

Computer Vision Algorithms for Intersection Monitoring

Harini Veeraraghavan, Osama Masoud, Nikolaos Papanikolopoulos, *Senior Member, IEEE**

{harini, masoud, npapas}@cs.umn.edu

Artificial Intelligence, Vision and Robotics Lab

Department of Computer Science and Engineering

University of Minnesota

Abstract— **The goal of this project is to monitor activities at traffic intersections for detecting/predicting situations that may lead to accidents. Some of the key elements for robust intersection monitoring are camera calibration, motion tracking, incident detection, etc. In this paper, we consider the motion-tracking problem. A multi-level tracking approach using Kalman filter is presented for tracking vehicles and pedestrians at intersections. The approach combines low-level image-based blob tracking with high-level Kalman filtering for position and shape estimation. An intermediate occlusion-reasoning module serves the purpose of detecting occlusions and filtering relevant measurements. Motion segmentation is performed by using a mixture of Gaussian models as in [19] which helps us achieve fairly reliable tracking in a variety of complex outdoor scenes. A visualization module is also presented. This module is very useful for visualizing the results of the tracker and serves as a platform for the incident detection module.**

Keywords— **motion segmentation, vehicle tracking, incident detection, camera calibration, occlusion reasoning.**

1 Introduction

Incident monitoring in outdoor scenes requires reliable tracking of the entities in the scene. In this project, we are interested in monitoring incidents at an intersection. The tracker should not only be able to handle the inherent complexities of an outdoor environment, but also the complex interactions of the entities among themselves and with the environment.

This paper combines low-level tracking (using image elements) with higher level tracking to address the problem of tracking in outdoor scenes. Reliable tracking requires that the tracked target

can be segmented out clearly. This can be done either using models that describes the appearance of the target or a model describing the appearance of the background. For our case of outdoor vehicle tracking, where the tracked vehicles are unknown and quite variable in appearance (owing to the complexity of the environment), it is easier to build models for the background (which is relatively constant). The model should be able to capture the variations in appearance of the scene due to changing lighting conditions. It should also be able to prevent foreground objects from being modeled as background (e.g., the slow stop and go motion of vehicles in crowded intersections). A poor model of the background results in effects like “ghosting” as shown in Figure 1.

Tracking based on blobs (segmented foreground) though extremely computationally efficient, results in significant loss in information regarding the tracked entities due to its simplified representation. This has an outcome in tracking difficulties due to the target-data association problem. We show that tracking can be improved significantly through more reliable data association, by integrating cues from the image with the estimated shape and the motion of the tracked target itself. We use oriented bounding boxes as opposed to axis aligned boxes which captures information about the orientation of the blobs giving a much tighter fit than the conventional axis aligned boxes. This is illustrated in Figure 2. Higher level models of the target that capture its motion and shape across frames are constructed. A Kalman filter is used for this purpose. Although several methods exist for modelling based on data, Kalman filters provide one of the best ways for doing real-time online prediction and estimation.

The low-level module which consists of blob tracking interacts with the image-processing module. The results from this level (tracked blobs) are passed onto the high-level where blobs are inter-

*author to whom all correspondences should be sent.



(a) Approximated image of background

(b) Current image

Figure 1: Approximated background and current image. The background shows a long trail of the bus as the bus was modeled into the background when it stopped.



(a) Oriented Bounding Box vs axis aligned box

Figure 2: The oriented bounding boxes provide much closer fit to the vehicles than axis aligned boxes.

interpreted as Moving Objects (MOs). Shape estimation consists of estimating the dimensions of the bounding box and the position of one corner point with respect to the blob centroid. The results from the shape estimator are used for occlusion reasoning. A visualization tool has been developed for visualizing the results of the tracking and the incident detection module.

The paper is arranged as follows: Section 2 discusses the problem and the motivation for this work. Section 3 discusses the related work in this area. The general tracking approach is discussed in Section 4. The Segmentation method is discussed briefly in Section 5. Section 6 discusses blob tracking, moving object tracking and Kalman filtering. Occlusion reasoning is presented in Section 7. The incident detection module and camera calibration are discussed in Section 8 and Section 9, respectively. Section 10 presents our results followed by discussion and conclusions in Sections 11 and 12.

2 Intersection Collision Prediction Problem

Intersection monitoring is an important problem in the context of Intelligent Transportation Systems (ITS). A real-time scene monitoring system capable of identifying situations giving rise to accidents would be very useful. Real-time incident detection would require robust tracking of entities and projecting the present state of scene to future time reliably and identification of colliding entities. The scope of this paper is concerned with a real-time vision based system for tracking moving entities. Reliable prediction requires very robust tracking. Achieving robust tracking in outdoor scenes is a hard problem owing to the uncontrollable nature of the environment. Furthermore, tracking in the context of an intersection should be able to handle non free-flowing traffic and arbitrary camera views. The tracker should also be capable of handling the large number of occlusions and interactions of the entities with each other in the scene reliably.

3 Related Work

3.1 Segmentation

Commonly used methods for motion segmentation such as static background subtraction work fairly well in constrained environments. These methods, though computationally efficient, are not suitable for unconstrained, continuously changing environments. Median filtering on each pixel with thresholding based on hysteresis was used by [18] for building a background model. A single Gaussian model for the intensity of each pixel was used by [22] for image segmentation in relatively static indoor scenes. Alternatively, Friedman *et al.* [8] used a mixture of three Gaussians for each pixel to represent the foreground, background, and shadows using an incremental, expectation maximization method. Stauffer *et al.* [19] used a mixture of Gaussians for each pixel to adaptively learn the model of the background. Non-parametric kernel density estimation has been used by [7] for scene segmentation in complex outdoor scenes. Cucchiara *et al.* [5] combined statistical and knowledge-based methods for segmentation. A median filter is used for updating the background model selectively based on the knowledge about the moving vehicles in the scene. Ridder *et al.* [16] used an adaptive background model updated using the Kalman filter. In [10], a mixture of Gaussians model with online expectation maximization algorithms for improving the background update is used.

3.2 Tracking

A large number of methods exist for tracking objects in outdoor scenes. Coifman *et al.* [4] employed a feature based tracking method for tracking free flowing traffic using corner points of vehicles as features. The feature points are grouped based on the common motion constraint. Heisele *et al.* [9] tracked moving objects in colored image sequences by tracking the color clusters of the objects. Other tracking methods involve active contour based tracking, 3-D model based tracking, and region tracking.

A multi-level tracking scheme has been used in [6] for monitoring traffic. The low-level consists of image processing while the high-level tracking is implemented as knowledge-based forward chaining production system. McKenna *et al.* [14] performed three level tracking consisting of regions, people and groups (of people) in indoor and outdoor environments. Kalman filter based feature tracking for predicting trajectories of humans was implemented by [17]. Koller *et al.* [11] used a tracker based on two linear Kalman filters, one for estimating the position and the other for estimating the shape of the vehicles moving in highway scenes. Similar to this approach, Meyer *et al.* [15] used a motion filter for estimating the affine parameters of an object for position estimation. A Geometric Kalman filter was used for shape estimation wherein the shape of the object was estimated by estimating the position of the points in the convex hull of the vehicles. In our application we are interested in the object's position in the scene coordinates. Position estimation in this case can be done reliably using a simple translational model moving with constant velocity. Furthermore, a region can be represented very closely by using an oriented bounding box without requiring its convex hull. Our approach differs from that of Meyer *et al.* [15] in that we use a simple translational model for estimating the position of the centroid and the bounding box dimensions for shape. Although vehicle tracking has been generally addressed for free flowing traffic in highway scenes, this is one of the first papers that address the tracking problem for non-free flowing, cluttered scenes such as intersections.

4 Approach

An overview of our approach is depicted in Figure 3. The input to the system consists of gray scale images obtained from a stationary camera. Image segmentation is performed using a mixture

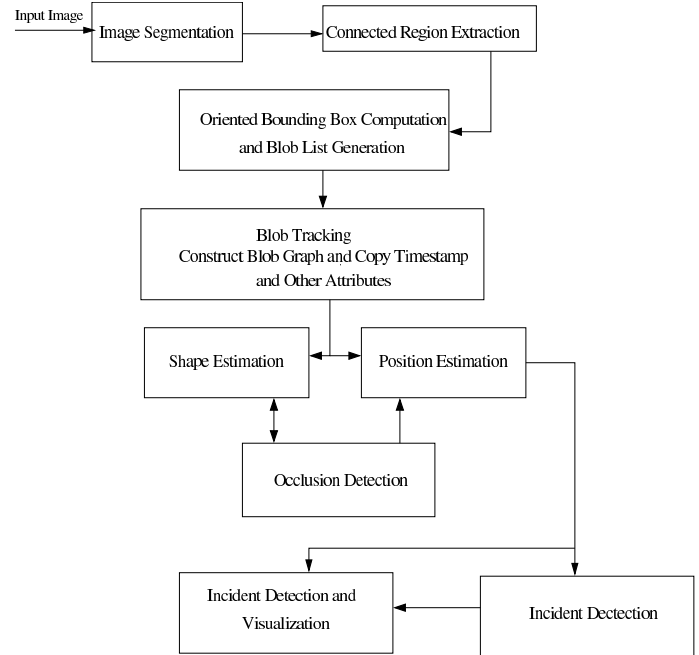


Figure 3: Tracking approach.

of Gaussian models method as in [19]. The individual regions are then extracted using a connected components extraction method. The various attributes of the blob such as centroid, area, elongation and first and second order moments are computed during connected component extraction. In order to obtain a close fit to the actual blob dimensions, appropriately rotated bounding boxes (which we call oriented bounding boxes) are used. These are computed from principal component analysis of the blobs.

Blob tracking is then performed by finding associations between the blobs in the current frame with those in the previous frame based on the proximity of the blobs. This is valid only when the entities do not move very far in between two frames. Given the frame rate and the scenes, this is a valid assumption. The blobs in the current frame inherit the timestamp, label, and other attributes such as velocity from related blob. The tracked blobs are later interpreted as MOs in the higher level. Position estimation of the MOs is done using an extended Kalman filter while their shape estimation is done using a standard discrete Kalman filter. The results from the shape estimator are used for occlusion detection.

The occlusion detection module detects occlusions on the basis of the relative increase or decrease in the size of a given blob with respect to

the estimated size of its MO. Two different thresholds are used for determining the extent of occlusion. The module also serves as a filter for the position measurements passed to the extended Kalman filter. The results from the tracking module are then passed onto the visualization module where the tracker results can be viewed graphically.

5 Moving Object Extraction

Tracking in outdoor, crowded scenes requires that the tracked entities can be segmented out reliably in spite of the complexities of the scene due to changing illumination, static and moving shadows, uninteresting background (swaying tree branches, flags) and camera motion. The method should also be fast enough so that no frames are skipped. Another requirement in this application is that stopped entities such as vehicles or pedestrians waiting for a traffic light should continue to be tracked.

5.1 Background Segmentation

An adaptive Gaussian mixture model method based on [19] is used. Each pixel in the image is associated with a mixture of Gaussian distributions (5 or 6) based on its intensities. Each distribution is characterized by a mean and variance $\{\mu, \sigma\}$ and a weight ω representative of the frequency of occurrence of the distribution. The method for segmentation is described in [19]. The Gaussian distributions are sorted in the order of most common to the least common distribution and the pixels with matching distribution having a weight above a certain threshold are classified as background while the rest are classified as the foreground. Moving entities are then extracted using a two pass connected components extraction method. In order to eliminate noise from being classified as foreground, a threshold is used so that any blob with area lower than the threshold is deleted from the foreground.

5.2 Oriented Bounding Boxes

Horizontal and vertical axis aligned boxes cannot provide tight fits to all vehicles moving in arbitrary directions. As a result, oriented bounding boxes are used to represent blobs. The oriented bounding box is computed using the two principal axes of the blob, which are in turn computed using principal component analysis. The covariance matrix used to compute this consists of the blob's first and second order moments

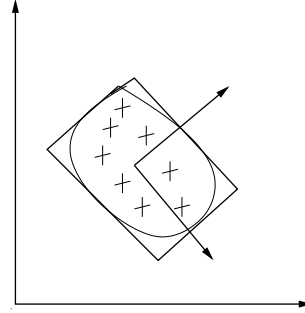


Figure 4: Principal component analysis.

$$\begin{bmatrix} M_{20} & M_{11} \\ M_{11} & M_{02} \end{bmatrix} \quad (1)$$

where, M_{ij} is the $(i, j)^{th}$ order moment of the blob. Diagonalizing M , gives

$$M = \Delta^T D \Delta \quad (2)$$

where $\Delta = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$ represents the eigenvectors and $D = \begin{bmatrix} e_1 & 0 \\ 0 & e_2 \end{bmatrix}$ represents the eigenvalues. If $e_1 > e_2$, we choose v_1 as the principal axis with elongation $2 \cdot e_1$. The angle made by the principal axis with respect to the x-axis of the image is also computed from the eigenvectors. Similarly, v_2 is chosen as the second principal axis with elongation $2 \cdot e_2$. The method of PCA is illustrated in Figure 4.

6 Tracking

Tracking is performed at two levels. The lower level consists of blob tracking which, interacts with the image processing module. The tracked blobs are then abstracted as MOs, which are tracked in the higher level.

6.1 Blob Tracking

In every frame, a relation between the blobs in the current frame is sought with those in the previous frame. The relations are represented in the form of an undirected bipartite graph which is then optimized based on [12]. The following constraints are used in the optimization:

1. A blob may not simultaneously participate in a split and merge at the same time.
2. Two blobs can be connected only if they have a bounding box overlap area at least half the size of the smaller blob.

The blob splits and merges are illustrated in Figure 5. The graph computation method is explained in detail in [20].

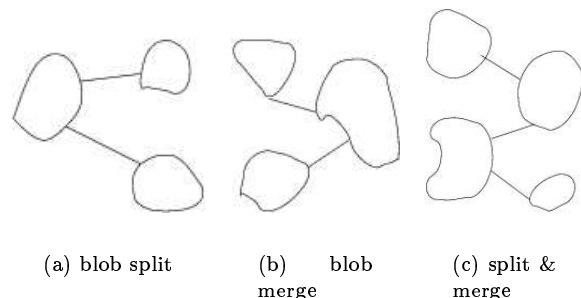


Figure 5: Blob splits and merges.

To compute the overlap between the bounding boxes of the blobs, a simple two-step method is used. In the first step, the overlap area between the axis-aligned bounding boxes formed by the corner points between the blobs is computed. This helps to eliminate totally unrelated blobs. In the next step, the intersecting points between the two bounding rectangles are computed. The points are then ordered to form a closed convex polygon whose area gives the overlap area. This is illustrated in Figure 6.

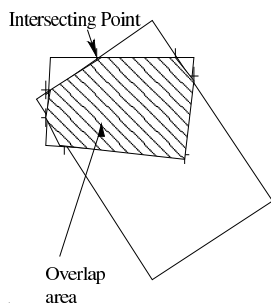


Figure 6: Computing overlap between two bounding rectangles. The intersecting points are first computed and then ordered to form a convex polygon. The shaded area represents the overlap area.

The results from this module are passed onto the high-level module where tracking consists of refining the position and shape measurements by means of Kalman filtering. An Extended Kalman filter is used for estimating the position of MO in scene coordinates while shape of the MO is estimated in image coordinates using a discrete Kalman filter.

6.2 Kalman Filter Tracking

An explanation of the Kalman filter theory can be found in [1, 21]. The position estimation filter is responsible for estimating the target position in scene coordinates. The entities are assumed to move with constant velocities and any changes in the velocity are modeled as noise in the system. Because of the nonlinearity in the mapping from the state space (world coordinates) to the measurement space (image coordinates), an extended Kalman filter is used. The state vector is represented as $X = [x, \dot{x}, y, \dot{y}]$, where x, y are the positions of the centroid in the x-y scene coordinates and \dot{x}, \dot{y} are the velocities in the x, y directions. The state trans-

sition matrix is given by $\begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$ where

δt is the time elapsed between two frames. The error covariance of the system noise is given by $Q = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} q$ where $A = \begin{bmatrix} \frac{(\delta t)^3}{2} & \frac{(\delta t)^2}{2} \\ \frac{(\delta t)^2}{2} & \delta t \end{bmatrix}$ and q is the variance in the acceleration.

The measurement error covariance R_k is given by $\begin{bmatrix} \sigma_k^2 & 0 \\ 0 & \sigma_k^2 \end{bmatrix}$. The measurement error standard deviation σ_k^2 is obtained based on the variance in the percentage difference in the measured and previously estimated size (area). The Jacobian of the measurement matrix H is used due to the nonlinearity in the mapping from image to world coordinates of the target's positions.

The filter is initialized with the scene coordinate position of the object obtained by back projecting the image measurements using the homography matrix. The homography matrix is computed from the camera calibration. The filter estimates a model of the motion of the target based on the measurements. Estimate of the model corresponds to estimating the position and the velocity of the target.

6.3 Measurement Vector

The measurement for an MO consists of the centroid of the blob (computed from the connected components extraction) and the oriented bounding box coordinates (computed using the principal component analysis). These measurements are obtained from the blob tracking module.

In order to ensure that the Kalman filters provide as accurate an estimate of the target as possible, it is necessary to provide the filters only with relevant measurements (measurements that can be distinguished as uniquely arising from the target). For example, when there is an occlusion, it is better to

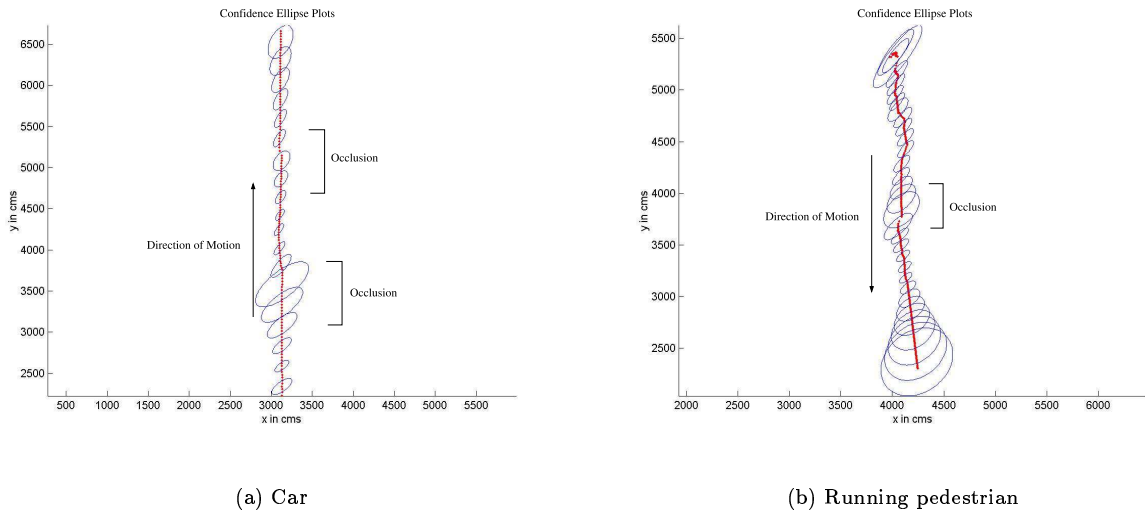


Figure 7: Confidence ellipses of the tracked targets. The position is in world coordinates. The increase in uncertainty is shown by the increase in the size of the ellipse in the regions where occlusion occurs and hence no measurement is available. The ellipse is centered around the position estimate at the current frame.

treat this case as an absence of measurement than using this for estimation as it is ambiguous as to which object this measurement must belong. The occlusion detection module acts as a filter serving to disregard erroneous measurements provided to the position and shape estimation Kalman filters. Erroneous measurements are those when a target does not have a unique measurement (there is no measurement which is associated only to this target) or when the measured blob's area differs significantly from the target's estimated bounding box area. Data association in case of a single object related to multiple blobs (multiple measurements) is done by using a combination of the most related blob (nearest neighbor) or the average centroid of all the related blobs (when all the related blobs are very close to each other). In case of multiple objects related to one or more same blobs (e.g., when two vehicles are close to each other and share one or more blob measurements), the measurements are best ignored and hence rendered as missing measurements in hope that the ambiguity will clear up after a few frames. In this case, the Kalman filter will take over with a prediction-only mode. The filter predicts based on its estimates of the velocity and the position obtained from the previous frame with increasing uncertainty as depicted in Figure 7 as long as no measurement is available. As soon as a measurement is obtained, the size of the ellipse decreases. As shown in the Figure 7, the ellipses are centered around the estimate at the current frame

and the area of the ellipse corresponds to the covariance of the estimate. Higher the area, larger the covariance in the estimate.

Generally, if the occlusions occur a few frames (at least 5 or 6) after target instantiation (so that the motion parameters have been learnt with fairly high accuracy), the filter's prediction is fairly reliable to several frames. However, one of the obvious limitations of discarding measurements is that the filter's prediction uncertainty increases and might become very large and hence unreliable when a large number of measurements has to be dropped. Such cases can arise very often in very crowded scenes. Although dropping measurements is better than using incorrect measurements, it would be better if we could somehow use at least some of the measurements by weighting the measurements probabilistically or by using cues other than just overlaps to identify the target's measurements (e.g., template of the target). Another related problem with using blob overlaps and target blob proximity for taking associated measurements is that, incorrect measurement associations might be formed (especially in cases when the target's position is highly uncertain) resulting in track divergence and track jumping.

6.4 Shape Estimation

Currently, the main motivation for doing shape estimation is for detecting occlusions. As a result, it suffices to do the estimation in the image coor-

ordinates. But later on, we would like to do this estimation in scene coordinates for providing better estimates to the incident detection module where collision detection is performed.

Three independent filters are used for shape estimation. The bounding box shape can be estimated from its length and height. However, we also need to have an estimate of where to place the box in the image. This can be known if the distance of one bounding box corner with respect to the centroid of the blob is known. Hence, the parameters estimated are the distance (x and y coordinate distance) of a corner point from the blob centroid, the length and the height (measured as x and y coordinate distances of the two other orthogonal corner points from this point). The state vector in each of the filter is represented as $X = [x, y]$ where x and y are the distances in image coordinates. The state transition matrix is $F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and the measurement error covariance for all the filters is based on the variance in the percentage difference in the estimated and the measured area of the MO.

7 Occlusion Reasoning

Occlusions can be classified as one of the two types. The first type is inter-object occlusion. This occurs when one MO moves behind the other. This kind of occlusion is the easiest to deal with as long as the two targets have been tracked distinctly. In this case, two MO's share one or more blobs. As only blobs are used for establishing associations, it can be difficult to associate a blob uniquely to one target. As a result, the best thing to do in this case is just ignore the measurements and let the individual filters of the MOs participating in the occlusion in prediction mode. The tracking in case of this occlusion is illustrated in the Figure 10 between vehicles numbered 37 and 39. This case cannot be dealt with when the two targets enter the view of the camera occluded in the first place. One case which cannot be handled is when the MOs deliberately participate in merging for example, a pedestrian getting into a car. In this case, the pedestrian MO filter completely ignores the merging of the two targets as occlusion and continues to estimate the pedestrian's position based on its previously estimated model.

The second type is object-background occlusion. This occurs when the tracked MO moves behind or emerges from behind an existing background structure. This can be further classified into two dif-

ferent types based on the effects on the blob size caused by the occlusion under the following two types

1. Object moving behind thin background structures: The scene structure in this case might be thin poles or trees and the effect of this occlusion results in blob splits as the target moves behind the structure. As long as there is only one target moving behind the structure, this can be dealt with as all the blobs are really related to the target and the measurement can be taken as a weighted average (weighted based on the percentage overlap of the blob with the target) of the blob centroid. This can get complicated when this occlusion is compounded with inter-object occlusion too. In that case, this is just treated as inter-object occlusion as described in the previous paragraph.
2. Object moving behind thick background structures: This is caused by structures such as buildings and overpasses, causing the foreground blobs that represent the MO to disappear from the scene for a certain length of time. As long as a good estimate of the MO is present, its position can be estimated and can be tracked as soon as it emerges out of the occlusion. One main problem associated with this is due to the use of the centroid of the blob as a measurement for the position of the MO. One common problem occurs in case of slow moving objects undergoing this kind of occlusion. As the MO starts moving behind the structure, it results in gradual reduction in its blob size. If this is not detected it can look like decrease in velocity of the target (as the centroid of the blob will shift towards the unoccluded portion). The effect of this is that the predicted MO (now being moved at a slower speed) will fail to catch up with the blob as it eventually re-emerges. In this case, it is important and useful to detect the onset of occlusion. This can be detected using shape estimation which is discussed in the following paragraph.

7.1 Shape Estimation Based Occlusion Reasoning

Occlusion reasoning is performed based on the discrepancy in the measured size and the estimated size. The results from the shape estimation module are used for this purpose. Accurate occlusion reasoning strongly depends on the accuracy of

the shape estimation. As long as there is no occlusion, the expected variation in the area will be the same as the measured variation. In other words, the expected area would be more or less the same as the measured area. However, when there is an occlusion, there will be a significant change in the expected area compared to the measured area. The same holds for the case when the object comes out of an occlusion. For example, when a tracked object moves behind a background region, the measured area will be much less compared to the expected area. Similarly, when a tracked object comes out of an occlusion, its measured area will be larger than the expected area

large occlusion,	$ E_a - M_a /\max(E_a, M_a) > T_{high}$
partial occlusion,	$T_{low} < E_a - M_a /\max(E_a, M_a) < T_{high}$
no occlusion,	otherwise

where, E_a is the expected area of the blob, M_a is the actual measured area, T_{low} is the low threshold and T_{high} the high threshold. The thresholds are used for determining the nature of occlusion (between partial or total). When the percentage size changes between the measured and the expected area) is above a certain high threshold, it is hypothesized to be a large occlusion and a partial occlusion if it is above a low threshold but lower than the high threshold. These thresholds were determined by trial and error based on testing using different values on different scenes. We use a low threshold of about .3 to .4 and a high threshold value of about .8 to .9. These thresholds correspond to the percentage increase in the area (measured in the image coordinates). However, using the same threshold in all places in the image itself has some limitations depending on the camera view. For instance, depending on zooming effects, these thresholds may work only when the vehicles are close to the center of the image.

The reason behind using two different thresholds is for detecting the nature of occlusion. In case of partial occlusions, the position measurement is used for filter update but the shape measurements are ignored. In case of total occlusions, both the position and shape measurements are ignored. The reason for detecting the nature of thresholds is twofold. By taking measurements for position in the event of partial occlusion, we can provide more measurements (though less reliable) to the filter. Secondly as the shape estimates are not updated, the onset

of the total occlusion can be identified earlier and hence the problem discussed in Section 7 on object moving behind thick background structures can be addressed.

8 Incident Detection Visualization Module

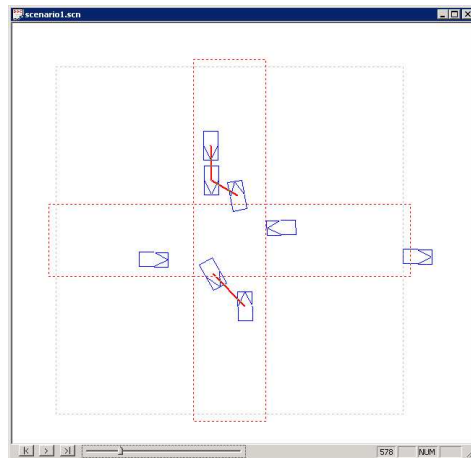


Figure 8: Incident detection interface.

The results from the vision module are passed to the incident detection module. The incident detection module is responsible for detecting situations such as possible collisions between vehicles. For this, it uses the position, velocity and shape (length and width), of the vehicles in scene coordinates obtained from the vision module. Currently, our focus is only on collision detection at the current frame. Collisions could be detected by checking if the distance between any two vehicle bounding boxes is less than a threshold and the results can be presented visually in the module as shown in Figure 8. The module acts as a graphical user interface providing real-time visualization of the data with an easy to use VCR-like interface. The module can also be used for presenting the results of the tracking which is hence a very useful tool also for debugging purposes. Figure 8 shows a snapshot of the interface. The vehicles are shown by rectangular boxes and the vehicles in very close proximity are indicated by the line segments. Camera calibration is used for recovering the scene coordinates of the traffic objects.

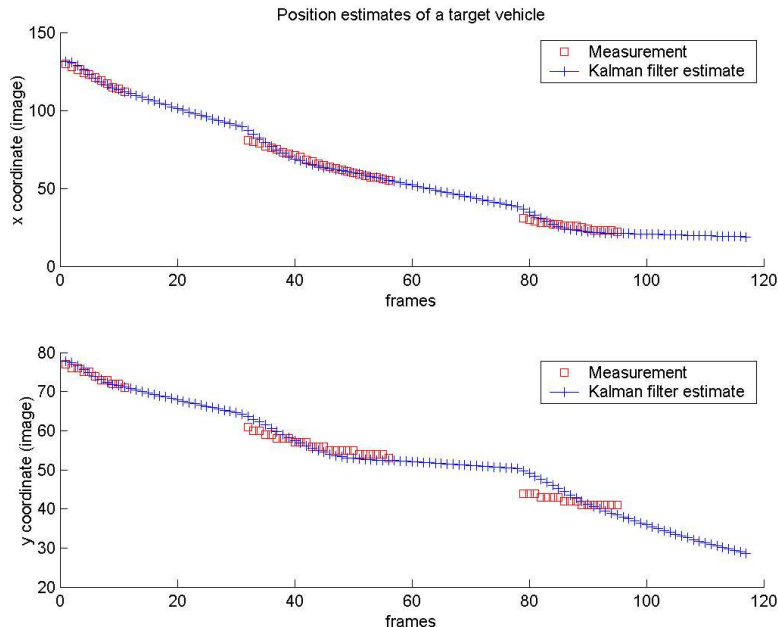


Figure 9: Position estimation.

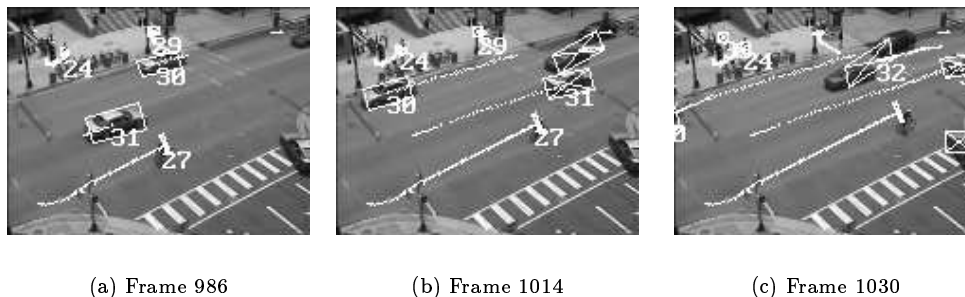


Figure 10: Tracking sequence.

9 Camera Calibration

Camera parameters are hard to obtain after the camera has already been installed in the scene. Hence, the parameters are obtained by estimation using the features in the scene. This is done by identifying certain landmarks in the scene that are visible in the image along with their distances in the real world. A camera calibration tool described in [13] is used for calibration. The input to the tool consists of landmarks and their distances in the scene. The tool computes the camera parameters using a non-linear least squared method. Once the scene is calibrated, any point in the image can be transformed to the scene coordinates (the corresponding point on the ground plane of the scene).

10 Results

Our tracking system has been tested on a variety of weather conditions such as sunny, cloudy, snow etc. The results of a track sequence are shown in Figure 10. The tracked sequence shown consists of a total of 44 frames with the results shown for frame number 986, frame number 1014, and frame number 1030. The lines behind the vehicles and pedestrians show the trajectories of the vehicles and pedestrians. The numbers on the pedestrians and vehicles are the track labels assigned to every tracked MO. The tracker handles the occlusions between the cars very well as can be seen from the sequence. Figure 13 shows occlusion handling between two vehicles. Tracking in a winter sequence

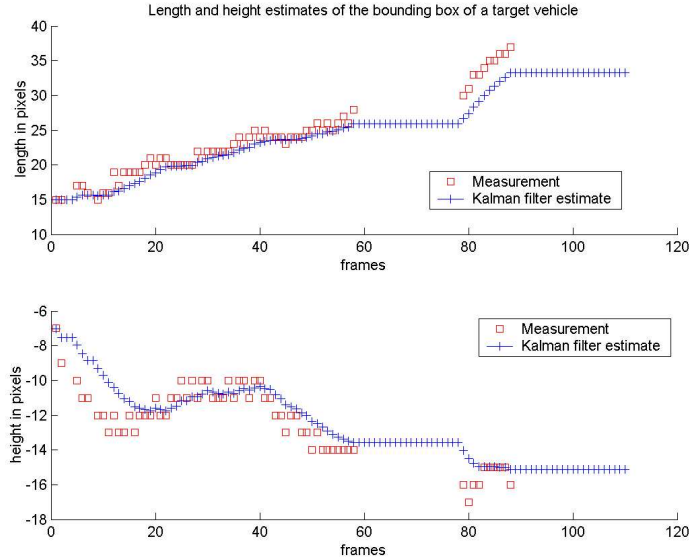


Figure 11: Shape estimation for occluded sequence. Occlusions are indicated by missing measurements.

is shown in Figure 14 while Figure 15 shows tracking in snow and shadow conditions.

The results of the Kalman filter position estimates for a vehicle are shown in Figure 9. The position estimates of the Kalman filter are presented against the actual measurements. These results are presented in the image coordinates. The results are presented for a vehicle that was occluded multiple times as shown in Figure 9 (this is indicated by the absence of measurements). The sequence also illustrates occlusion handling between two vehicles. The results for the shape estimation for a vehicle undergoing occlusions and a turning vehicle are shown in Figure 11 and Figure 12. The results are presented for the estimated length and height against the actual measurements. Turn sequence shows an increase in the length and height of the vehicle as its pose with respect to the camera changes. The length and height represent the coordinate difference between the estimated bounding box corner point to its adjacent corner points on the bounding rectangle. This is the reason why some of the length and height measurements in the Figures 11 and 12 have negative values.

11 Discussion

We now provide a brief discussion and insights to future work. The two level Kalman filter based tracking is capable of providing robust tracking under most scenarios. Combining a shape estimation

filter along with a position estimation filter helps not only to identify occlusions but is also useful in propagating only the reliable measurements to the high-level tracking module. Good data association is essential for the robust performance of the Kalman filter. The system can be applied reliably in most traffic scenes ranging from moderately crowded to even heavily crowded (as long as some reliable measurements can be provided to the filter through the sequence).

The Gaussian mixture model approach works fairly well for most traffic scenes and can handle illumination changes fairly quickly. However, this method cannot be used for tracking stopped vehicles. The reason being that the stopped vehicles are modeled into the background. But for our purpose, we cannot assume vehicles or pedestrians waiting for a traffic signal as background as they stop only for short periods of time. Although we can detect static cast shadows and model them into the background, we cannot detect moving cast shadows in the image. Moving cast shadows distort the shape of the vehicle and affect the quality of the tracker. These problems are addressed to some extent by the two level tracking approach. For example, the Kalman filter can continue to track a vehicle for some time even after it gets modeled into the background based on its previous estimates. Similarly, if the region where the moving shadows occur is a small region, the shape estimator can ignore this as a bad measurement.

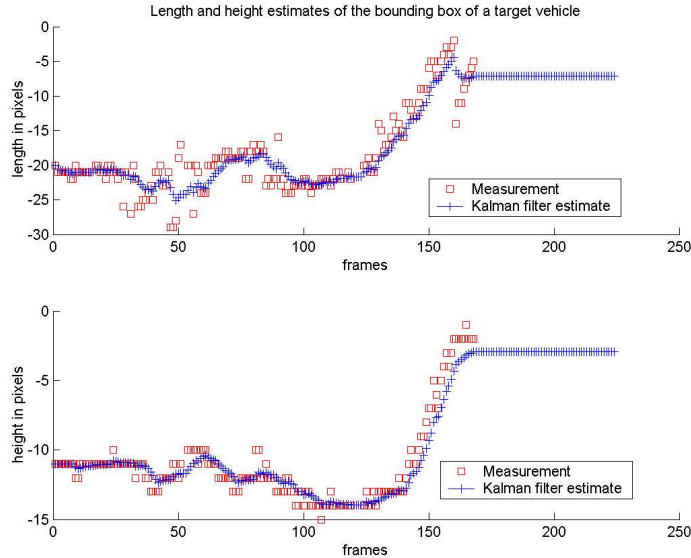


Figure 12: Shape estimation for a turn sequence.

Although the tracker performs very well in moderately crowded scenes with less background clutter, the performance deteriorates in very cluttered scenes owing to the reason that an increasing number of measurements are ignored with increased density of the crowd resulting in tracking divergence. One problem with the current shape estimation method is that sometimes it is difficult to distinguish between an occlusion and a pose change based on relative size increase or decrease. Treating both the cases with the same hypothesis (size change) is not sufficient and results in tracker inaccuracies. This in itself suggests several improvements to the tracker. Instead of using a single hypothesis from the Kalman filter, we should be able to formulate multiple hypotheses for tracking. The need for multiple hypothesis-based tracking arises from the increased ambiguity in the data in the presence of clutter and the ambiguity in distinguishing different motions (turn vs. vehicle passing under a background artifact). A probabilistic data association filter has been used by [2] for tracking targets in clutter. Similarly, a multiple hypothesis approach which maintains a bank of Kalman filters has been used by Cham *et al.* [3] for tracking human figures. Another direction for improvement would involve using more cues from the image itself. Although stopped vehicles can be tracked for some more time by the Kalman filter, they cannot be tracked reliably over long periods of time without actual measurements. This requires changes

to the existing segmentation method or improvements in the segmentation method by additional measurements through a template of the region (constructed from previous tracking instants) for example.

12 Conclusions

A multi-level tracking approach for tracking the entities in intersection scenes is presented. The two level tracking approach combines the low-level image processing with high-level Kalman filter based tracking. Combinations of position and shape estimation filters that interact with each other indirectly are used for tracking. The shape estimation filter serves the purpose of occlusion detection and helps provide reliable measurements to the position estimation filter. An incident detection visualization module has been developed which provides an easy to use graphical interface and on-line visualization of the results.

13 Acknowledgement

This work has been supported by the ITS Institute at the University of Minnesota, the Minnesota Department of Transportation, and the National Science Foundation through grants # CMS-0127893 and # IIS-0219863. The authors would also like to thank the anonymous reviewers for their helpful and constructive comments.



Figure 13: Tracking sequence showing occlusion handling.

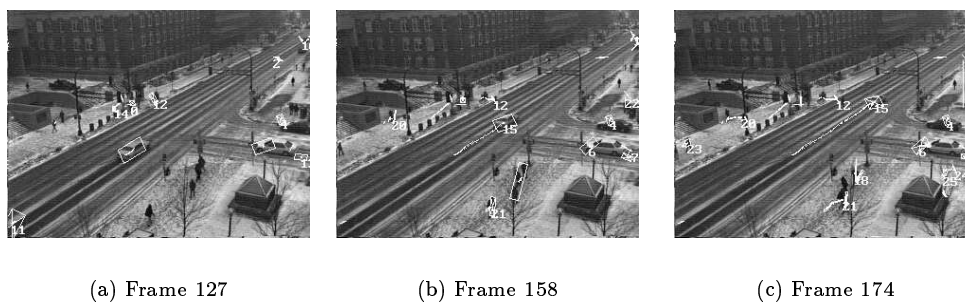


Figure 14: Tracking sequence in winter.

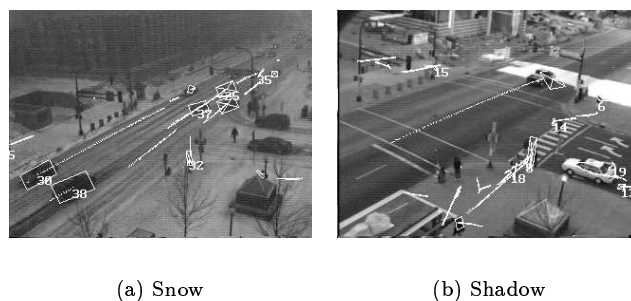


Figure 15: Tracking results in snow and shadow.

References

- [1] Y. Bar-Shalom, X. Rongli, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. John-Wiley and Sons, 2001.
- [2] K. Birmiwal and Y. Bar-Shalom. On tracking a maneuvering target in clutter. *IEEE Transactions on Aerospace and Electronic systems*, AES-20(5):635–644, September 1984.
- [3] T. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR'99)*, pages 239–245, June 1999.
- [4] 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*, 6(4):271–288, 1998.
- [5] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Statistic and knowledge-based moving object detection in traffic scenes. In *Proc. of IEEE ITSC Intl. Conf. of Intelligent Transportation Systems*, 2000.
- [6] R. Cucchiara, P. Mello, and M. Piccardi. Image analysis and rule-based reasoning for a traffic monitoring system. *IEEE Transac-*

- tions on Intelligent Transportation Systems*, 1(2):119–130, 2000.
- [7] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, July 2002.
- [8] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.
- [9] B. Heisele, U. Kressel, and W. Ritter. Tracking non-rigid, moving objects based on color cluster flow. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 257–260, 1997.
- [10] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, September 2001.
- [11] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proc. of the European Conf. on Computer Vision (1)*, pages 189–196, 1994.
- [12] O. Masoud. *Tracking and analysis of articulated motion with application to human motion*. PhD thesis, University of Minnesota, 2000.
- [13] O. Masoud, S. Rogers, and N. P. Papanikolopoulos. Monitoring weaving sections. Technical Report CTS 01-06, October 2001.
- [14] S. J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *Proc. 4th Intl. Conf. on Automatic Face and Gesture Recognition*, pages 348–353, 2000.
- [15] F. G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *Computer Vision, Graphics and Image Processing: Image Processing*, 60(2):119–140, September 1994.
- [16] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using Kalman filtering. In *Proc. International Conference on Recent Advances in Mechatronics*, pages 193–199, 1995.
- [17] R. Rosales and S. Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. Technical Report 1998-007, 1998.
- [18] P. L. Rosin and T. J. Ellis. Detecting and classifying intruders in image sequences. In *Proc. of the 2nd British Machine Vision Conf., Glasgow*, pages 293–300, 1991.
- [19] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. Computer Vision and Pattern Recognition Conf. (CVPR '99)*, June 1999.
- [20] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos. Vision-based monitoring of intersections. In *Proc. of the IEEE Conf. on Intelligent Transportation Systems*, 2002.
- [21] G. Welch and G. Bishop. An introduction to the Kalman filter (tutorial). SIGGRAPH 2001, 2001.
- [22] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfnder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.