IMAGE SEGMENTATION IN VIDEO SEQUENCES USING MODIFIED BACKGROUND SUBTRACTION

Mr. D. W. Chinchkhede[1] & Mr. N. J. Uke[2]

[1]SVC Polytechnic, Pune University, Pune, India.

dwchinch@gmail.com

[2]Sinhgad College of Engg., Pune University, Pune, India.

nilesh.uke@gmail.com

ABSTRACT

In computer vision, "Background subtraction" is a technique for finding moving objects in a video sequences for example vehicle driving on a freeway. For to detect non stationary (dynamic) objects, it is necessary to subtracting current image from a time-averaged background image. There are various background subtraction algorithms for detecting moving vehicles or any moving object(s) like pedestrians in urban traffic video sequences. A crude approximation to the task of classifying each pixel on the frame of current image, locate slow-moving objects or in poor image qualities of videos and distinguish shadows from moving objects by using modified background subtraction method. While classifying each pixel on the frame of the current image, it is to be detect the moving object at foreground and background conditional environment that we can classify each pixel using a model of how that pixel looks when it is part of video frame classes. A mixture of Gaussians classification model for each pixel using an unsupervised technique is an efficient, incremental version of Expectation Maximization (EM) is used for the purpose. Unlike standard image-averaging approach, this method automatically updates the mixture component for each video frame class according to likelihood of membership; hence slow-moving objects and poor image quality of videos are also being handled perfectly. Our approach identifies and eliminates shadows much more effectively than other techniques like thresholding.

1. INTRODUCTION

A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good background subtraction algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as moving leaves, rain, snow, and shadows cast by moving objects. Finally, its internal background model should react quickly to changes in background such as starting and stopping of vehicles [10]. Image segmentation is a process used to distinguish objects within images, such as photographs, radar outputs, or x-rays, from their background. Segmentation subdivides an image into its constituent regions or objects. That is, it partitions an image into distinct regions that are meant to correlate strongly with objects or features of interest in the image. Segmentation can also be regarded as a process of grouping together pixels that have similar attributes. The level to which the subdivision is carried depends on the problem being solved. That is, segmentation should stop when the objects of interest in an application have been isolated. Image segmentation methods can also be used to find and store three-dimensional objects by analyzing multiple images of the same objects from different

DOI : 10.5121/ijcsit.2012.4108

angles. Image segmentation is the problem of partitioning an image into its constituent components. In wisely choosing a partition that highlights the role and salient properties of each component, we obtain a compact representation of an image in terms of its useful parts. Depending on the end application, the problem of segmentation can be subjective or objective. Digital image processing is being used in many domains today. In image enhancement, for example, a variety of methods now exist for removing image degradations and emphasizing important image information, and in computer graphics, digital images can be generated, modified, and combined for a wide variety of visual effects. In data compression, images may be efficiently stored and transmitted if translated into a compact digital code.

2. SEGMENTATION OF OBJECTS IN IMAGE SEQUENCES

Images are segmented into objects to achieve efficient compression by coding the contour and texture separately. As the purpose is to achieve high compression performance, the objects segmented may not be semantically meaningful to human observers. The more recent applications, such as content-based image/video retrieval and image/video composition, require that the segmented objects be semantically meaningful. Finding moving objects in image sequences is one of the most important tasks in computer vision and image processing. Background subtraction approach means to compute the stationary background image and to identify the moving objects as those pixel in the image that differ significantly from the background. Background subtraction can provide an effective means of locating & tracking moving vehicles in freeway traffic. Moving shadow cause the serious problem, it differ from the background image, it is parts of moving object when traffic is slow moving or stationary, background image becomes corrupt by the vehicle themselves. To classify each pixel of each image as moving object, shadows and background. Basic idea is to classify each pixel using a probabilistic model of how that pixel looks when it a part of different classes. The probabilistic classification of the current pixel values can be used to update the model appropriately, so that vehicle pixel do not become mixed in with the background model when traffic is moving slow. The application of EM and related technique to image reconstruction, image segmentation and motion identification [2]. This is a work done with following basic objectives: to identify homogeneous regions in images as distinct from the background and belonging to different objects, and to cluster pixel into salient image regions, i.e. region corresponding to individual surfaces, objects, or natural parts of object. Frame differencing and adaptive median filter are the simple technique, to more sophisticated probabilistic modeling technique. Pre and post processing of the video might be necessary to improve the detection of moving objects. Image segmentation’s goal is to identify homogeneous region in images as distinct from background and belonging to different objects. A common approach is to classify pixel on the basis of local features (e.g. color, position, texture), and then group them together according to their class in order to identify different objects in the scene. For the specific problem of finding moving objects from static cameras, the traditional segmentation approach is to separate pixels into two classes: background and foreground. This is called Background Subtraction[1] and constitutes an active research domain; the interested reader is referred to[2] for an overview of the field’s state of the art. The output of most background segmentation techniques consists of a bitmap image, where values of 0 and 1 correspond to background and foreground, respectively ([3], [4],[5]). Having such a bitmap, the next processing step consists of merging foreground pixels to form bigger groups corresponding to candidate objects; this process is known as object extraction. One common procedure to perform object extraction consists of finding 4 or 8-connected components. This is done using efficient algorithms whose time complexity is linear with respect to the number of pixels in the bitmap [6],[7]. There are many challenges in developing a good background subtraction algorithm such as robust against changes in illumination and avoid detecting non-stationary background objects such as swinging leaves, rain, snow, and shadow cast by moving objects. Intension of different background subtraction algorithms is use to detecting, tracking moving vehicles and pedestrians in traffic video sequences. Nearby stray pixels identified as "moving" due to image noise, group the pixels belonging to each vehicle and to compute and track a smoothed convex hull. Background subtraction method succeeds not only in detecting moving vehicles, but also their shadows. Shadows is the most serious problems for video-based traffic surveillance. Slow moving means that the time of traversal is non-negligible compared with the 1/α. The time constant of expontial forgetting process. When this happen the background image becomes corrupted and object detection fail completely. By using a probabilistic classifier and a stable updating algorithm problem will be solved.

$$B(x, y, t)= (1 - \alpha) B(x, y, t - 1) + \alpha I (x, y, t) \quad .....(1)$$

Parameter is I(x, y, t)- instantaneously pixel value, (x, y) pixel at time t, B (x, y, t)- background, 1/α -time constant forgetting process, variance can be computed. Moving object can be identified by thresholding is the distance between I(x,y,t) & B(x,y,t).

3. PIXEL MODELS

Consider a single pixel and the distribution of its values over time. Some of the time it will be in its "normal" background state-for example, a small area of the road surface. Some of the time it may be in the shadow of moving vehicles, and some of the time it may be part of a vehicle. Thus, in the case of traffic surveillance, we can think of the distribution of values $I_{x,y}$ of a pixel (x,y) as the weighted sum of three distributions $R_{x,y}$(road), $S_{x,y}$(shadow), and $V_{x,y}$ (vehicle):

$$I_{x,y} = W_{x,y}. (r_{x, y}, S_{x, y}, V_{x, y}) \quad ..... (2)$$

These distributions are subscripted to emphasize that they differ from pixel to pixel; $R_{x,y}$ is a probability distribution for the way that this specific pixel looks when it is showing unshadowed road at the corresponding physical location. It is essential to have different models for each pixel, because, for example, some parts of the image may correspond to white road markings, others to dark streaks in the centers of lanes, and still. Some of the features used while detecting the moving object such as intensity, color, shape of the region texture, motion in video, display in stereo image, depth in the range Camera temperature in Far infrared, mixture relation between region and stereo disparity.

4. ALGORITHMS FOR LEARNING PIXEL MODELS

EM Algorithm for learning mixture models from observed data with the class variable hidden. EM algorithm is a method for finding maximum likelihood. EM is an iterative method which alternates between performing an expectation (E) step, which computes the expectation of the log- likelihood evaluated using the current estimate for the parameters, and performing a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E Step. These parameter-estimates are then used to determine the distribution of the latent variable in the next E step. In the incremental EM, describe an efficient, incremental version suitable for to satisfy real-time implementation.

5. EM FOR MIXTURE MODELS

The goal of maximum likelihood is to adjust a statistic model with a specific data set estimating its unknown parameters so the function that can describe all the parameters in the dataset. EM algorithm is an unsupervised clustering method. It follows an iterative approach, sub-optimal, which tries to find the parameters of the probability distribution that has the maximum likelihood of its attributes. A sequence of pictures 1,...,T, and that **I**(x, y, t) is the value of pixel (x, y) in the t^{th} image. We want to learn the parameters of the distributions **R**x,y, **S**x,y, **V**x,y as well as the relative weights **W**x,y. To formally set up the learning problem, we define the likelihood of a set of parameters Θ to be the probability of the data given Θ:

$$\mathbf{\Pi\ ^{T}{}_{t=1}\ P(L = l_{t}\,,\, I(x,\, y,\, t)\mid \Theta)} \qquad \text{.....}\ (3)$$

Suppose that the images were annotated by the labels of all pixels, and suppose that L_t is the label of x, y in the t^th image. In this case, learning these parameters would be easy. Standard arguments show that the optimal settings of parameters for this case can be computed as follows. We define the sufficient statistics for this mixture estimation to be N_l, M_l, and Z_l, where N_l is the number of images for which Lt = l, M_l is the sum of the input vectors, $\sum_{t=l,...,T,\ Lt=1}$ **I**(x, y, t), and Z_l is given by $^{1}/_{T} \sum_{t=l,...,T,\ Lt=1}$ **I**(x, y, t) **.** **I**(x, y, t)^T , the sum of the outer products of the input vectors with themselves. From these sufficient statistics, we can compute ω_1 , μ_l , and $\sum_1$ as:

$$\omega_1 = {}^{Nl}/\textstyle\sum_{l'} N_{'1} \qquad \text{.....}\ (4)$$

$$\mu_l \ = \ M_l / N_l \qquad \text{.....}\ (5)$$

$$\textstyle\sum_1 \ = \ (^1/N_l)Z_l - \mu^{T}{}_{1}\mu_l \qquad \text{.....}\ (6)$$

We define the likelihood with respect to the observable data: $L(\Theta) = \Pi^{T}{}_{t=1}\ P(I(x,\ y,\ t) / \Theta)$. Learning mixture models is one of the classic examples of missing values. The standard solution in the literature is the EM algorithm [8]. Roughly speaking, the EM algorithm explores a sequence of parameter settings, where each setting is found by using the previous one to classify the data. More precisely, assume that we have some parameter setting Θ^k. We can use the probability of different labels according to Θ^k as an estimate of their true distribution. We now re-estimate the parameters, where we "split" each vector I(x, y, t) between the different Gaussians according to these proportions. Formally, we compute the expected value of the sufficient statistics as follows:

$$E[N_l / \boldsymbol{\Theta}^{\mathbf{k}}] = {}^{T}\Sigma_{n=1}\ P(L_{t=l} / I(x,y,t),\ \boldsymbol{\Theta}^{\mathbf{k}})$$

$$E[S_l / \boldsymbol{\Theta}^{\mathbf{k}}] = {}^{T}\Sigma_{n=1}\ P(L_{t=l} / I(x,y,t),\ \boldsymbol{\Theta}^{\mathbf{k}})\ I(x,y,t)$$

$$E[Z_l / \boldsymbol{\Theta}^{\mathbf{k}}] = {}^{T}\Sigma_{n=1}\ P(L_{t=l} / I(x,y,t),\ \boldsymbol{\Theta}^{\mathbf{k}})\ I(x,y,t)\ .\ I(x,y,t)^{T}$$

We then define Θ^k by using Equations 4-6 with the expected sufficient statistics. This process has two important properties. First, L(Θ^{k+1}) ≥ L(Θ^k). That is, Θ^{k+1} provides a better approximation to the distribution of the data. Second, if $\Theta^{k+1} = \Theta^k$, then Θ^k is a stationary point (e.g., a local maximum) of the likelihood function. Combining these two properties, we see that this procedure will eventually converge to a stationary point [8]. If we start the process several times with different setting of Θ^o, we hope to find a good approximation to the optimal setting. In addition, for our application, we must have a way to identify which component of the model should be labeled as road, which as shadow, and which as vehicle.

6. INCREMENTAL EM

From previous observation of EM procedure, it is necessary to reviewed requires us to store the values of pixel (x, y) for all the images we have observed. This is clearly impractical for our application. Moreover, batch processing of the complete image sequence is not possible in a real-time setting. We now describe an incremental variant of EM that does not require storing the data. This procedure was introduced by Nowlan [10], and is best understood in terms of the results of Neal and Hinton [9]. Neal and Hinton show that we can think of the EM process as continually adjusting the sufficient statistics. Here, on each iteration when we process an instance, we remove its previous contribution to the sum and replace it with a new one. For example, when we update N_l, we remove **P**(L_t = l | **I**(x, y, t), $\Theta^{k'}$) and add **P**(L_t = l | **I**(x, y, t),

Θ^{k}), where $\Theta^{k'}$ are the parameter settings we used to compute the previous estimated statistics from $\mathbf{I}(x, y, t)$, and Θ^{k} are the current parameter settings. Similarly, to update M_l, we remove $P(L_t = l \mid \mathbf{I}(x, y, t), \Theta^{k'})I(x, y, t)$ and add $P(L_t = l \mid \mathbf{I}(x, y, t), \Theta^{k})I(x, y, t)$. Neal and Hinton show that after each instance is processed, the likelihood of the data increases. This argument motivates the following incremental approach. Whenever we observe a new instance, we add its contribution to the sufficient statistics. This means that we are increasing our training set at each step, yet we never reprocess the previous instances in the training set. This procedure is no longer guaranteed to be monotonic in *L,* but on the average this process increases *L.* Thus, in the long run, this process converges to a local maximum with high probability. The resulting procedure for each pixel *(x, y)* has the following structure: Initialize parameters Θ.

```
$t \leftarrow O$
for $l \in \{r, s, v\}$        $N_l \leftarrow k\omega_l$
                             $S_l \leftarrow k\omega_l . \mu_l$
                             $Z_l \leftarrow k\omega_l . (\sum_l + \mu_l . \mu_l^T)$
                             do forever  $t \leftarrow t+ 1$
for $l \in \{r ,s, v\}$        $N_L \leftarrow N_L + P(L_t = l \mid \mathbf{I}(x, y, t), \Theta)$
                             $S_l \leftarrow S_l + P(L_t = l \mid \mathbf{I}(x, y, t), \Theta)I(x, y, t)$
                             $Z_l \leftarrow Z_l + P(L_t = l \mid \mathbf{I}(x, y, t), \Theta)I(x, y, t)I(x, y, t)^T$
```

Compute Θ from $\{N_L, S_l, Z_l\}$.

The initialization step of the procedure sets the statistics to be the expected statistics for the initial choice of Θ. Then, in each iteration we add the expected statistics for the new instance to the accumulated statistics. This procedure performs quite well, and the reported experiments in empirical result are based on it. However, since the sufficient statistics terms keep growing, this procedure can run into problems in the long run. Moreover, the procedure never removes the contributions of old instances from the sufficient statistics. Intuitively, the models that were used to compute the expectation for these instances are quite out of date. Thus, the procedure would perform better if these instances were removed.

7. PROPOSED SYSTEM ARCHITECTURE FOR IMAGE SEGMENTATION

The proposed System architecture for image segmentation using background subtraction and EM technique shown in below figure.

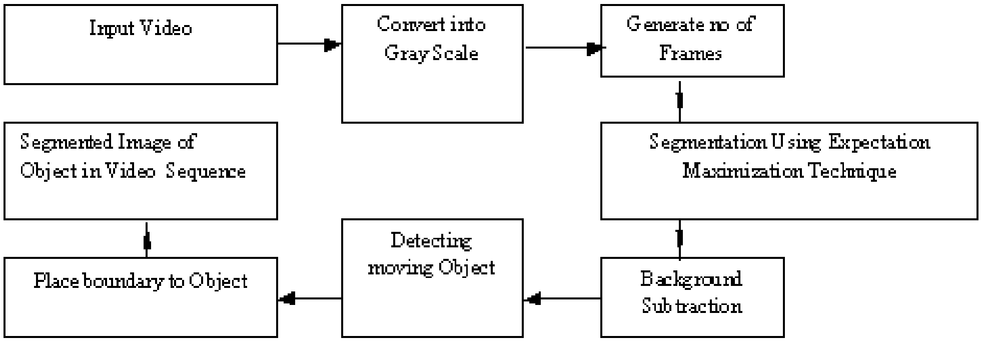


Figure 1 System Architecture

When execute program, input video called for to detect moving object. By using RGB2GRAY predefine function convert colorful video into gray color. The model can be used to detect a moving object in a video. The method generate motion image from consecutive pair of frame. Object detected in video frames and motion images. From local windows a neighborhood pixel around background and extract features. Features and background are used to construct and maintain a model, stored in a memory of a computer system. By using EM, each moving object image is a difference between two consecutive frames in the input video. The next step construct and maintain a background model over time for effective moving object detection. Here, the image matrix is generated very sparse and more information is maintain for each matrix entry. A frequency count is incremented when a corner is detected at a particular location in a frame. A mean vector of the 256 dimension descriptor is generated. The mean vector is used to identify a foreground corner. The mean and variance of the correlation also monitored over time. For a 352x240 resolution video sequence, about 1,00,000 pixels are processed in each frame when a pixel level background model is used. A 10x10 rectangle is placed around each motion corner in five consecutive frames i.e. the 3D block filter is 10x10x5 pixel. At last segmented image of object in video sequence. The K-means algorithm is an iterative technique that is based to partition an image into K Cluster. The basic algorithm is:

1. Pick K cluster centers, either randomly or based on some heuristic.
2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center
3. Re-compute the cluster centers by averaging all of the pixels in the cluster.
4. Repeat steps 2,3 until convergence is attained (e.g. no pixel change clusters)

8. EMPIRICAL RESULTS

Our general procedure for processing a video sequence is as follows: Initialize mixture models for each pixel with a weak prior. For each frame: (a) Update the estimated mixture model for each pixel using incremental EM (b) Heuristically label the mixture components, (c) Classify each pixel according it's current mixture model. The heuristic labeling process is needed in cases where the mixture components are not in the same order as the prior model indicates. For example, the prior model expects that vehicle pixels will be brighter, in general, than road pixels. For white road markings, the brightness order is reversed.

Our heuristics are as follows: label the darkest component as shadow; of the remaining two components, label the one with the largest variance as vehicle and the other as road. Sample results are shown in Figure 4; we show the original image and the identified shadow pixels, together with an image resulting from replacing shadow pixels with the corresponding road value.

The mask image showing the vehicle pixels are much cleaner than that obtained by background subtraction. At the time of writing, we are beginning the experiments needed to show that this improvement carries over into vehicle detection and tracking performance. We are also rerunning our experiments with RGB models instead of intensity models; we assume that use of color information will largely eliminate the tendency of darker vehicles to disappear, since very few vehicles have the same hue as the road, even if they have the same intensity values.

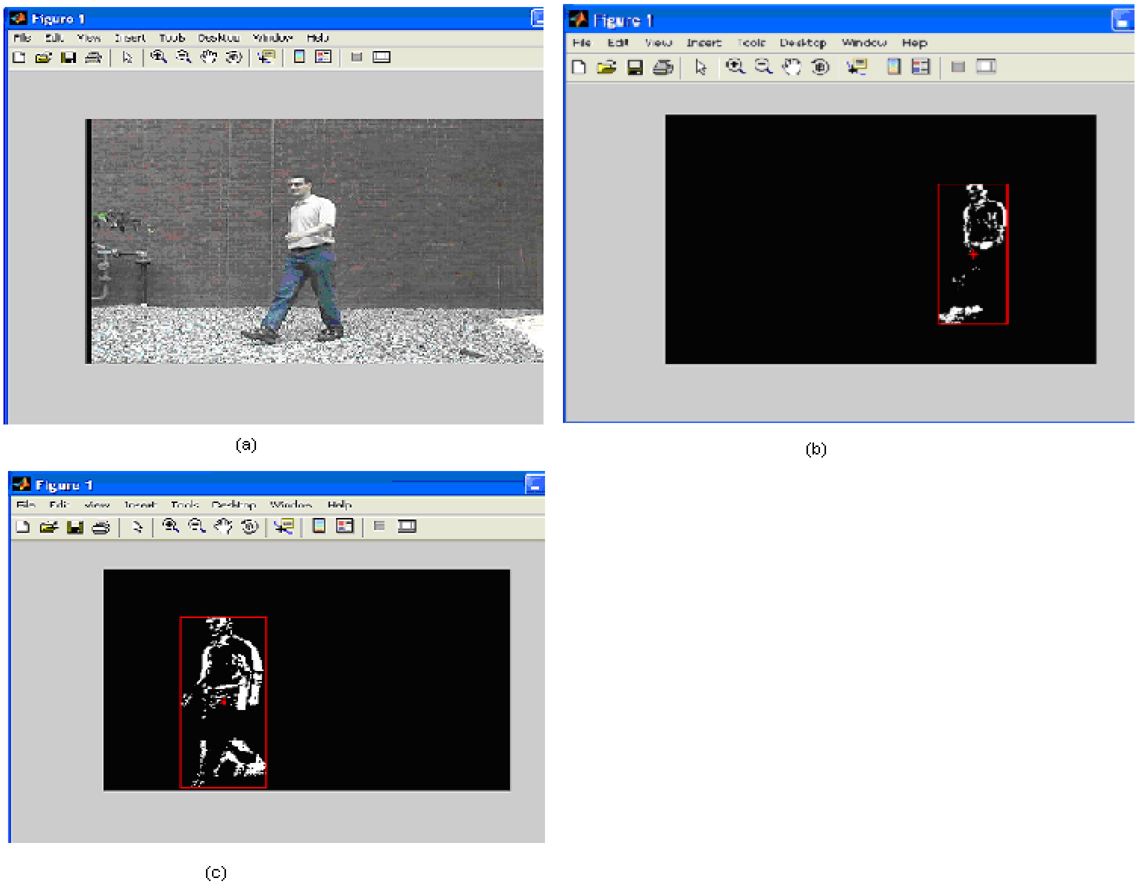


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 1
File Edit View Insert Tools Desktop Window Help

(a)

(b)

(c)

Figure 2 (a) Original Input Video 1, (b) Output Video where Moving Object is Tracked 1, (c) Moving Object Tracked at the end after Subsequent Video Frames.

9. APPLICATIONS AND DRAWBACKS

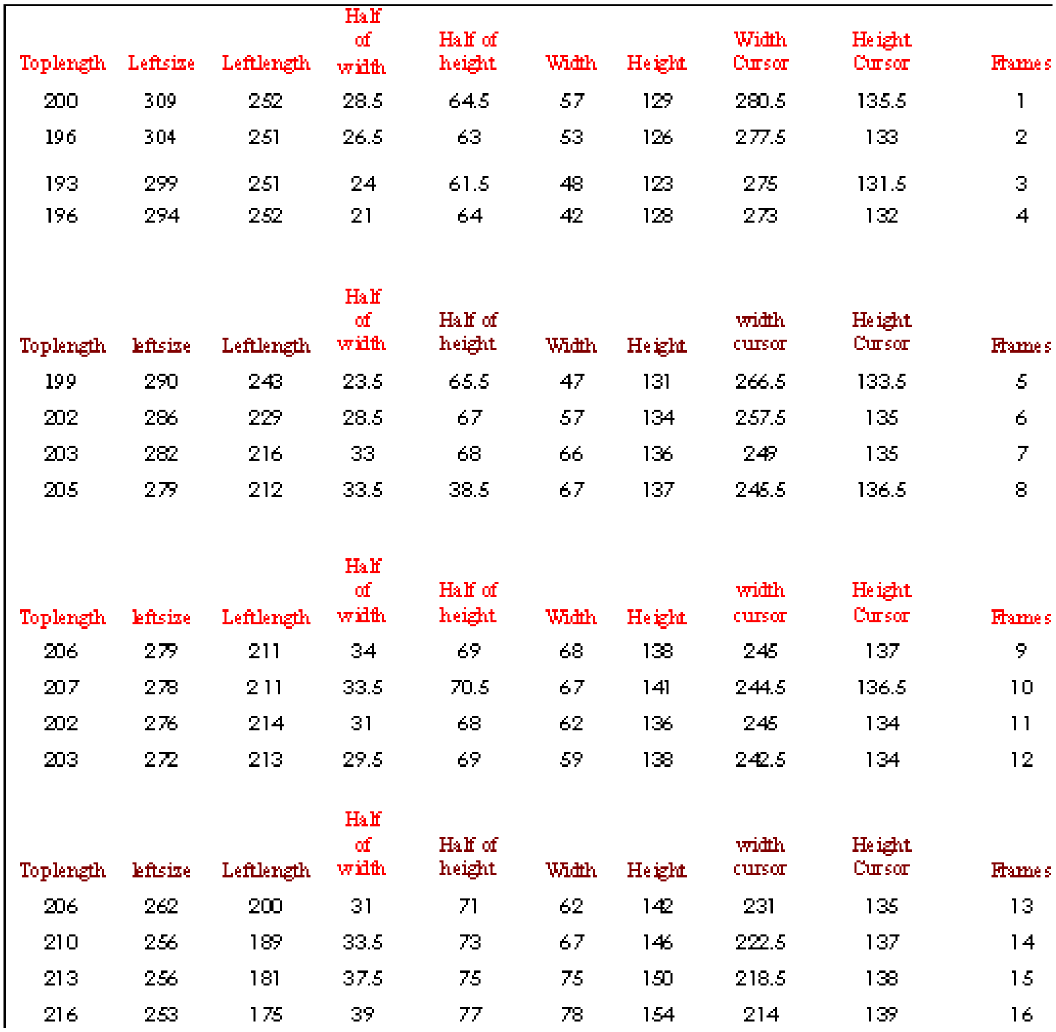
9.1 APPLICATIONS:

1. Automated video surveillance: In these applications computer vision system is designed to monitor the movements in an area, identify the moving objects and report any doubtful situation. The system needs to discriminate between natural entities and human, which require a good object tracking system.
2. Robot vision: In robot navigation, the steering system needs to identify different obstacles in the path to avoid collision. If the obstacles themselves are other moving objects then it calls for a real-time object tracking system.
3. Traffic monitoring: Traffic is continuously monitored using cameras. Any vehicle that breaks the traffic rules or is involved in other illegal act can be tracked down easily if the surveillance system is supported by an object tracking system.
4. Animation: Object tracking algorithm can also be extended for animation.
5. Motion capture in sports & tracking of multiple Human in Crowded Environments.

6. Magnetic Resonance Imaging (MRI): Doctors can get highly refined images of the body's interior without surgery. MRI is particular useful for imaging the brain and spine, as well as the soft tissues of joints and the interior of bones.

7. Positron Emission Tomography (PET): It is a scanning technique used in conjunction with small amount of radio labeled compounds to visualize brain anatomy and function.

Table 1: Attribute values of single moving Object

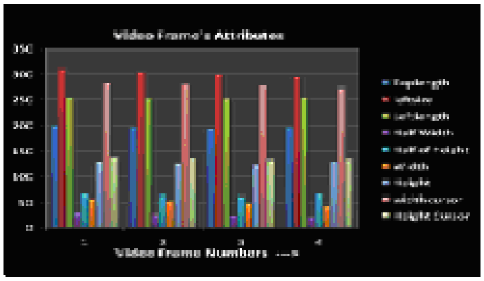


Toplength	Leftsize	Leftlength	Half of width	Half of height	Width	Height	Width Cursor	Height Cursor	Frames
200	309	252	28.5	64.5	57	129	280.5	135.5	1
196	304	251	26.5	63	53	126	277.5	133	2
193	299	251	24	61.5	48	123	275	131.5	3
196	294	252	21	64	42	128	273	132	4

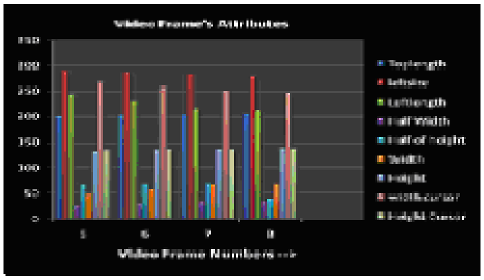
Toplength	leftsize	Leftlength	Half of width	Half of height	Width	Height	width cursor	Height Cursor	Frames
199	290	243	23.5	65.5	47	131	266.5	133.5	5
202	286	229	28.5	67	57	134	257.5	135	6
203	282	216	33	68	66	136	249	135	7
205	279	212	33.5	38.5	67	137	245.5	136.5	8

Toplength	leftsize	Leftlength	Half of width	Half of height	Width	Height	width cursor	Height Cursor	Frames
206	279	211	34	69	68	138	245	137	9
207	278	211	33.5	70.5	67	141	244.5	136.5	10
202	276	214	31	68	62	136	245	134	11
203	272	213	29.5	69	59	138	242.5	134	12

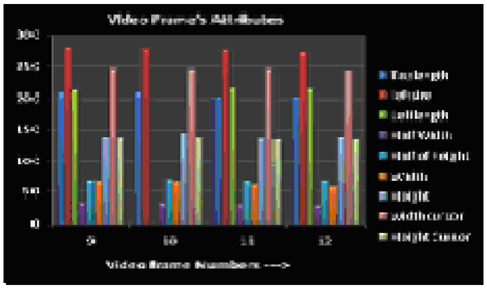
Toplength	leftsize	Leftlength	Half of width	Half of height	Width	Height	width cursor	Height Cursor	Frames
206	262	200	31	71	62	142	231	135	13
210	256	189	33.5	73	67	146	222.5	137	14
213	256	181	37.5	75	75	150	218.5	138	15
216	253	175	39	77	78	154	214	139	16



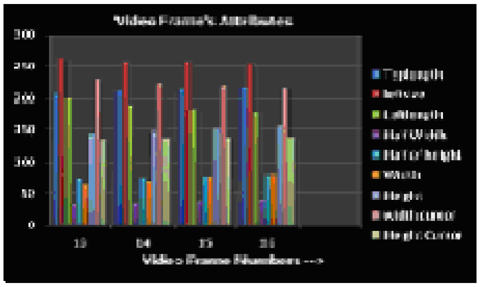
Graph 1 Showing variations in attribute values in frames No's 1 to 4



Graph 2 Showing variations in attribute in frames No's 5 to 8



Graph 3 Showing variations in attribute values in frames No's 9 to 12



Graph 4 Showing variations in attribute values in frames No's 13 to 16

Graph Shows variation in attributes values in frames
No's 1 to 16 (group of 4 frame)

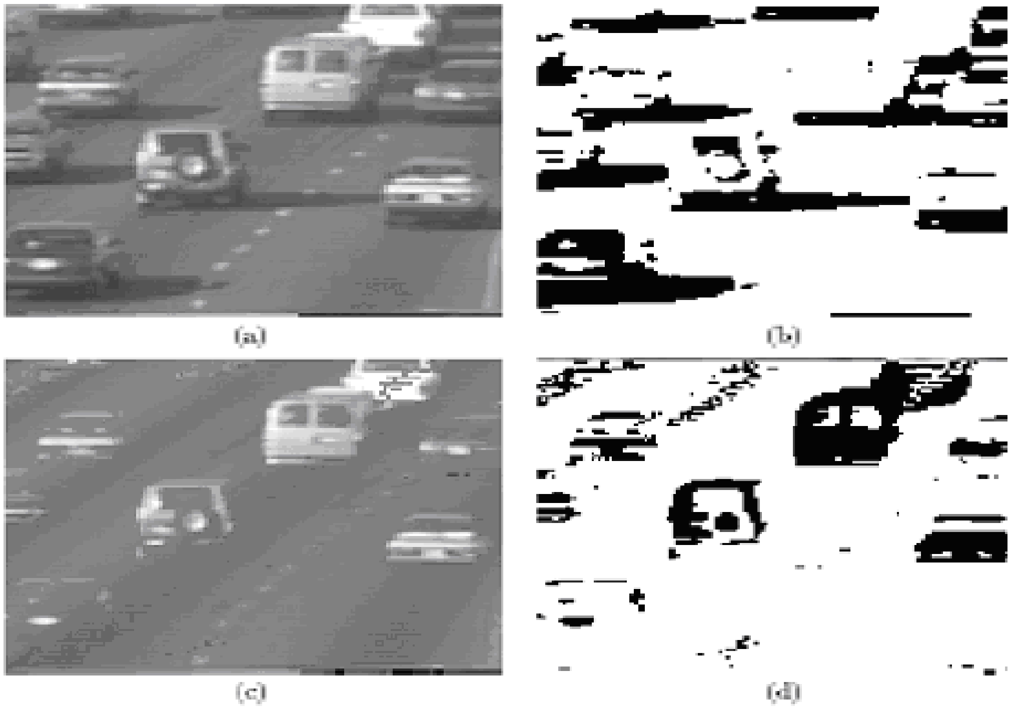

(a)
(b)
(c)
(d)

Figure 3 (a) Original image, (b) Pixels identified as shadow, (c) Image with shadow pixels replaced by corresponding road value, (d) Mask showing pixels classified as vehicles.

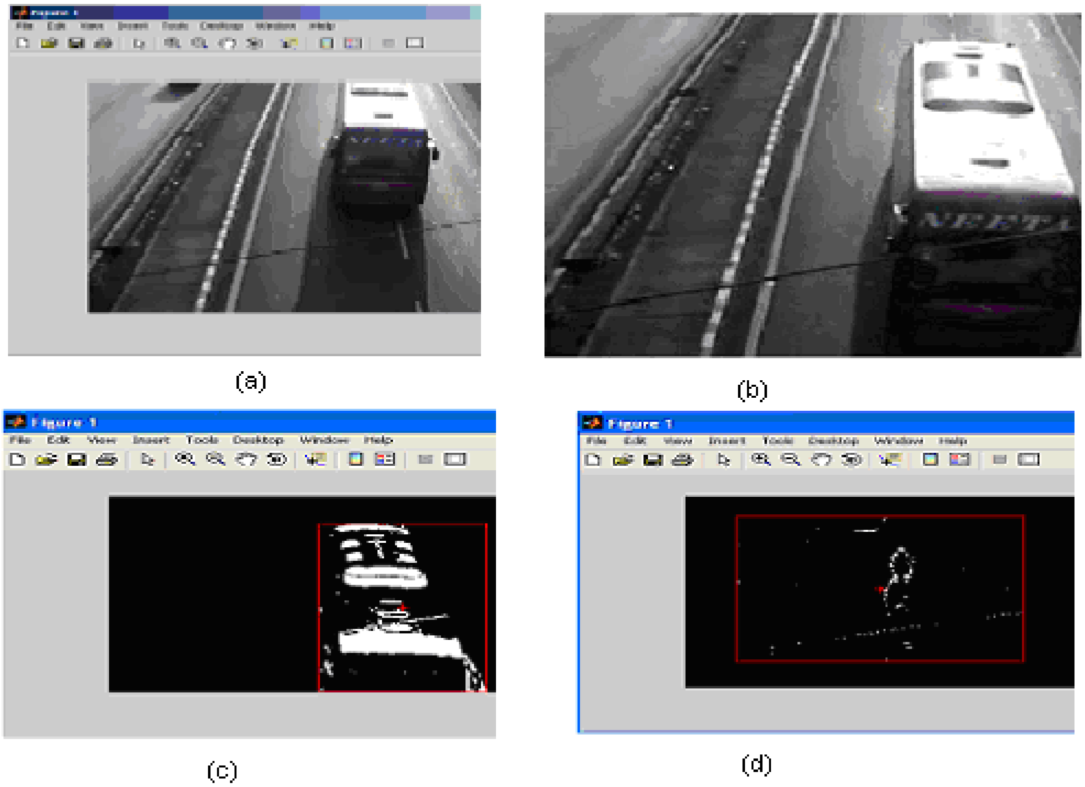
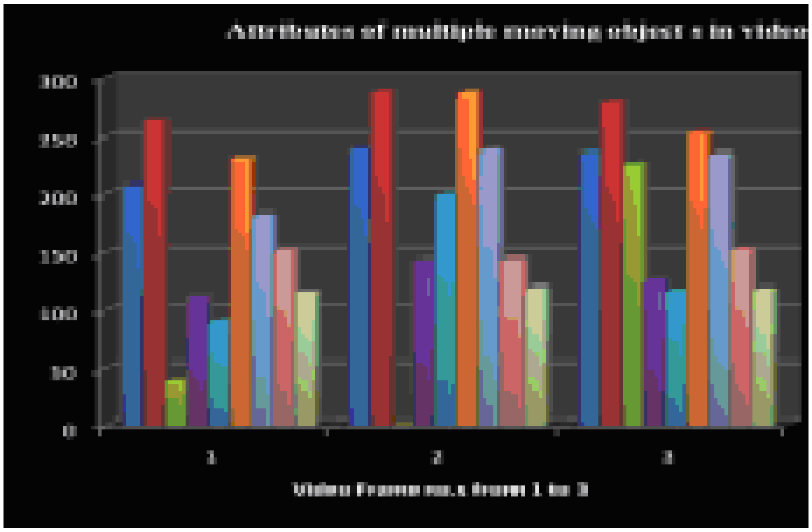


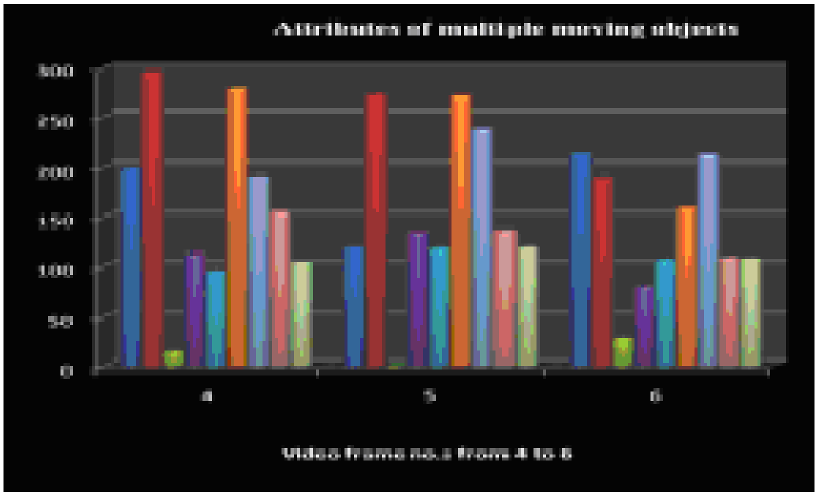
Figure 4 (a) Original Video Image (b) Tracking single Moving Object from Original Video Image (c) Output Video where single Moving Object Tracked from Multiple Moving Object. (d) Output Video where Multiple Moving Object is Tracked

Table 2: Attribute values of multiple moving Object

Frame	Top	Left	Left	Half	half	Width	Height	Width	Height
No.s	Length	Size	Length	Width	Height			cursor	cursor
1	208	264	41	111.5	91.5	233	183	152.5	116.5
2	240	289	1	144	199.5	288	239	145	120.5
3	237	281	226	127.5	118	255	236	153.5	119
Frame	Top	Left	Left	Half	Half	Width	Height	Width	Height
No.s	Length	Size	Length	Width	Height			cursor	cursor
4	199	296	16	114	95.5	280	191	156	103.5
5	120	272	1	135.5	119.5	271	239	136.5	120.5
6	215	190	29	80.5	107	161	214	109.5	108



Graph 5 Showing variations in attribute values of multiple moving object in frames No's 1 to 3



Graph 6 Showing variations in attribute values of multiple moving object in frames No's 4 to 6

9.2 DRAWBACKS:

1. The size of video should not be more than 550 KB. As we analyze video frame by frame, if the size is larger than the number of frames will also increase which required more main memory (RAM) for computation.
2. In MATLAB Video can be read only in the .avi format.

10.CONCLUSION AND FUTURE SCOPE:

Image segmentation for video sequencing has been implemented using EM and background subtraction methods. Also additionally moving object in video sequences has been detected. Future Scope is: it can be extend for extract moving object from original video sequence to any new windows, its processing function or images are avail only in .avi format. So there is a scope to convert original video images format from .avi to any other image formats like .mp3, .3gb, .dat etc, the video should be converted into gray scale. If we want process video of color image i.e. RGB space then more computations is involved in processing. Several improvements are needed in our system before it can be fielded. The most important from the point of view of robustness is the need for better initialization and labeling of models. Our heuristic approach may not work well in extremes of lighting conditions. We expect to reach the limits of pixel-based techniques fairly soon. Using more background knowledge, encoded as probabilistic models, one can expect much better performance. Temporal contiguity in a pixel's classification can be enforced using a simple Markov model in which any given classification has a high probability of persisting. Similar schemes can be used for spatial contiguity, but these require Markov networks that impose a high computational burden. In any case, this level of models should interface nicely with, the higher-level dynamic belief network models of vehicle behavior used in [4] to predict motions and to detect events such as stalled vehicles and accidents. Several researchers have explored and implemented different approaches for tracking. The success of a particular approach depends largely on the problem domain. In other words, a method that is successful in robot navigation may not be equally successful in automated surveillance. Further there exists a cost/performance trade off. For real time applications we may need a fast high performance system on the other hand offline applications we may use a relatively cheap (and slower in performance). It can also be seen from the diverse nature of the techniques used that the field has a lot of room for improvement.

REFERENCES:

[1] Dengsheng Zhang and Guojun Lu, '*Segmentation in moving object in imagesequence :A review*', Circuit system signal processing,Vol.20,N0.2,2001,Pp.143-183

[2] Nir Friedman, Stuart Russell, '*Image segmentation in video sequences :A probabilistic approach*', University of California, Berkeley, CA 94720-1776.

[3] Dempster 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.

[4] T.Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. '*Automatic symbolic traffic scene analysis using belief networks*'. In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pages 966-972, Seattle, Washington, August 1994.

[5] M.J. Jepson, A.; Black. '*Mixture models for optical flow computation'*, in CVPR pages 760-1, 1993.

[6] M.Kilger '*A shadow handler in a video-based real-time traffic monitoring system'*, IEEE Workshop on Applications of Computer Vision, 1992.

[7] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. '*Towards robust automatic traffic scene analysis in real-time*', in Proceedings of the International Conference on Pattern Recognition, Israel, November, 1994.

[8] G.J.McLachlan & T. Krishnan,'*The EM Algorithms & Extensions*', Wiley Interscience, 1997.

[9] R. M. Neal and G. E. Hinton, '*A new view of the EM algorithm that justifies incremental and other variants*', unpublished manuscript, 1993.

[10] S. J. Nowlan, '*Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures*', PhD thesis, School of Computer Science, Carnegie Mellon University, 1991.