sim p$—not a member distribution—. Fig. 2 shows the the histograms obtained after normalizing p to be in [0,1] with 20 bins.

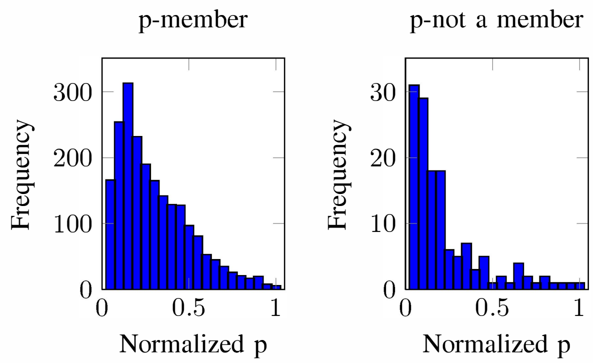

Fig. 2: left-probability distribution obtained for p—the proximity— for distance of a feature point that lies inside the ellipse boundary, right-probability distribution obtained for $\sim p$—non proximity— for distance of a feature point that lies outside the ellipse boundary.

The parameter m—the history of motion— refers to the speed of a feature point. We compute the m-member distribution by tracking each feature point within next 20 frames and obtaining the number of speeds of the feature point that lie close to the median value out of 20. In order to generate $\sim m$—not a member distribution—, number of speeds of each feature point that do not lie close to the median out of 20 have been used. Fig. 3 shows the the histograms obtained after normalizing m to be in [0,1] with 20 bins.

For each appearance of the feature point, the speed is computed using two recent consecutive frames. How similar a particular speed compared to its median is computed as follows:

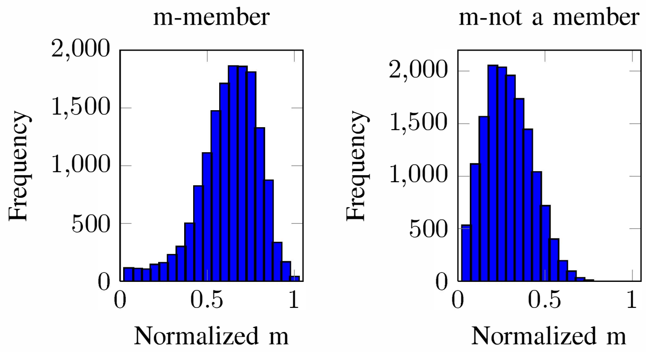

Fig. 3: left-probability distribution obtained for m—the history of motion— for speeds of the feature point that lie close to the median value, right-probability distribution for $\sim m$ for speeds of the feature point that lie away from the median value.

$$M = \frac{\text{abs}(S_j - \bar{S})}{S_{\max} - S_{\min}} \quad \text{where} \tag{5}$$

M is the measure that determines how similar the detected speed to the median speed, j is the number of times the speed could be computed within the next 20 frames, $\bar{S}$ is the median speed, $S_{\max}$ and $S_{\min}$ refer to the maximum and minimum speeds respectively.

Therefore the posterior probability of the cluster membership of a candidate feature point is computed separately considering the extracted parameters individually as follows:

$$P(\text{member}|x) = \frac{P(x|\text{member})}{P(x|\text{member}) + P(x|\sim \text{member})} \tag{6}$$

where $x = r, p, m$. As in Kim's work [2] we have also assumed conditional independence of these extracted parameters. Therefore the above 3-equations can be expressed as follows:

$$P(\text{member}|r,p,m) = \frac{P(r,p,m|\text{member})}{P(r,p,m|\text{member}) + P(r,p,m|\sim \text{member})} \tag{7}$$

where

$$P(r,p,m|\text{state}) = P(r|\text{state}) \times P(p|\text{state}) \times P(m|\text{state})$$

$$\text{where} \qquad \text{state} = \text{member}/\sim \text{member} \tag{8}$$

C. *Cluster tracking*

In order to track the same cluster continuously, we use both the Euclidean distance between the previous and the current positions of the cluster centers and the trajectory. In every frame once the feature points are tracked in next frame, it is assumed that the feature points to follow an equation of a straight line and the gradients of all the feature points are obtained. The likely gradient of the straight line the cluster center may follow is estimated to be the median of the obtained set of gradients. For each frame, feature points are detected and tracked in the next 20 frames to extract the necessary parameters to validate before clustering in the current frame. Thus the median speed of all the candidate feature points of a cluster is already computed. Therefore the the expected center of the cluster in the next frame can be estimated using the frame rate, current speed and the gradient of the cluster center. Then by thresholding, the same cluster is identified.

When computing the speeds of the feature points of an object, we can observe a range of different speeds within it. For instance, the speeds of the feature points appearing on the front end of the vehicle may be dissimilar to the speeds of the feature points appearing on the back-end of the same vehicle. This is mainly due to the perspective effect of the camera. Therefore to achieve robustness, the speed of the cluster center is considered to be the median of the speeds of the candidate feature points.

III. Results and Analysis

We track vehicles under two different scenarios, namely, when i) vehicles move in isolation, ii) vehicles move under partial occlusion. Using EM algorithm and an agglomerative hierarchical clustering algorithm the vehicle tracking results were obtained. The video footage used for the experiment were Kim's video clip [2], specific regions of interest of VIRAT video clips [12] and freshly obtained local footage. We have further improved Kim's foreground estimation technique by incorporating color information together with the conventional morphological operations. We have also introduced a single level agglomerative hierarchical clustering approach to directly cluster corner features using the most recent 20 frames.

The resultant object blobs obtained after incorporating color information together with the conventional morphological operations and after validating the foreground with the corner feature points are shown in the Fig. 4.

Fig. 4: Binary images represent the estimated foreground obtained by the improved foreground estimation technique indicating an accurate foreground estimation compared to the existing work.

In order to quantitatively determine the accuracy of the algorithm, the state of each pixel of a given frame is considered as a two-class prediction problem in which the outcomes are labeled as positive (P) for a pixel in the actual foreground and negative (N) for a pixel in the actual background. We have applied the foreground estimation algorithm on a set of completely different images. The population is considered as the total number of pixels of the set of images selected. The obtained confusion matrix is given below.

TABLE I: The confusion matrix obtained by considering the state of the pixel being foreground or background-the total number of pixels of the set of images considered is 2683029.

	Actual Class	
	Foreground (P)	**Background (N)**
Predicted Foreground	239057-(TP)	43431-(FP)
Predicted Background	6665-(FN)	2393876-(TN)

The foreground estimation algorithm estimates the foreground with a accuracy of 98% according to the confusion matrix. This indicates that when the video camera is fixed, given a frame, the developed color based foreground estimation technique is cable of estimating the actual foreground region with a accuracy of 98%. Our method assumes the position of the camera to be fixed in order to estimate the foreground region. Therefore when there is a slight movement of the camera, certain false foreground regions are generated.

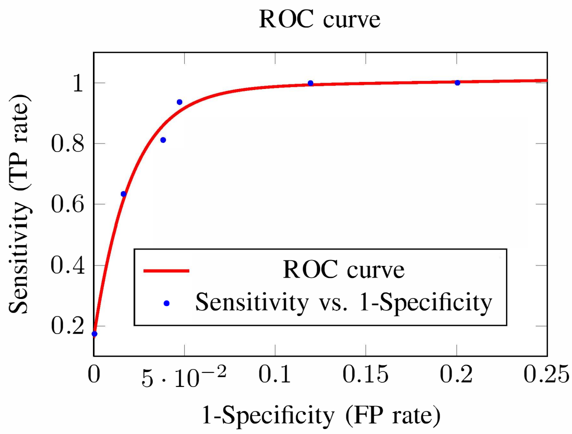

Fig. 5: When percentage color change compared to the previous background is thresholded to 30% the classification ability of the foreground estimation technique increases. Thus a better foreground estimation could be arrived at by applying simple threshold on the background subtraction result.

In foreground estimation, when the threshold applied on the obtained maximum percentage color change compared to the recent background update is varied, the classification ability of the foreground estimation algorithm could be evaluated using a ROC curve (Fig. 5). According to the ROC curve, closer a point to the top left corner, higher the correctness of the classification results generated by its threshold value. Therefore we have used 30% as the threshold value to decide whether the pixel belongs to the foreground or background region.

The results obtained by applying EM algorithm to cluster the validated cluster points when vehicles move in isolation is shown in Fig. 6. In order to achieve the clustering of feature points to separately identify an object, it is important to initialize the EM algorithm with a point close to the desired cluster. Therefore when vehicles move in isolation, we use blob centers to initialize the EM algorithm. In Fig. 6, the validated corner features and the obtained trajectories are shown.

Fig. 6: Trajectories obtained by the EM algorithm for a specific region of interest selected in a VIRAT video clip when vehicles move in isolation. Integer part of the displayed number represents the frame at which the vehicle is detected and the non integer part of the number indicates the assigned cluster number when it is detected. The obtained tracking result is good and is best viewed in color.

Although agglomerative hierarchical clustering considers each feature point to be a different cluster and pairs of clusters are merged along the hierarchy, it still requires a certain cut-off value to determine for what extent the clusters must be merged. In this implementation the number of blobs are used as the number of objects in a frame. If the size of the blob is large, it is eroded and the blob count is used to identify the number of objects per frame. In order to validate the number of blobs, another criteria has been applied as follows: When fresh clusters are formed, based on the trajectory and the speed of the cluster center, the likely position of the same cluster center appearing in the next frame can be computed. Therefore when new feature points are detected on the current frame, a search is performed to find out whether the detected and validated feature points could be assigned to the same cluster. Thus it allows previously detected clusters/number of objects to be carried forward to the next frame until the object disappears from the field of view of the camera. The results obtained through agglomerative hierarchical are shown below.

Fig. 7: Trajectories obtained after applying hierarchical clustering to a specific region of interest in Kim's video. The obtained tracking result is satisfactory and is best viewed in color.

In Fig. 7, the two vehicles initially appeared being partially occluded. But generating the feature matrix using the x and y position coordinates, speed and the trajectory has allowed the agglomerative hierarchical clustering approach to group the feature points directly in to objects as required.

In the computation of probabilities in Fig. 1 for each ratio -r and $\sim r$, the count is an integer ranging from [1-20]. But each validated feature point is tracked thrice and therefore when the ratios are obtained and normalized to [0-1] range and 20 bins are used, no values are observed in certain normalized r values corresponding to initial counts. In Fig. 2 the probabilities for each parameter -p and $\sim p$ are computed as follows: The values of minimum distance from the ellipse boundary are continuous and therefore once the minimum distance is obtained, it is normalized to a value that lies between [0-1]. The probability is assigned by checking to which binned range the normalized value belongs to and retrieving its corresponding probability value.

We have applied our tracking algorithms for several video clips and several selected regions of interest of VIRAT video clips [12]. All the vehicles that move in isolation are detected and tracked. But at certain times when vehicles move under partial occlusion, the hierarchical clustering algorithm failed simply because of not correctly estimating the number of objects per frame to cluster. In such circumstances, when the number of objects to cluster are manually adjusted, most of the time, the desired clustering is achieved.

IV. Conclusion and Future work

We have directly made an attempt to achieve object level clustering resulting in object tracking. For a given frame, cluster membership of the feature points are computed based on a probabilistic framework. We have tracked each feature point for next 20 frames to extract parameters. Then, we have assigned the probability of being a member by using the generated probability distributions. The implemented background model requires the background subtraction result and the detected and validated corner feature result to function. The color information of a pixel and the conventional morphological operations have been used to preserve the shape of the estimated silhouettes. According to the obtained confusion matrix the accuracy of our foreground estimation algorithm is 98%.

The validated corner features are clustered based on two approaches. When objects move in isolation, an EM algorithm is used to cluster and all the vehicles are tracked. When objects move under partial occlusion, the corner features are clustered based on an agglomerative hierarchical clustering approach and the tracking result is satisfactory. Since a more reliable estimate of the number of objects increases the ability to cluster, further work will be done in this respect in the future. If the number of objects and the center position estimate of the grouped feature points are estimated for each object, the application of EM algorithm could be sufficient to achieve the desired clustering required. As the EM clustering and the agglomerative hierarchical clustering algorithms are implemented off-line, we will continue our work to group the feature points using both of these grouping approaches under real time conditions in the future.

REFERENCES

[1] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 4, pp. 271–288, August 1998.

[2] Z. Kim, "Real time object tracking based on dynamic feature grouping with background subtraction." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2008, pp. 1–8.

[3] T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transations on Patteren Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631 – 1643, October 2005.

[4] N. Buch, J. Orwell, and S. Velastin, "3D extended histogram of oriented gradients (3DHOG) for classification of road users in urban scenes." Proceedings of the British Machine Vision Conference, September 2009, pp. 15.1–15.11.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 1997, pp. 731–737.

[6] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, December 2006, article 13.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," vol. 1. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2005, pp. 886 – 893.

[8] S. Birchfield, "Derivation of Kanade-Lucas-Tomasi tracking equation," January 1997.

[9] J. Shi and C. Tomasi, "Good features to track." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 1994, pp. 593–600.

[10] C. Bishop, *Pattern recognition and machine learning*. Springer Science Business Media, LLC, 2006.

[11] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[12] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, C., T. Lee, J., S. Mukherjee, K. Aggarwal, J., H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, R. Chowdhury, A., and M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2011, pp. 3153–3160.