Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Vision and Image Understanding 103 (2006) 121–138
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Robust target detection and tracking through integration of motion, color, and geometry

Harini Veeraraghavan, Paul Schrater, Nikos Papanikolopoulos *

Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55445, USA

Received 4 June 2005; accepted 24 April 2006
Available online 30 June 2006

Abstract

Vision-based tracking is a basic elementary task in many computer vision-based applications such as video surveillance and monitoring, sensing and navigation in robotics, video compression, video annotation, and many more. However, reliable recovery of targets and their trajectories in an uncontrolled environment is affected by a wide range of conditions exhibited by the environment such as sudden illumination changes and clutter. This work addresses the problem of (i) combining information from a set of cues in order to obtain reasonably accurate estimates of multiple targets in uncontrolled environments and (ii) a collection of data association methods for cues containing less information for robust tracking through persistent clutter. Specifically, we introduce a novel geometric template constrained data association method for robust tracking of point features, while using the Joint Probabilistic Data Association (JPDA) method for blob cue measurements. Extensive experimental validation of the tracking and the data association framework is presented in the work for several real-world outdoor traffic intersection image sequences.

Keywords: Multiple cue combination; Measurement error estimation; Expectation maximization; Data association

1. Introduction

Vision-based tracking is an important elementary task in several computer vision-based applications including, video surveillance and monitoring, sensing and navigation in robotics, key-frame detection, video summarization, and many more. As a sensing modality, the low cost and large information contained in vision are often offsetted by the large ambiguities present in the data, making localization and tracking a very challenging task. This work addresses the problem of multiple target localization and tracking in general outdoor image sequences by using a family of algorithms that make use of cue combination from multiple vision-based cues and appropriate data association strategies for the measurements.

Tracking has been addressed as the problem of obtaining target localization through multiple cues [1–4], robust data association using multiple hypothesis [5,6], probabilistic data association [7], as well as, non-parametric or particle filtering approaches as in [8,9]. Multiple cue based methods assume that at least one of the cues provides accurate localization of the target in each frame and thus ignore the problem of ambiguous target measurement association in the presence of clutter. On the other hand, methods that explicitly address the problem of data association often assume that some measurement of the target is always available. In contrast to the aforementioned methods, non-parametric estimators use a flexible model by sampling from the target density, thereby, addressing the problem of localization and tracking in clutter and missing measurements at the cost of immense computational overhead. Hence, though robust, these methods are limited in their scalability to the number of targets. Additionally, owing to the flexible model representation, there is a danger of collapsing distinct targets to a single one.

* Corresponding author. Fax: +1 612 625 0572.
E-mail addresses: harini@cs.umn.edu (H. Veeraraghavan), schrater@cs.umn.edu (P. Schrater), npapas@cs.umn.edu (N. Papanikolopoulos).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.04.003

This work addresses the limitation of the aforementioned approaches to obtain a reasonably fast, yet robust tracking for a large number of targets by combining cue combination and data association in a single systematic framework. The main contributions of this work are: (i) an incremental, weighted cue combination method for combining heterogeneous measurements; position and velocity from multiple vision cues, such as motion segmented blobs, color, point features, as well as image templates, and (ii) appropriate data association strategies for the individual cues. The data association is integrated with the cue combination to yield a single framework for target localization and tracking. In addition, extensive experimental validation of tracking in real outdoor image sequences is presented.

While the use of multiple cues is similar to other approaches [1–3], this work differs from the above mentioned work in the following respects:

(1) Cues are combined incrementally so that it is not essential for all the cues to be available at a given step. Also, not all cues provide measurement of the same variable. For example, while blob and color-based localization are used to obtain the position of the targets in the image, point features provide the velocity of the targets in image. In addition, we make use of information such as occlusions inferred using one cue to resolve ambiguities in another cue along with a systematic evaluation of the measurement errors.
(2) Appropriate data association for the individual features is employed along with cue combination to obtain a unique and robust tracking framework.

1.1. Tracking approach

Our basic approach to tracking is outlined in Fig. 1. Targets are automatically initialized using the results of blob tracking. Thus, a blob provides not only the localization of a target in each frame, but also delineates the targets of interest. In order to eliminate false targets, the following two heuristics are used:

(1) Only those blobs moving with at least a minimum velocity and having been successfully tracked for a few consecutive frames are initialized as targets. The minimum velocity assumption and the successful tracking assumption prevents erroneous foreground regions resulting from segmentation errors from being initialized as targets.
(2) Only those targets that have a distinct color distribution compared to the surrounding background are initialized. This is based on the assumption that all true targets differ from the background in color.

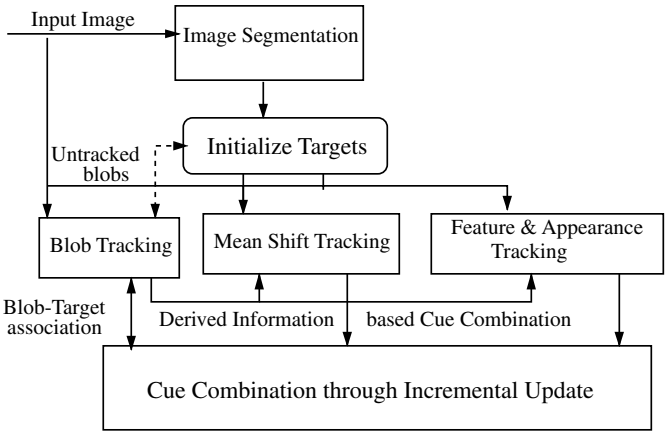


Fig. 1. Tracking approach. Targets are initialized using blobs not associated to any target. In each step, the measurements from blobs, color, and features are combined incrementally. Occlusion information derived from the blob tracker is used to constrain the results of color and feature-based target localization.

As shown in Fig. 1, three different cues, namely, blobs, color, and point features are used to represent the targets. Additionally, a template of the target blob region is used in conjunction with the other cues by a voting-based cue combination method as in [1] for comparison with our tracking approach. The target motion is modeled using a simple first-order or constant velocity motion model. By using one or all the three cues (as they are available), each target's state (position, velocity) in the scene is estimated using an extended Kalman filter. Measurements from blobs and point features are refined using two different data association methods as described in Sections 4.1 and 4.2.

1.2. Paper outline

This paper is arranged as follows: after introducing the problem in Section 1, the individual tracking modalities are discussed in Section 2, and Section 3 discusses the theory of cue combination. Details of the data association methods for blob and feature tracking are in Section 4. Section 5 presents the experimental results and Section 6 presents the discussion of results. Finally, Section 7 concludes the paper.

2. Tracking method

Targets are detected using blobs or foreground regions obtained through an adaptive background segmentation approach [10]. Once initialized, the target's color and its geometric appearance are constructed from the image region enclosed by its associated blob.[1] Both the color and the geometric appearance model are adaptive and are

[1] In this work, geometric appearance refers to the local configuration of features.

replaced when the corresponding model can no longer provide meaningful localization.

The target state consists of $[x, \dot{x}, y, \dot{y}]$, namely, the position (x, y), and velocity $(\dot{x}, \dot{y})$ in the scene coordinates. An extended Kalman filter model estimates the target state through an incremental incorporation of each cue. This corresponds to a sequential update of the filter state [11], where the individual measurements are incorporated in the order of arrival.

2.1. Blob tracking

Blobs serve the dual purpose of localizing and initializing the targets of interest. The two main issues that arise when using blob measurements include, (i) ambiguity in the interpreted blob-target associations, and (ii) the difficulty in quantifying the error in each measurement.

2.2. Data association ambiguity

Data association ambiguity typically arises from:

- Errors in the background segmentation resulting from sudden illumination changes, camera motion, target stalled for a long period of time, random image noise, etc., and
- Occlusions, background as well as foreground.

Commonly used methods address the data association ambiguity by applying one of the methods described in [6,7,12]. Given the real-time tracking constraints, we use the joint probabilistic data association method for computing the target-blob associations. The method is based on computing the probability of a blob measurement z arising from a target, given its current position estimate $\hat{x}$ and the current state covariance Σ at time t obtained through the Kalman filter estimation:

$$p(z|\hat{x}) = \frac{1}{\sqrt{2\pi}\Sigma^{d/2}} \mathrm{e}^{(-0.5*[z-\hat{x}]'\Sigma^{-1}[z-\hat{x}])} \tag{1}$$

where d is the dimension of the target position (which in our case is 2).

2.3. Measurement errors

The error in blob measurements arises due to errors in blob segmentation. Directly quantifying these errors solely based on the changes in the blob area is not accurate. Hence, we estimate these errors empirically using the standard Expectation Maximization approach. The input to the EM algorithm consists of manually selected target trajectories containing little or no occlusions. As such, the measurements obtained from occluded blobs are assigned an arbitrarily large error so that such measurements are discounted in comparison to more reliable measurements.

2.4. Color: mean shift tracking

Target color contains more information about the target, thereby making localization easier. Ambiguities arise when the occluding targets share a similar target distribution or when the target is passing under regions with significant illumination changes. The mean shift algorithm proposed by Comaniciu et al. [13] is used for localizing the targets using color.

The color model is automatically initialized from the blob region associated with the target. In order that only the salient parts of the target are captured in the model, portions of the blob distinct from the background are weighed more heavily compared to those parts similar to the background. The target model is replaced by a new one when the old model can no longer provide meaningful localization.

The basis of the tracking method consists of matching a stored model of the target T_m with a candidate model $T_c(y)$ computed around a region y. The match is computed by using the Bhattacharya coefficient as,

$$d(y) = \sqrt{1 - f(T_c(y), T_m)} \tag{2}$$

where the function $f(.)$ is chosen as,

$$f(T_c(y), T_m) = \sum_{i=1}^{m} \sqrt{T_c(y)^i T_m^i} \tag{3}$$

and m is the number of histogram bins chosen to represent the color distribution. The target and the candidate model are computed as,

$$T_m = C \sum_{i=1}^{N} k(\|u_i\|^2)\delta[b(u_i) - l] \tag{4}$$

$$T_c = C \sum_{i=1}^{n_h} k\left(\left\|\frac{y - u_i}{h}\right\|^2\right)\delta[b(u_i) - l] \tag{5}$$

where the set of pixels in a region are represented as $u_{1,\ldots,N}$ or $u_{1,\ldots,n_h}$ for a target model and a candidate location y, $b(.)$ is a function that maps a pixel value at a given location to the corresponding bin of the color histogram, the term δ is the Kronecker delta function which computes the probability of the pixel value belonging to the target model $l = 1, \ldots, m$, and h is the scale of the target represented in the x and y coordinates. To account for the changing scale h of the target, as it moves closer to or farther away from the camera, the scale is recomputed in each frame as long as the target is not detected to be occluded.

The error in color measurement arises from (i) the error in position localization, and (ii) error in the region size used for computing the match. This results from inaccuracies in the choice of the scale. Hence, in this work, the measurement error is computed as a function of position and scale. For this, we use the Sum-of-Squared Differences (SSD) error metric as described in [14,13] where a scaled Gaussian distribution of the SSD matches around a target location is fit and the variance of the Gaussian corresponds to the

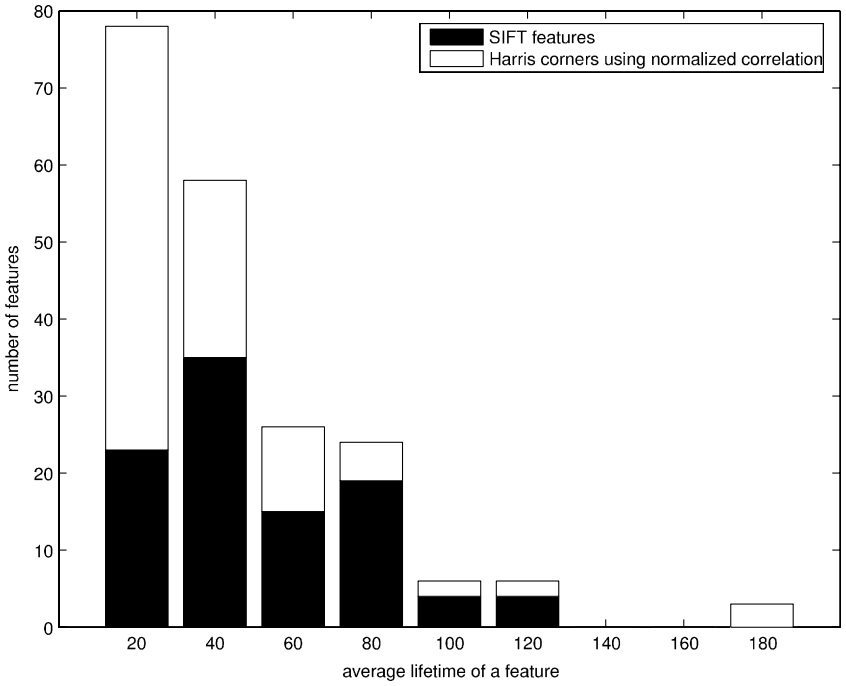


Fig. 2. Comparison of average lifetime of SIFT features matched using normalized correlation and SIFT descriptor matching. As can be seen, both SIFT descriptor-based matching and normalized correlation-based matching lose most of the features after frame number 40 (around 70–80%) thereby, requiring re-initialization of new features. The x-axis corresponds to the number of frames, while the y-axis represents the average number of features normalized to the range 0–100.

error in the localization. In our formulation, this Gaussian is fit by computing the SSD match across both position and scale as,

$$e = \sum_h \sum_x \sum_y [ssd_{h,x,y} - ssd_\mu][ssd_{h,x,y} - ssd_\mu]' \tag{6}$$

where e is the measurement error, ssd_μ is the Sum-Squared-Difference error at the mean target location μ, h are the discrete scales, x and y are the discrete locations along the x and y coordinates. Since the error does not change significantly between every individual pixel, a discrete step size δw is chosen for the x and y positions.

2.5. Feature and geometric appearance tracking

The collection of features on a target provides the measurement of its image velocity. As the target moves in the scene, new features are initialized to replace those lost due to target pose changes and mistracking. An adaptive geometric template enforcing the rigidity constraint between pairs of features resolves the data association ambiguities in localizing point features. Details of the data association method are discussed in a later Section 4.2. Scale Invariant Feature Transform (SIFT) [15] is used for detecting the salient features on the target. However, given that the targets are non-static and exhibit large rotations and translations, any feature matching method is bound to fail after a few frames since the features are completely out of the camera view. Fig. 2 shows a comparison of the average lifetime of SIFT features matched using normalized correlation and SIFT descriptor matching. As can be seen, a large fraction of features, about 83% of the features are lost by frame 60 by both the matching methods, requiring re-initialization of new features. Hence, in favor of the computational speed, normalized correlation of the keypoint feature templates are used for subsequent matching.[2]

Each feature is tracked in the image coordinates using a first-order discrete Kalman filter as:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} \delta x_t \\ \delta y_t \end{bmatrix} + w \tag{7}$$

where $(\delta x_t, \delta y_t)$ corresponds to the displacement of the point at time t and w is some random Gaussian noise in the model. The state of the filter consists of $[x, \dot{x}, y, \dot{y}]$ where x, y correspond to the position of the feature in the image while $\dot{x}$, and $\dot{y}$ correspond to its velocity in the x and y image coordinates.

The target's velocity is obtained from the features as a weighted average of the individual feature's estimated image velocity. Individual feature weights α_i are obtained from the estimated velocity covariance of each feature σ_v^i as,

[2] Computing SIFT descriptor matching requires the computation of several convolution operations in different octaves, followed by non-maxima suppression and nearest neighbor matching, all of which are computationally intensive, thereby, resulting in very slow frame rates of the tracker.

$$v = \sum_{i=1}^{K} \alpha_i \dot{x}_i \tag{8}$$

$$\alpha_i = \frac{\frac{1}{\sigma_v^i}}{\sum_{j=1}^{K} \frac{1}{\sigma_v^j}}.$$

The individual feature measurement errors are computed by fitting a scaled Gaussian distribution over the SSD matches around the mean feature position as in [14].

3. Cue combination

Given a set of measurements arising from different cues, democratic integration [16] is a straightforward way of combining measurements from multiple cues. However, weighting all the cues equally in general environments is not valid since each cue varies in its performance with the scene conditions. Besides it is not appropriate to weight heterogeneous measurements equally. Previous methods such as [1,4,17,18] combine multiple homogeneous cues by weighting the cue measurements based on their performance, thereby developing a more realistic cue combination method for general scenes. However, these methods often assume that it is straightforward to compute the cue performance directly from the image measurements and ignore the effects of occlusions on obtaining reliable data association. This work addresses the problem of cue combination in cluttered environments by (i) applying a variable error-based cue combination, where the error in each cue measurement is computed in each frame and (ii) including data association with cue combination. The following Section 3.1 and Section 4 discuss error estimation and the data association in detail.

3.1. Error estimation

For each cue, the measurement error corresponds to the extent of ambiguity present in its measurement. While this error can be computed directly for some cues such as color and features (based on a computable distance measure such as the similarity of the color histograms, the local image template with the stored target model, or some feature descriptors), it is not so trivial in the case of cues such as blobs which lack target specific information. The error in the nominal blob measurements are thus computed empirically using an Expectation Maximization approach as discussed in the following paragraph.

3.1.1. Blob error estimation

The problem of estimating the measurement error covariance R is cast as a parameter estimation problem for a linear dynamic system:

$$X_t = A_t X_{t-1} + Q_t \tag{9}$$

$$Y_t = H_t X_t + R_t \tag{10}$$

where A_t, H_t correspond to the state and observation transition, while Q_t, R_t correspond to the state and measurement error covariances. The problem consists of estimating the values of the system and measurement error covariances given a set of trajectory sequences. Since, we are interested only in the unoccluded blob measurement errors, only sequences free of occlusions are used for estimation. The basic problem of estimation consists of computing the joint density of the state variable X and the observation Y in the E-step of the EM algorithm. For details of the EM-based state estimation interested readers can refer to [19,20].

$$\begin{aligned}\log P(X,Y) = &-\sum_{t=1}^{K}\left(\frac{1}{2}[Y_t - HX_t]^{\mathrm{T}} R^{-1}[Y_t - HX_t]\right) - \frac{K}{2}\log|R| \\ &-\sum_{t=2}^{K}\left(\frac{1}{2}[X_t - AX_{t-1}]^{\mathrm{T}} Q^{-1}[X_t - AX_{t-1}]\right) \\ &-\frac{K-1}{2}\log|Q| - \frac{1}{2}[X_1 - X_0]^{\mathrm{T}} P_0^{-1}[X_1 - X_0] \\ &-\frac{1}{2}\log|P_0| - \frac{K(D1+D2)}{2}\log 2\pi \end{aligned} \tag{11}$$

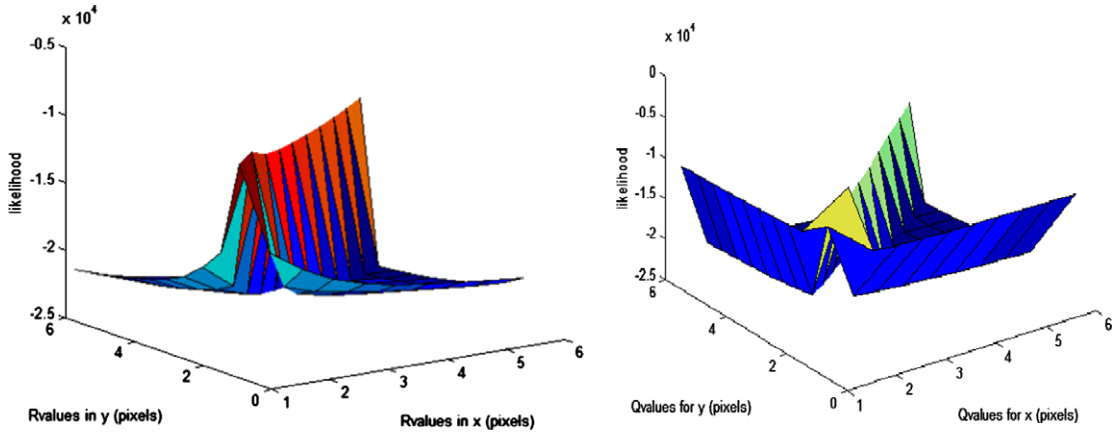
where X_0 and P_0 correspond to the initial state estimate and the state covariance. $D1$ and $D2$ correspond to the dimension of the state and observation vectors, respectively. In short, the E-step of the EM algorithm consists of computing the likelihood of $P(X, Y|Y)$ while the parameters of the state space model are estimated in the M-step. Skipping details of derivation, the measurement or observation covariance R_n and the system covariance, Q_n for N sequences of length $K_1, K_2, \ldots, K_N$ are given by the following equation:

$$\begin{aligned} R_n &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K_i}\sum_{t=1}^{K_i}[Y_t - HX_t]^{\mathrm{T}}[Y_t - HX_t] + HP_tH^{\mathrm{T}} \\ Q_n &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K_{i-1}}\sum_{t=2}^{K_i}(P_t - 2AP_{t-1} + AP_{t-1}A^{\mathrm{T}}). \end{aligned} \tag{12}$$

Estimation consists of an iterative computation of the joint likelihood, based on the smoothed state estimates in the E-step, Eq. (11) and the updated parameters, R and Q computed in the M-step, Eq. (12). Fig. 3 shows the log likelihood surface plot obtained at the end of the EM procedure for the measurement error and system error covariance. The likelihood surface is depicted for the errors in x and y positions for both error covariances, (R, Q) in image coordinates. Since the estimation of the error covariance is done in the image coordinates, the system error covariance is transformed to the corresponding error in the scene coordinates for different target positions.

3.2. Derived information-based cue combination

Occlusions inferred from a cue can be used for constraining the measurements obtained from other cues. For instance, no explicit data association is applied for measurements obtained from color-based localization since ambiguities arise only when targets with similar color distribution participate in an occlusion. In such a case, infor-



(a) Measurement error covariance estimate. **(b)**System error covariance estimate for position.

Fig. 3. Log likelihood estimates of measurement error. The surface plots correspond to the log likelihood of the errors in (x, y) positions for the measurement and system error expressed in the image coordinates, obtained at the end of EM procedure. x- and y-axis correspond to the error in x and y coordinates (in pixels), while the z-axis corresponds to the log likelihood obtained in the E step.

mation about an occlusion is useful in interpreting the measurements obtained for the targets from the color cue. Occluding targets are detected from the blob cues as a result of data association. Targets detected as sharing one or more measurement with other targets are flagged as occluded for the mean shift tracker based localization. Similarly, the measurements arising for features inside the regions detected to be occluded are associated with a large measurement error thereby, eliminating the effects of spurious matches.

3.2.1. Color measurements and scale adaptation

The target color match is obtained by matching the color histogram computed around a region encompassing each candidate location with the stored target model. The size of the region or the scale used for matching should be varied as the target moves towards or away from the camera. Hence, the scale is recomputed in each frame except during occlusions or for periods when no measurement is obtained for the target from color. Occlusions are detected using the blob tracking as explained in the previous paragraph. The new scale h_{new} update can be summarized as,

$$h_{\text{new}} = \begin{cases} h_{\text{curr}} \times (1-\alpha) + h_{\text{old}} \times \alpha, & \text{if no occlusion} \\ h_{\text{old}}, & \text{otherwise} \end{cases} \tag{13}$$

where h_{curr} is the scale computed at the current time step, h_{old} is the scale computed from the past frames, and α is the chosen weight. Note that under some circumstances, where only two targets participate in an occlusion, this can be used to infer the relative depth of the targets.

4. Data association

The problem of data association is well addressed in the tracking literature [11,6]. This problem is particularly severe in the case of cluttered scenes especially when using cues contain very little information particular to the target. Two such cues used in this work, include, blobs and features.

4.1. Blob-target data association

In this research, we make use of the Joint Probabilistic Data Association (JPDA) filter [7]. The solution to the data association problem is formulated as computing sets of valid target-measurement pairs with the constraint that no two targets share a measurement, and each measurement has only one unique target in each joint target-measurement association event. The probability of a joint event, $\Omega(t)$ at time t, given the set of measurements $Z_1, \ldots, Z_t$ from $1, \ldots, t$ for a given set of target states $X_1, \ldots, X_N$ is given as,

$$P(\Omega(t)|Z_t) = cp[Z_t|\Omega(t), X_t]P(\Omega(t)) \tag{14}$$

where c is a normalization constant. Note that, this method assumes that the measurements arise from the set of known targets alone. This is not true in our case, since measurements might arise from uninitialized targets due to delayed track initiation which can result in mistracking in some cases. This is currently addressed through gating the measurements using a Mahalanobis distance gating before computing the target measurement association probabilities using the JPDA method.

4.2. Adaptive geometric template constrained feature data association

Although image features provide robust and a computationally simple way of determining target motion, it is difficult to accurately localize point features when they are located close to each other or during occlusions. Hence, we employ a geometric template that constrains the motion of a feature by its local spatial configuration with respect to

other features. Essentially, a common motion constraint is enforced on the features to obtain robust data association.

4.3. Geometric template

The geometric template consists of a spanning tree connecting the set of features on the target. While a fully connected graph or clique-based representation of features will allow us to enforce a stronger constraint on the motion of features with respect to each other, currently, we chose a spanning tree representation for ease of implementation and computational speed in the estimation. The template as illustrated in Fig. 4, is also adaptive to newly added and removed features from the target. Furthermore, since only the features that are contained in the template for a target are used for providing the target velocity, this method also provides a novel method for eliminating features not belonging to the target. The geometric template is briefly discussed in Section 4.3 while the data association method is discussed in Section 4.4.

Furthermore, since no prior model need to be specified for a target, the template can be generalized for modeling any rigid object.

4.4. Geometric template-based data association

Under the assumption that the tracked targets are rigid, the deformations in the geometric template will be restricted to slight changes in the spatial relations between the features on the target. Hence, the problem of data association consists of finding the set of measurements in a given frame that minimizes the extent of deformation in the template. This is the same as maximizing the joint likelihood of all the measurements given the appearance. This can be expressed as,

$$\Lambda(a) = p(Z|a) = p(z_1, z_2, \ldots, z_n|a) \tag{15}$$

where $\Lambda(a)$ is the likelihood of the appearance, and $Z = \{z_1, z_2, \ldots, z_n\}$ is the set of measurements for N features on the template, and a is the appearance. Since each feature measurement is obtained independent of one another, Eq. (15) can be written as,

Fig. 4. A geometric template imposed on targets. The template consists of a spanning tree connecting the individual features on the target.

$$\Lambda(a) = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} p(z_i|a)p(z_j|a) \tag{16}$$

$$a^{\mathrm{ML}} = \operatorname{argmax} \Lambda(a). \tag{17}$$

Thus, the maximum likelihood solution is obtained by the maximum of the appearance likelihood for all the feature measurements. The likelihood of a feature measurement is computed based on the appearance and the estimate of the feature motion as,

$$\begin{aligned} p(z_i, \hat{x}_i|a) &= p(z_i|\hat{x}_i)p(\hat{x}_i|a) \\ &= N(z_i; \hat{x}_i, \Sigma_i)N(\hat{x}_i; a, \Sigma_a) \end{aligned} \tag{18}$$

where z_i is the measurement for a given feature i, $\hat{x}_i$ and Σ_i are the estimated position and covariance associated with the feature i, respectively. Σ_a is the estimated covariance in the appearance a.

One nice property of this relation is that in the worst case, where we have a very large uncertainty in the geometric apparance, the likelihood reduces to computing the Mahalanobis distances of the individual feature measurements. The algorithm for data association is described in Table 1. As shown, each node is tested with two different hypotheses: (a) estimated position at time t resulting from inclusion of measurement for a feature i at time t, and

Table 1
Algorithm for computing data association for features

```
Set Hypothesis of all features to estimated
compute total node residual NDmin
STEP I:
for each feature i <- 1 to N
  Hypothesis(i) <- alternate(Hypothesis(i))
  compute total node residual Ndnew
  if(Ndnew < Ndmin)
    Ndmin <- Ndnew
    alteredFeature <- i
  else
    Hypothesis(i) <- alternate(Hypothesis(i))
end
push(Stack, alteredFeature)
push(alteredList, alteredFeature)

STEP II:
while(Stack is not empty)
  alteredFeature <- pop(Stack)
  for each feature (j <- 1 to m
    connected to alteredFeature
    & feature j is not in alteredList)
    Hypothesis(j) <- alternate(Hypothesis(j))
    compute node residual Ndnew
    if(Ndnew < Ndmin)
      Ndmin <- Ndnew
      push(Stack, j)
      push(alteredList, j)
    else
      Hypothesis(j) <- alternate(Hypothesis(j))
  end
  alteredFeature <- pop(Stack)
end
```

Alternate hypothesis for a node is predicted if the original hypothesis for the node i is estimated and vice-versa.

(b) predicted position resulting from the Kalman filters propagation step at time t.

4.5. Geometric template adaptation

The geometric template consists of a tree structure whose nodes are comprised of the point features on the target and the links connecting those features. A minimum spanning tree is used to connect the features on a target. Hence, for a target consisting of N features, the template consists of N nodes with $N-1$ links.

The geometric template deforms with target motion owing to feature translation. The structure is altered by (i) new features added to the target, (ii) removal of untracked features, or when (iii) the template contains features that do not truly belong to the target. The last case arises from the initialization of features that were never a part of the target, but were detected as part of the target owing to errors in the blob segmentation. The template is modified in each of the above cases either by: (i) addition or removal of links, (ii) by replacing parts or the whole template with a new template.

A new feature is added to a node closest to the feature as shown in Fig. 5. In the case of (ii), the template is updated by removing the links connected to the removed feature. In order to maintain the tree structure, the appearance is adapted by reconnecting the appropriate features. Fig. 6 illustrates examples for the removal of an internal node indicated by the shaded circle, and an external node (clear node). In this case, nodes are reconnected to preserve the tree structure without constructing a new tree such that the learned appearance model is not discarded. In the case of (iii), the features that do not truly belong to the target are removed from the appearance model. Such features are those that consistently lie outside the targets blob boundary. As mentioned in the previous paragraph, in any of case (i), (ii), and (iii) only part of the template is modified leaving the rest intact (other than the case when the tree is completely reinitialized with a new template). Hence, the estimates for the parts of the template containing the old links are preserved while new links are initialized in the Kalman filter, using the covariance

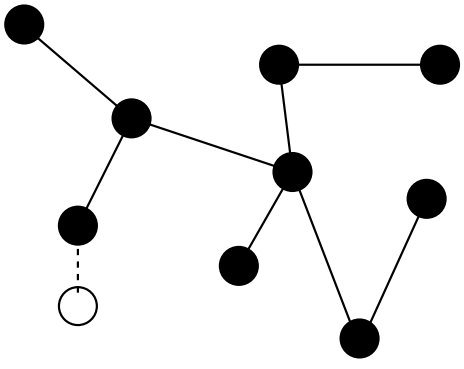


Fig. 5. Feature addition. The new feature, indicated by light circle, is added to the closest node in the template.

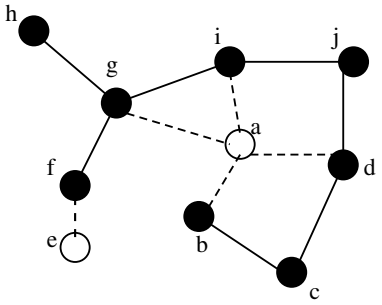


Fig. 6. Feature removal. The features to be removed are indicated by lightly shaded and clear circles. As shown, removal can be simply performed by deleting the node and its associated link for a external node like the clear circle, while removal of an internal node such as the lightly shaded circle, requires addition of new links among the remaining features.

$\Sigma_{ij} = \sqrt{\Sigma_i \Sigma_j}$, where Σ_i and Σ_j are the covariances in the position of the features i and j. The subscripts for x and y coordinates are omitted for clarity.

The geometric template is estimated using a variable state dimension Kalman filter. The state of the filter consists of the spatial distances between the features forming the links in the template. Since, only portions of the filter state corresponding to the newly added or removed links in the template are altered, the filter is merely augmented or diminished rather than initializing a new filter on every appearance update. A zeroth order motion model is used to model the spatial constraint of features. The filter state and transition matrices are expressed as,

$$X = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_{N-1} \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix},$$

$$H = \begin{bmatrix} 0 & -1 & 0 & \dots & 1 \\ 0 & 1 & \dots & -1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix}$$

where X corresponds to the state of the filter and $L_1, \ldots, L_{N-1}$ are the link lengths or spatial distance between features, A is the state transition matrix, and H is the observation transition matrix. Note that the dimensionality of H is $N-1 \times N$ since it transforms the individual feature positions to the links. The measurement error for the links is expressed as a product of the error covariances of the features forming the link.

5. Results

5.1. Experiment objective

The objective of the experiments was to validate the efficacy of the proposed tracking method in real-world image

sequences. Experiments evaluated the relative performance of the proposed cue combination strategy with a voting-based approach as in [1], and the performance of data association methods. In particular, the efficacy of the geometric template-based data association was compared with a Mahalanobis distance gated data association applied to individual features.

5.2. Experimental setup

Experiments were performed in outdoor as well as indoor video sequences obtained from a single camera mounted at a fixed location in each scene. Experiments in outdoor scenes used (a) a sparsely crowded color image sequence, Fig. 7, (b) a medium to densely crowded color image sequence, Fig. 9, (c) a medium crowded color image sequence with a distant view, Fig. 8, and (d) a medium crowded grayscale image sequence as shown in Figs. 10 and 20. Image sequences varied in length from 10 to 30 mins. The images were calibrated prior to the experiments using [21]. The resolution of images were 320×240. The measurements of target position and velocity are obtained in the image coordinates and the target state estimated in the scene coordinates. In all of the tested sequences, there were occlusions between targets as well as with the background.

5.3. Experimental results

Hypothesis I: *Weighted cue combination of all the available cues yields a more reliable tracker compared to a voting or a combination using normalized [0–1] weighting*. In order to test the above hypothesis, we compared the results of tracking using weighted cue combination with data association and a voting-based cue combination approach as in [1] in several outdoor traffic intersection video sequences. Each sequence differed in the kind of intersection (T-junction, 4-junction), the distance from the camera, and traffic density. In all these video sequences we tested the performance of the system in the presence of occlusions and illumination variations. For example, scene I as in Fig. 7 and scene III as depicted in Fig. 8 were fairly cluttered with occlusions (Scene I with around 44% and Scene III with around 78%) from other targets or from background. Illumination was not controlled in any of sequence as a result of which, there were portions where the target color resembled the background. Table 2 shows the comparison of the tracking results on two different outdoor scenes for the proposed weighted cue combination with data association method with the multiple cue-based tracking method as in [1].

Figs. 7–10 show examples of tracking multiple targets using the proposed approach consisting of weighted combi-

Fig. 7. Scene I: tracking with multiple occlusions. The geometric template derived from the features is overlaid on the targets. As can be seen, targets undergo multiple occlusions with other targets and some partial occlusion with the traffic pole. Also seen is that the target templates initialized during occlusion result in poor initialization, but as the target moves out of occlusion, a more appropriate template is initialized as indicated by target numbered 11 in (c), which was originally initialized as 9 with the truck in (a).

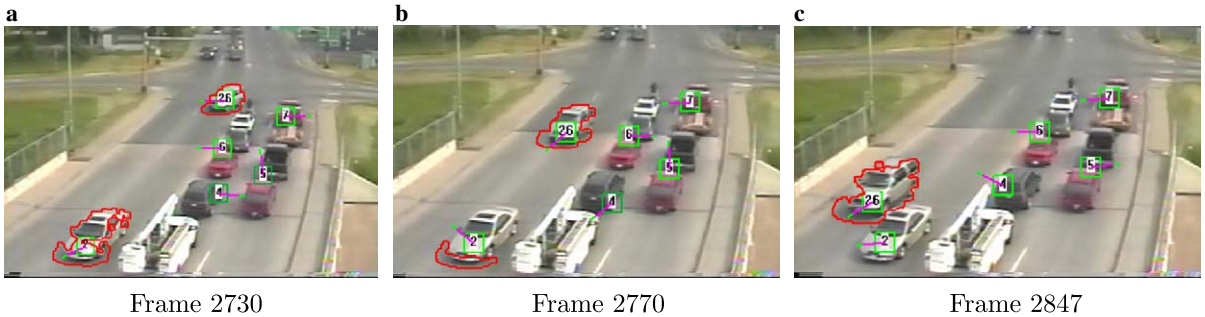


Fig. 8. Scene III: tracking in a crowded scene with stopped vehicles. The contours around the vehicles indicate the blobs detected. For vehicles not having the blob contour, the blob cue was missing.

Table 2
Comparison of tracking errors by the proposed method and voting-based multiple cue tracking method in the outdoor traffic sequences I and II

Scene	Error	Proposed method		Error	Vote-based tracking	
		Num targets	% of targets		Num targets	% of targets
I	Track lost	6	7.1	Track lost	21	24
	Track switch	2	2.3	Track switch	3	3.5
	Track drift	7	8.1	Track drift	3	3.5
II	Track lost	25	5.3	Track lost	39	9.0
	Track switch	20	4.6	Track switch	17	3.9
	Track drift	18	4.1	Track drift	31	7.2

The total number of vehicles in scene I were 86 while in scene II were 432.

nation of blob, color and feature cues with data association. An example of tracking in an indoor scene are also shown in Fig. 11. Figs. 12 and 13 show the comparison of tracking errors, namely, the mean error in the position estimate and the ground truth, and the variance in the state estimates expressed as the trace of the state error covari-

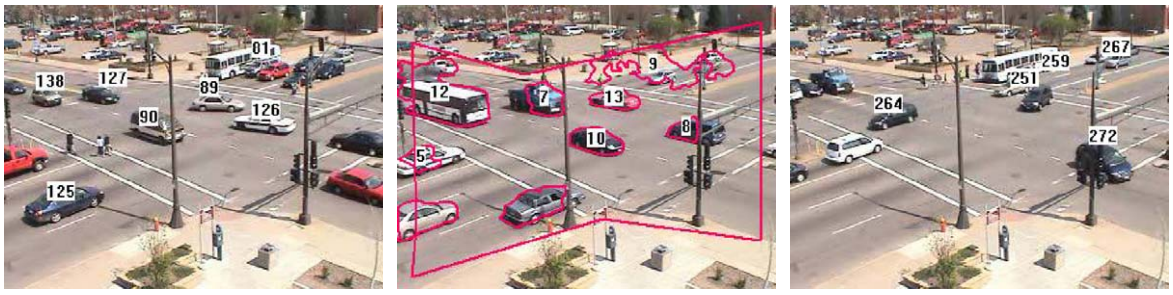
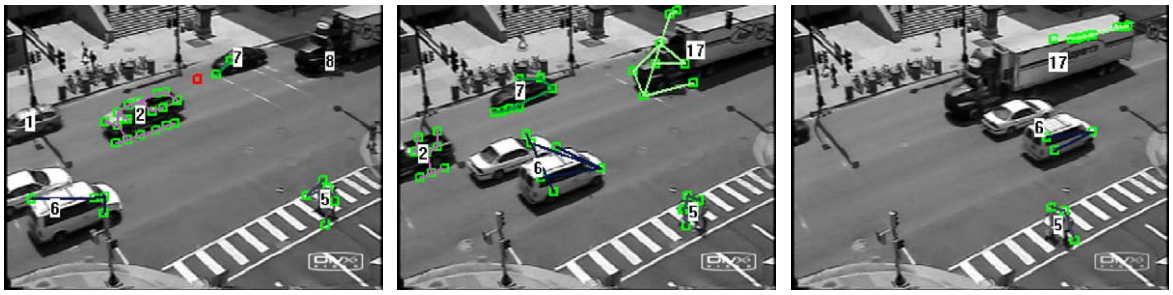


Fig. 9. Scene II: tracking vehicles in crowded scenes. The contours correspond to the blobs detected around the vehicles. Some of the contours are larger due to oversegmentation resulting from an illumination change.



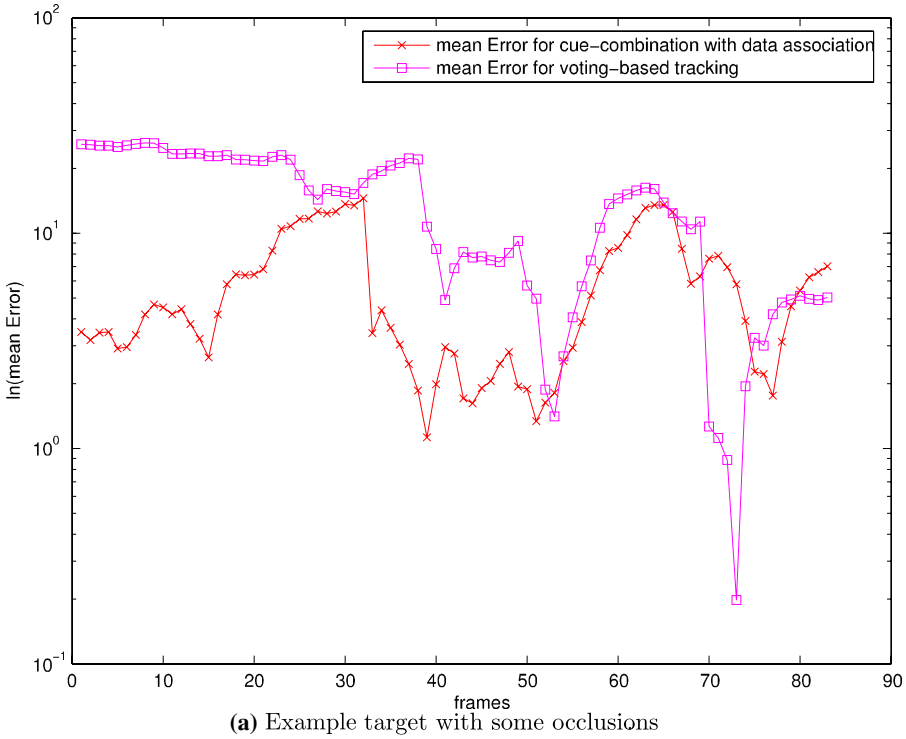
(a) Frame 246 **(b)** Frame 282 **(c)** Frame 321

Fig. 10. Scene IV: tracking in a medium crowded grayscale image sequence. Tracking is made more difficult in this sequence owing to lack of any color information.

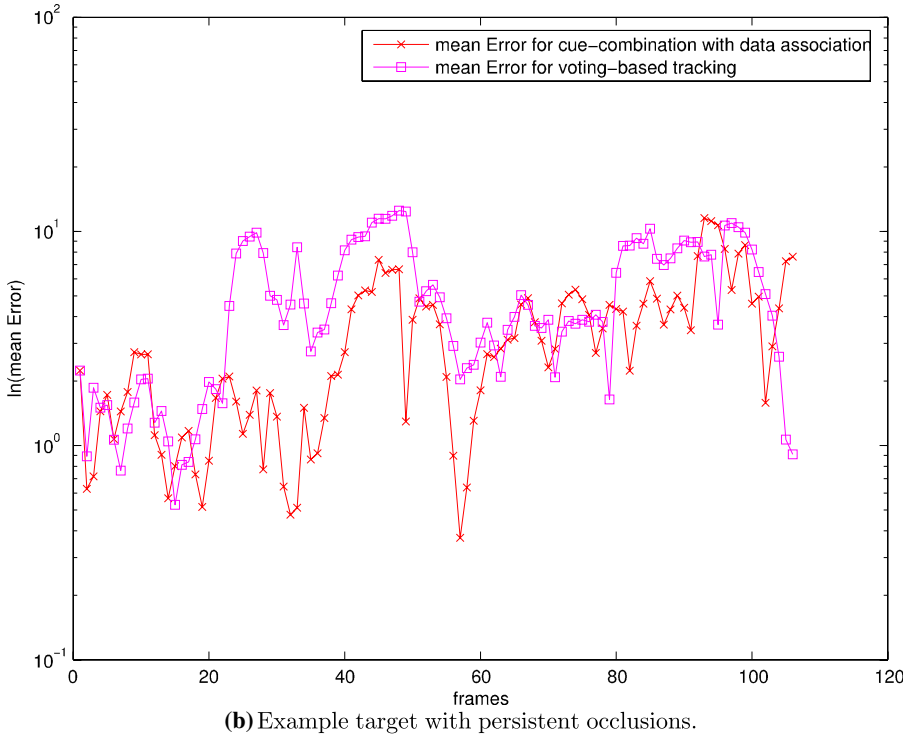
Fig. 11. Tracking pedestrians in an indoor scene. The pedestrians are tracked despite very similar background color and the occluding foreground.

ance, respectively, for the proposed tracking method and a voting-based cue combination method proposed by [1].

Hypothesis II: *Reliable tracking can be achieved even in the presence of persistent occlusions by using good data association methods.* Given that the JPDA is a standard method for computing data association, our experiments only evaluated the performance of the geometric template constrained data association method. In order to test the geometric template constrained data association, the experiments were performed using fairly crowded outdoor traffic intersection image sequences. The input to the tracker consists of velocity measurements obtained from the features



(a) Example target with some occlusions

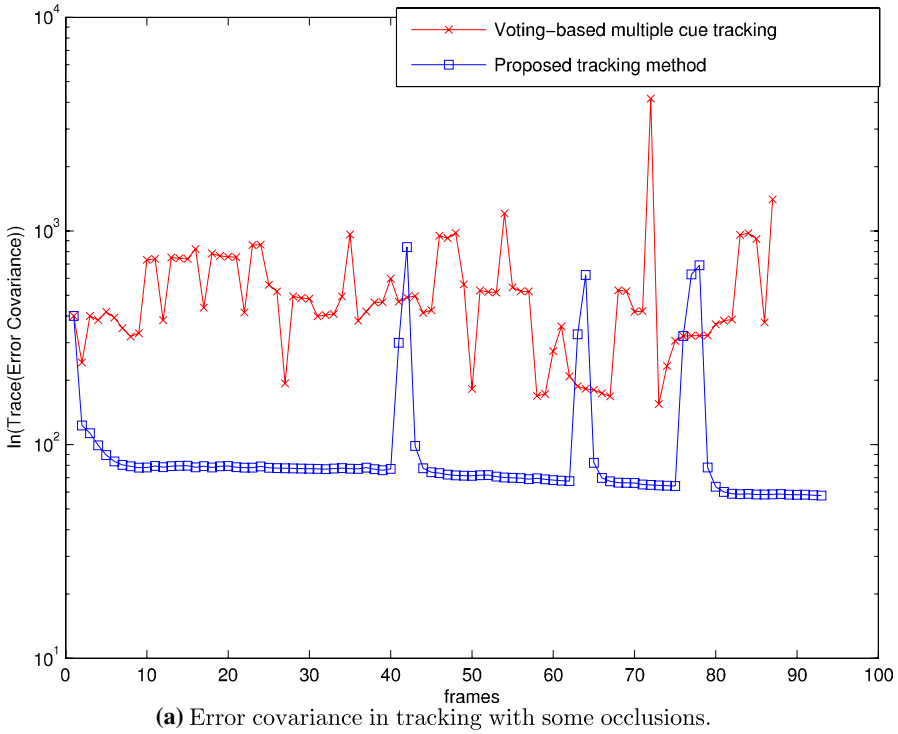


(b) Example target with persistent occlusions.

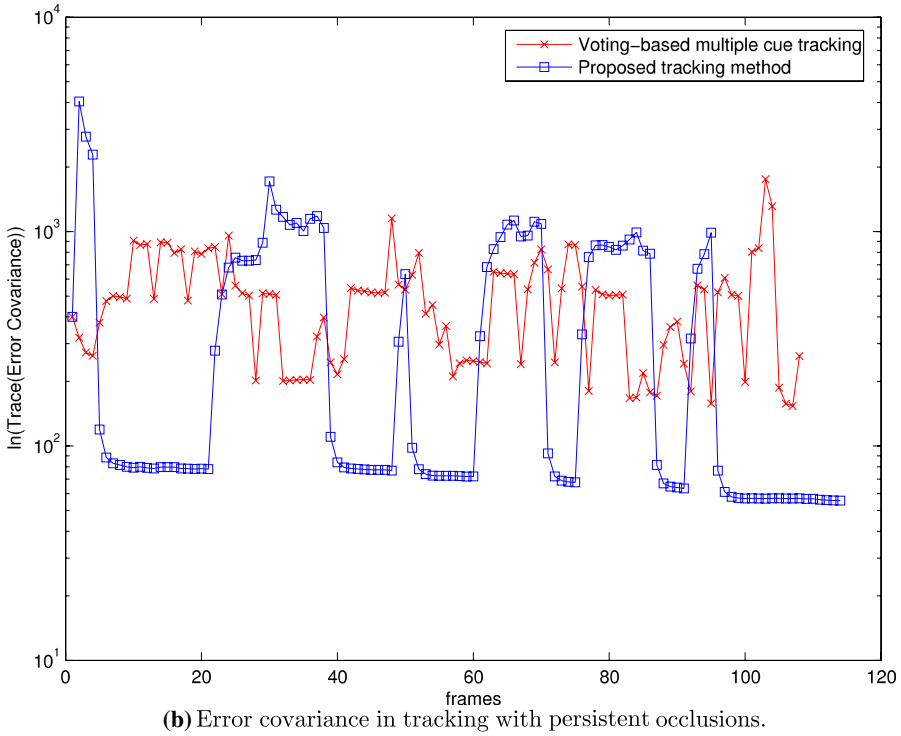
Fig. 12. Mean error in tracking using the weighted cue combination with data association tracker and voting-based multiple cue fusion method. The x-axis corresponds to the number of frames and the y-axis corresponds to the mean error expressed in log scale.

and position measurements obtained from standard motion segmented blobs. However, the measurements from features were weighted more heavily compared to the position measurements in order to reduce the influence of the blob tracker on the tracking.

Figs. 14–16 show the tracking errors for two different targets for the following two cases: (a) feature measurements constrained using Mahalanobis distance-based gating applied to individual features, and (b) feature measurements constrained using the geometric template. The plots show the trace of the error covariance in the estimated target state (position and velocity in the scene coordinates) with time. As shown in Fig. 14, the error covariance for the case where the feature measurements



(a) Error covariance in tracking with some occlusions.



(b) Error covariance in tracking with persistent occlusions.

Fig. 13. Variance in the error in tracking using the weighted cue combination with data association tracker and voting-based multiple cue fusion method. The x-axis corresponds to the number of frames and the y-axis corresponds to the trace of the error covariance expressed in log scale.

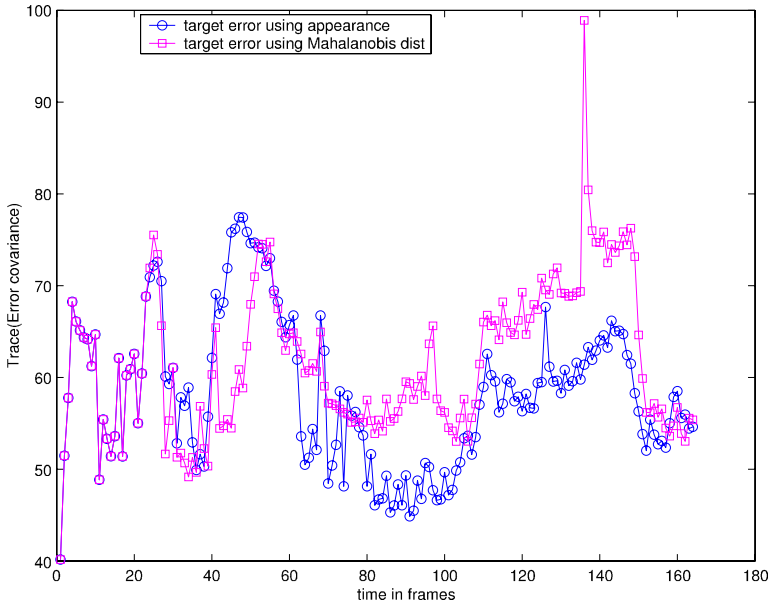


Fig. 14. Tracking error for feature tracking using Mahalanobis distance, and feature tracking using the geometric template.

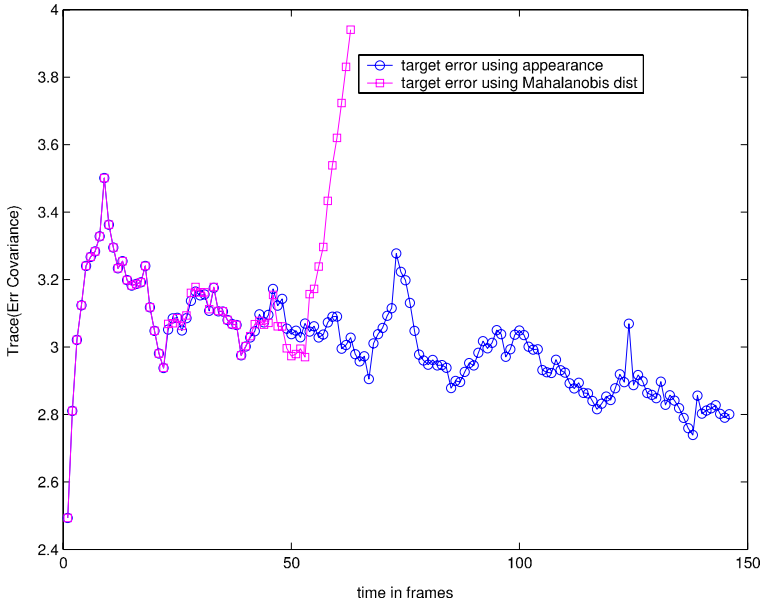


Fig. 15. Tracking errors for feature tracking using the Mahalanobis distance and geometric template constrained data association. The increase in error for the Mahalanobis distance constrained feature tracker after frame 40 is the result of the tracker switching to another target during an occlusion.

are constrained using the geometric template produces the lowest errors even during occlusions. Similarly, in Fig. 15, the increase in the error covariance after frame 50 for the Mahalanobis distance gated features is the result of tracker switching to another target, whereas, the geometric appearance constrained tracker produces consistent results without losing the target (Fig. 17).

Fig. 18 shows the trace of error covariance in the estimates of a geometric template with time. The increase in the error covariance in the beginning and the end of tracking occurs when new features are added thereby, changing the template. Fig. 19 depicts the results of tracking two targets in an indoor sequence through an occlusion.

6. Discussion

6.1. Cue combination

Combining multiple cues using the voting-based method essentially computes the weighted average of each measurement from all the cues and is updated once on the Kalman filter. Thus, for the position and velocity measurements,

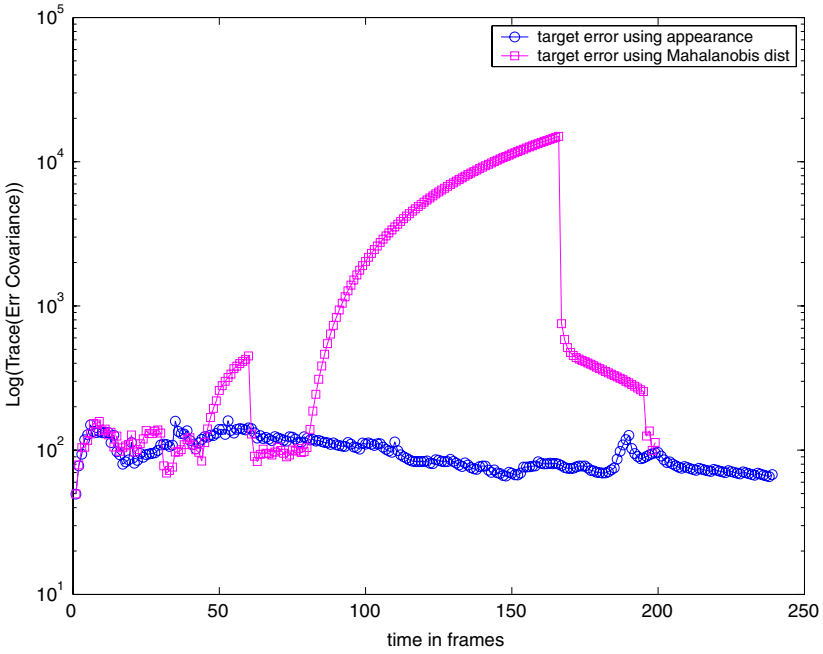
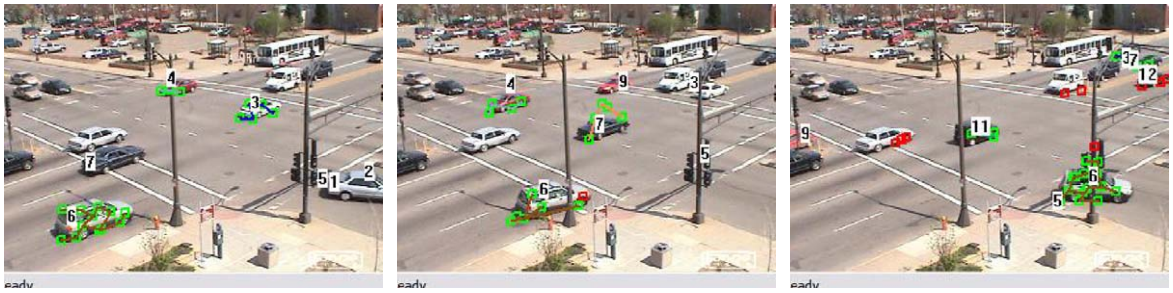


Fig. 16. Tracking errors for blob and feature tracking using the Mahalanobis distance, blob and feature tracking using the geometric template. As shown in the figure, the former method produces a large error as the target was lost for a few frames. The reduction in the error for the Mahalanobis distance based data association occurs after the frame 200 as the target is recaptured. On the other hand, the geometric template constrained method produces a consistent and low error in the tracking throughout the occlusion.



(a) Frame 1300 **(b)** Frame 1360 **(c)** Frame 1464

Fig. 17. Scene II: geometric template constrained tracking of a turning vehicle in an outdoor scene.

this method requires only two updates. In terms of algorithmic complexity in comparison to our method, the computation of the position measurements from each cue is the same, except for the additional data association employed for the blob and feature cues in our method and the additional Kalman filter updates required for each cue. Given that the number of cues is small, in our case 4; blob, color, feature, and an image template, the additional number of filter updates is only two. As such, our proposed incremental multiple cue fusion with data association based tracking achieves a tracking speed of 6–12 fps on a standard Pentium PC depending on the crowd density in the scene.

In terms of tracking accuracy, the voting-based tracking performs similar to the proposed method in the absence of any occlusion and slight illumination changes. However, the performance of the former method declines in the presence of occlusions and scene clutter. The results of tracking for both the voting-based tracking and the proposed incremental multiple cue combination with data association methods are presented in Table 2.

As shown in Section 5, combining information from multiple cues helps tracking under challenging conditions. Figs. 20 and 21 show the results of tracking under two such difficult conditions. In the case of Fig. 21, the movie used for tracking was highly compressed as a result of which, most information about the color of the targets was lost. Further, the presence of outside illumination and inter-reflections resulting from rain, and exhaust fumes from the vehicles made segmentation very difficult. Despite this, the system gave reasonable tracking performance. Similarly, in Fig. 20, although a grayscale image, we still get better tracking results by combining blob and color cues.

Table 4 shows the result of tracking for the two outdoor traffic video sequences, Scene I and Scene III as shown in

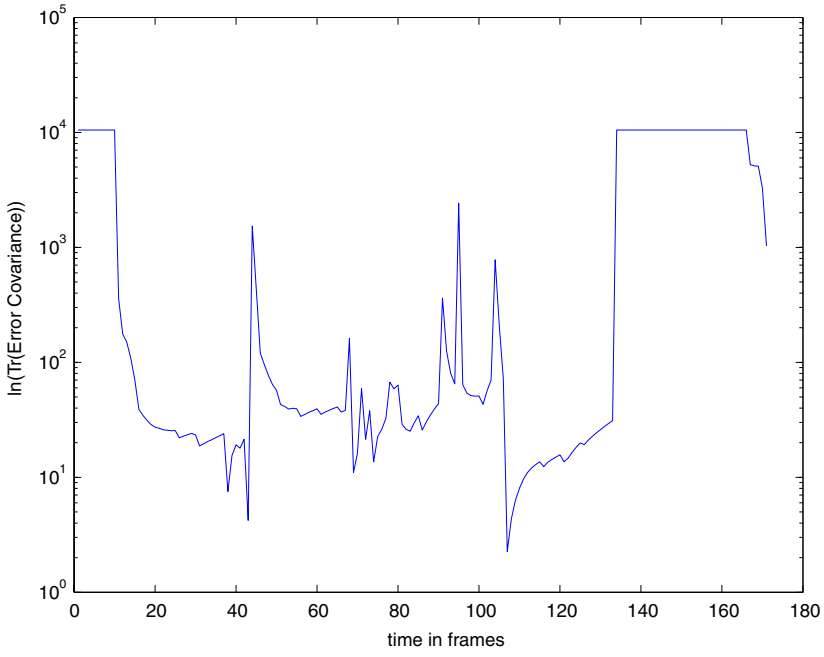
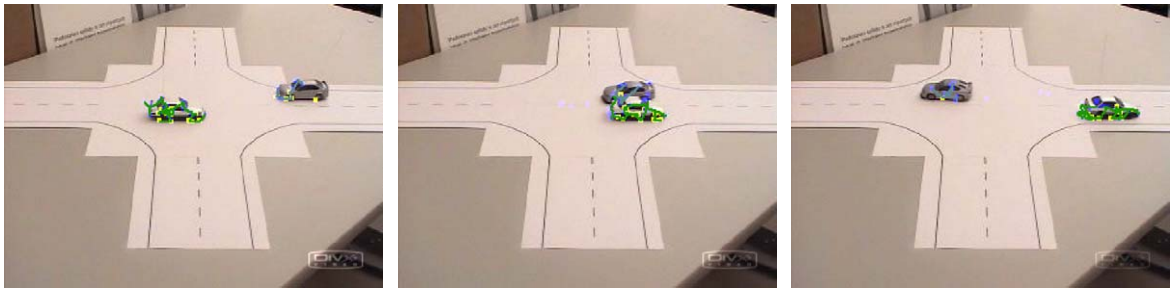


Fig. 18. Errors in geometric template estimation.



(a) Frame 350 **(b)** Frame 371 **(c)** Frame 394

Fig. 19. Indoor tracking sequence using blob and geometric features.

(a) Tracking under background occlusions. **(b)** Tracking under foreground occlusions.

Fig. 20. Example tracking in grayscale image sequence with occlusions.

Figs. 7 and 8. The corresponding scene statistics for the two sequences are in Table 3. The main sources of errors in target detection (missed targets, total and partial) in all the sequences resulted from the entry of multiple targets occluding each other. False positives or redundant segmentation resulted from the over-segmentation of the targets. Over-segmentation results during sudden illumination changes, targets moving behind background parts, as well as from

(a) Tracking under occlusions and inter-reflections. **(b)** Tracking under strong illumination variations.

Fig. 21. Example tracking in a noisy video sequence with occlusions, strong illumination and inter-reflections.

Table 3
Scene statistics for outdoor traffic sequences I and III

Scene statistics	I		III	
	Num targets	% of targets	Num targets	% of targets
Total number of targets	86		113	
Persistent occlusions	13	15	67	59.3
Passing occlusions	25	29	22	19.5
Stopped/slow-moving	8	9	30	26.5

The statistics were collected through manual inspection of the video sequences.

Table 4
Detection and tracking errors in the outdoor traffic sequences I and III

Scene	Error	Detection errors		Error	Tracking errors	
		Num targets	% of targets		Num targets	% of targets
I	Missed Targets (total, partial)	1, 7	1, 8	Track lost	6	7.1
	False positives	4	4.6	Track switch	2	2.3
	False negatives	8	9.3	Track drift	7	8.1
III	Missed targets (total, partial)	5, 6	4.4, 5.3	Track lost	17	15.7
	False positives	1	0.9	Track switch	10	9.3
	False negatives	5	4.4	Track drift	6	5.6

specular reflections from the target such as the windshield. Tracking failures mostly resulted from (i) occlusions, and (ii) stopped targets being modeled into the background which adversely affects the accuracy of the region segmentation. While the tracker can still continue to track a stopped target despite the absence of information from the blob cue, as the target moves, poor segmentation can result in track loss in the presence of a large number of targets.

Occlusions are classified as persistent or passing based on the duration of the occlusion. In all our experiments, occlusions lasting longer than 100 frames were classified as persistent occlusions, while occlusions lasting from 30 to 100 frames were classified as passing occlusions. As can be seen from the results of tracking, the system performs fairly well despite a large number of prolonged occlusions and stopped or slow moving targets.

6.2. Data association

The use of joint probabilistic data association for tracking has been well studied in the previous works [11,12]. The joint likelihood tracking method proposed by [22] helps obtain a robust data association for multiple targets by enumerating all possible depth orderings. However, when required to compute all possible depth orders for n targets, the number of hypotheses increases exponentially, thereby, limiting the number of targets for which this method is applicable without losing computational complexity. Fur-

thermore, for cues such as blobs, where the individual measurements themselves contain no information particular to the target, computing the image likelihood term for the data association is irrelevant.

As indicated by the results, geometric template constrained tracking helps attain a consistent tracking as well as minimizes the covariance of the estimated trajectory in comparison to the Mahalanobis distance constrained tracking. In comparison to methods such as [23–25] that compute a fundamental matrix through optimization, the proposed approach requires no such optimization. Furthermore, while the accuracy of the geometric template based data association will improve in the presence of stable features and with a stable estimate of the geometric appearance, there is no requirement for a large collection of features. This method requires no prior knowledge of the template. Thus, in comparison to adaptive mesh-based methods, no explicit physical models are required. The use of a loose appearance template to enforce the spatial and motion constraints between features on a target is simple and flexible. The enforced constraints are directly related to the certainty in the appearance estimate. In the worst case, when the appearance template has a large uncertainty, the data association method degenerates to individual feature measurement gating using the Mahalanobis distance. Tracking the features constrained by the geometric appearance based data association improves the results of tracking of the features significantly. Fig. 22 shows the average life of features detected using the Harris corner with normalized image correlation based matching and the SIFT features with the SIFT descriptor-based matching. For the given application, where the features undergo large translation and rotation, the geometric template constrained normalized correlation-based matching for Harris features performs nearly as well as the SIFT feature tracker. In fact as indicated in the Fig. 22, the geometric template constrained matching helps us to track a small number of features way beyond the time they are tracked using the SIFT feature detector and the nearest neighbor matching of the descriptors.

7. Conclusions and future work

This work presented a computationally simple, albeit robust tracking method for general unconstrained scenes using a set of simple trackers with appropriate data association and derived information-based cue combination. In particular, this work studied the problem of combining cues weighted by the individual uncertainties in the measurement through a Kalman filter framework. The results obtained from extensive experiments conducted on three different scenes indicate that reasonably good tracking can be obtained even in scenes with a large number of targets and occlusions, as long as the errors in the individual cues reflect the state of nature more or less accurately. The presented method addressed the problem of data association particularly for cues with low information such as blobs and features using a joint probabilistic data associa-

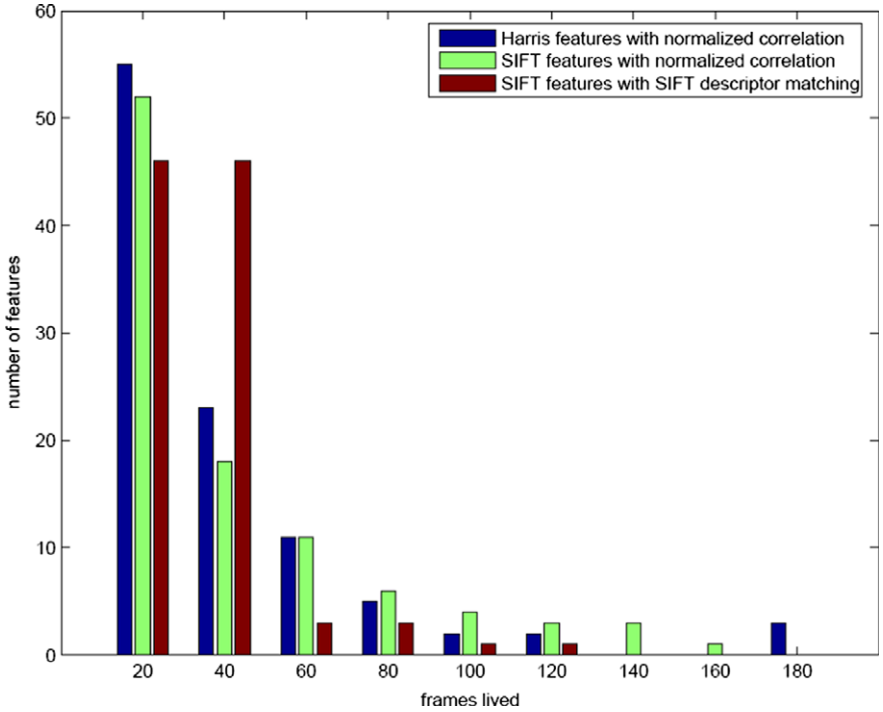


Fig. 22. Average feature life-time or the number of frames features for Harris features and SIFT features with normalized correlation with geometric template constrained matching, and SIFT features using SIFT descriptors for matching using nearest neighbors. The height of the bars indicates the number of features that were successfully tracked till the frames indicated in the x-axis. The number of features represented in the graph are normalized in the range 1–100 for all the feature trackers.

tion framework for the former, and a geometry constrained data association coupled with the knowledge of occlusion for feature tracking.

This work also considered the problem of tracking with a very weak model of target appearance. While using a motion segmented blob for initializing the target is computationally efficient, the main limitation of this approach is differentiating between multiple targets entering the scene occluded. This is one of the directions for future work. Again, in this work, blobs are used for detecting occlusions between targets as a result of data association. Occlusions resulting from static background occluders cannot be detected which can also affect tracking accuracy adversely, especially if the occluder is large. The above mentioned problems are being addressed in the current work.

Acknowledgments

This work has been supported in part by the National Science Foundation through grant IIS-0219863, Architecture Technology Corporation, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota. The authors also thank the reviewers for their useful comments and suggestions.

References

[1] D. Kragić, H. Christensen, Cue integration for visual servoing, IEEE Trans. Robot. Automat. 17 (1) (2001) 18–27.
[2] T. Zhao, R. Nevatia, Tracking multiple humans in complex situations, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1208–1221.
[3] Y. Azoz, L. Devi, M. Yeasin, R. Sharma, Tracking the human arm using constraint fusion and multiple-cue localization, Mach. Vision Appl. 13 (5-6) (2003) 286–302.
[4] S. Lu, D. Metaxas, D. Samaras, J. Oliensis, Using multiple cues for hand tracking and model refinement, in: Proc. Comput. Vision Pattern Recogn. Conf., 2003, pp. 443–450.
[5] C. Hue, J.L. Cadre, P. Pérez, Sequential Monte Carlo methods for multiple target tracking and data fusion, IEEE Trans. Signal Process. 50 (2) (2002) 309–325.
[6] T. Cham, J.M. Rehg, A multiple hypothesis approach to figure tracking, in: Proc. Comput. Vision Pattern Recogn. Conf., 1999, pp. 239–245.
[7] Y. Bar-Shalom, T.E. Fortmann, Tracking and Data Association, Academic Press, New York, 1987.
[8] P. Pérez, J. Vermaak, A. Blake, Data fusion for visual tracking with particles, Proc. IEEE 92 (3) (2004) 495–513.
[9] J. Vermaak, A. Doucet, P. Pérez, Maintaining multi-modality through mixture tracking, in: Proc. IEEE Intl. Conf. on Computer Vision, vol. 2, 2003, pp. 1110–1116.
[10] S. Atev, O. Masoud, N. Papanikolopoulos, Practical mixtures of Gaussians with brightness monitoring, in: Proc. IEEE Conf. on Intelligent Transportation Systems, 2004, pp. 423–428.
[11] Y. Bar-Shalom, X. Rongli, T. Kirubarajan, Estimation with applications to tracking and navigation, John-Wiley, New York, 2001.
[12] K. Birmiwal, Y. Bar-Shalom, On tracking a maneuvering target in clutter, in: IEEE Trans. on Aerospace and Electronic Systems, vol. AES-20, 1984, pp. 635–644.
[13] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, in: IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, 2003, pp. 564–577.
[14] K. Nickels, S. Hutchinson, Estimating uncertainty in SSD-based feature tracking, Image Vision Comput. 20 (1) (2002) 47–58.
[15] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.
[16] J. Triesch, C. von der Malsburgm, Democratic integration: self-organized integration of adaptive cues, Neural Computat. 13 (9) (2001) 2049–2074.
[17] R. Brunelli, D. Falavigna, Person identification using multiple cues, IEEE Trans. Pattern Anal. Mach. Intell. 17 (10) (1995) 955–966.
[18] J. Sherrah, S. Gong, Fusion of perceptual cues using covariance estimation, in: British Machine Vision Conference, vol. 2, 1999 pp. 564–573.
[19] R. Shumway, D. Stoffer, Dynamic linear models with switching, J. Am. Stat. Assoc. 86 (1991) 763–769.
[20] Z. Ghahramani, G. Hinton, Parameter estimation for linear dynamical systems. Tech. Rep. CRG-TR-96-2, University of Toronto, 1996.
[21] O. Masoud, N. Papanikolopoulos, Using geometric primitives to calibrate traffic scenes, in: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, vol. 2, 2004, pp. 1878–1883.
[22] C. Rasmussen, Joint likelihood methods for mitigating visual tracking disturbances, in: IEEE Workshop on Multi-Object Tracking, 2001, pp. 69–76.
[23] P.H.S. Torr, D. Murray, Outlier detection and motion segmentation, in: P.S. Schenker (Ed.), Sensor Fusion VI, vol. 2059, SPIE, 1993, pp. 432–443.
[24] T. Tommasini, A. Fusiello, E. Trucco, V. Roberto, Making good features track better, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, IEEE Computer Society, Silver Spring, MD, 1998, p. 178.
[25] D. Huynh, A. Heyden, Outlier detection in video sequences under affine projection, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 695–701.