Detection, Classification, and Collaborative Tracking of Multiple Targets Using Video Sensors

P.V. Pahalawatta, D. Depalov, T.N. Pappas, and A.K. Katsaggelos

ECE Dept. Northwestern University, 2145 Sheridan Rd, Evanston, IL 60208
{pesh, depalov, pappas, aggk}@ece.northwestern.edu

Abstract. The study of collaborative, distributed, real-time sensor networks is an emerging research area. Such networks are expected to play an essential role in a number of applications such as, surveillance and tracking of vehicles in the battlefield of the future. This paper proposes an approach to detect and classify multiple targets, and collaboratively track their position and velocity utilizing video cameras. Arbitrarily placed cameras collaboratively perform self-calibration and provide complete battlefield coverage. If some of the cameras are equipped with a GPS system, they are able to metrically reconstruct the scene and determine the absolute coordinates of the tracked targets. A background subtraction scheme combined with a Markov random field based approach is used to detect the target even when it becomes stationary. Targets are continuously tracked using a distributed Kalman filter approach. As the targets move the coverage is handed over to the "best" neighboring cluster of sensors. This paper demonstrates the potential for the development of distributed optical sensor networks and addresses problems and tradeoffs associated with this particular implementation.

1 Introduction

In the past few decades, we have seen many advances in wireless communication techniques and in microsensor technology. These advances combined with growing interest in both the military and the civilian domain in using sensor networks for remote monitoring applications have led to the concept of a wireless sensor network. A wireless sensor network can consist of a densely distributed set of sensors of various modalities (e.g., acoustic, seismic, infrared, imaging) that gather data from the physical environment and then process the data collaboratively to obtain a coherent high level description of the current state of the system.

Due to their low production costs and low energy consumption, acoustic and seismic sensors are among the most commonly studied types of wireless microsensors for battlefield surveillance. However, these sensors have some weaknesses. Since acoustic sensors depend on the acoustic signature of the target, they will not be able to detect a vehicle when it becomes stationary with its engines off. They can also be distracted by acoustic changes caused by gearshifts as well as accelerations and decelerations of a vehicle. Also, these sensors can be affected by acoustic noise caused by wind. Similar problems exist with seismic sensors.

F. Zhao and L. Guibas (Eds.): IPSN 2003, LNCS 2634, pp. 529–544, 2003.

We propose the use of multiple video sensors to enhance the capabilities of a wireless sensor network. Video sensors can track accelerating or decelerating targets with relative ease. They continue to "see" targets that become stationary even if the targets are completely silent. Also, video sensors can obtain unique attributes of a target such as its shape, color, and texture that can be used for classification as well as for pose estimation.

Automatic video-based vehicle surveillance has been studied mainly in the context of traffic monitoring applications. We can identify three main approaches that have been used with some success in these applications.

One approach uses three-dimensional models in order to classify a vehicle as well as to identify its position and orientation [1,2,3]. In this method, a sample taken from a database of geometrical wireframe models of possible vehicle shapes is projected on to the image plane and then compared with the object seen in the image. The main advantage of this method is that the vehicle can be classified as a part of the detection process. A disadvantage is that detailed geometrical models of vehicles must be available. Also, this approach can be very computationally intensive.

The second approach uses a contour of the motion-segmented image (i.e., pixels belonging to moving vehicle) to track the dynamics of the vehicle [4][5][8]. The weakness inherent to this method is that if multiple vehicles are in the field of view of the camera, and some vehicles are partially occluded by others as they are initially detected, then the vehicle contours cannot be correctly initialized.

The third approach, which is the one explored in this paper, simply tracks specific features within the vehicle instead of tracking the entire object. An example of a feature-based vehicle tracking system is presented in [6]. An advantage of this method is that some features of an object will still be visible even under partial occlusion.

The first phase or our system requires the detection of the moving target in each camera image. This can be achieved through background subtraction. An early approach to background subtraction was to assume that changes in intensity of a pixel that does not belong to a moving object can only occur due to camera noise and to model each pixel in the background to have a Gaussian intensity distribution. Then, for each pixel in a new frame, a significance test could be used to determine whether it belonged to the background model, or not [7]. However, this method assumes that the background image is completely static, which is not true for outdoor scenes involving foliage, or dust. One approach to deal with this problem has been to model each pixel with a mixture of Gaussians instead of as a single mode distribution [8]. In [9], a non-parametric approach is used to model the statistics of the background. In this case, one does not assume that the shape of the pdf of the pixel intensity is known, but instead, one assumes that the pixel intensities obtained from actual measurements represent samples taken from the pdf of the distribution. In this paper, we have used a simplified version of the approach proposed in [9] with a few modifications.

The next phase of our system is to compute matching feature points from images of the target taken by two cameras and by each camera at different points in time. Due to its key applications in the self-calibration of cameras and in object motion tracking, feature point correspondence is an area that has received much attention in the field of computer vision. The proposed methods can be placed in two broad categories based on the applications for which they are used.

The first category of methods can be used for applications in which the cameras are set up with a short baseline (the baseline is the distance between the centers of

projection of each camera) relative to the viewing distance of the object from each camera. In this case, the appearance of the images will be more or less uniform in the two cameras, and matching feature points will be within a searchable local area of the image.

In our application, however, the baseline between the cameras is unlikely to be small compared to the distance from each camera to the target vehicle. The main difference in a wide baseline setup is that different cameras will have significantly different viewpoints of the scene. Therefore, the image of an object will undergo a perspective transformation when it is viewed from a different camera. In this case, a direct correlation of the pixel intensity neighborhood will not provide a correct measure of the similarity between features. Also, feature detection itself becomes a much harder problem in a wide baseline setup because it is not guaranteed that different cameras will detect the same points of the object as feature points.

In [10] and [11], a scale space approach is used to detect scale invariant feature points in images. Typically, the points that can be detected consistently in images from different viewpoints are the points of the object that cause the local pixel intensities in the image to vary two-dimensionally. Such points are generally referred to as corners and a measure based on the horizontal and vertical image gradients can be used as a measure of their "cornerness" [12].

Even if the same feature points are detected from images in both cameras, the matching task is still difficult due to the significant differences in viewpoint between the two images. In [13], the concept of affine Gaussian scale space is introduced whereby image neighborhoods are smoothed using non-symmetric Gaussians in order to make them invariant to affine transformations. It is shown in [14] that affine scale space methods can be used for feature matching in wide baseline applications.

The feature point correspondences are used for camera calibration. There are two main approaches to camera calibration: (i) Calibration using a calibration object, usually a grid with features of known dimensions [15], and (ii) Self-calibration, which exploits the constraints contained in the images themselves (epipolar, image of the absolute conic) [16]. Due to the nature of our problem we must use a self-calibration technique since it does not require the placement of any foreign object in the scene.

2 Problem Formulation

We consider a scenario in which an approaching vehicle must be continuously detected and its position and velocity tracked by a set of video sensors located in the field. We assume the sensors have been placed arbitrarily in the field and that they are not calibrated. We also assume that the sensors are able to communicate with each other and that they are capable of using GPS or some other method to determine their position. We do not assume that the target movement is constrained in anyway other than that it will be moving on the ground plane.

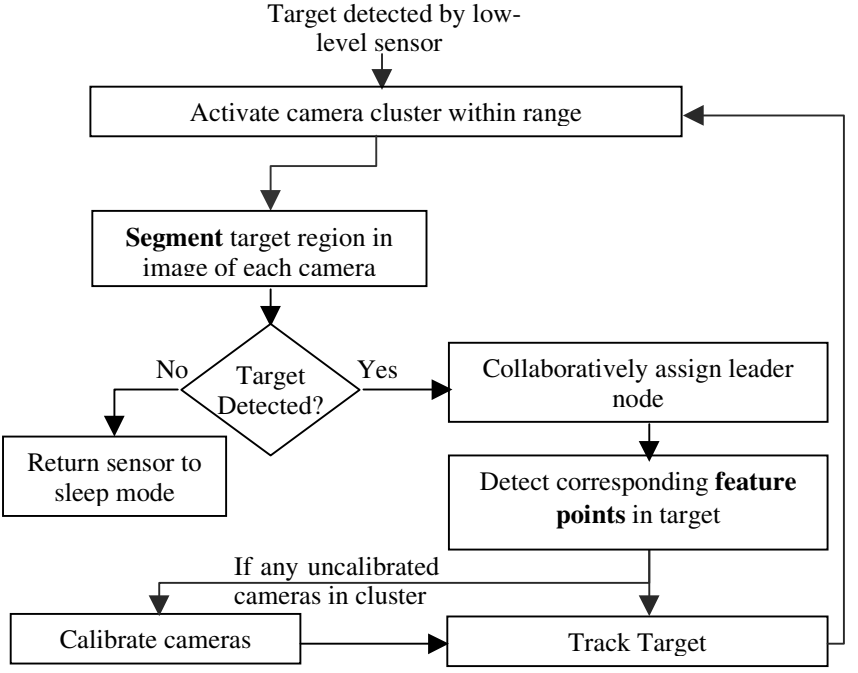

Fig. 1. Block diagram of system.

While the basic goal of the system is to simply detect and track any target vehicle that enters the sensor field, we must also consider the issue of power efficiency in the system. Wireless sensor nodes have access to a limited power supply, and therefore, we must utilize the available power in a way that would maximize the lifetime of the network. Since video sensors require a relatively large amount of processing, our system should be such that a video sensor is used only when a target is approaching the field of view of the sensor. Other less power-consuming sensors such as passive infrared sensors can be used as *tripwires* to turn on the video cameras in the perimeter of the sensor field. Also, due to the large energy cost associated with data transmission, we must avoid transmitting raw video data, and instead, transmit higher-level information generated at each sensor node whenever it is possible to do so.

The system we propose performs two main functions. The first is to automatically calibrate the video cameras in the sensor network based on point correspondences obtained from the moving target. The other is to use feature point correspondences obtained from subsequent frames in the video sequence combined with the camera calibration parameters, to detect the exact position of the target in the field. Then, we use this information to track the target over time and determine its velocity and predict its future state. A general block diagram of the proposed system is given in figure 1.

3 Background Subtraction

In our method, an initial estimate of the moving target region is obtained using a non-parametric model for the background — a method originally proposed in [9]. Then, this estimate is refined using spatial and temporal constraints within a Markov random field framework that has previously been used for image and video segmentation applications [17,18].

We can identify a few main requirements for the background subtraction algorithm. They are:

1. Adaptability to gradual changes in illumination
   As the time of day or weather conditions change, the lighting conditions of the system will also change. Therefore, it is essential that the background model be updated temporally based on the current lighting conditions of the scene.

2. Robustness to vacillations in background
   In outdoor scenes, trees waving in the wind can cause a particular pixel in the image frame to be a projection of a part of a leaf (green), a branch of the tree (brown), or the sky (blue). In all these cases, the particular pixel should be labeled as background although its intensity may differ significantly between successive frames.

3. Small training period
   Due to energy considerations in a wireless sensor network, the camera should not be expected to be on at all times. Therefore, the background subtraction algorithm needs to initialize and generate a background model within a few seconds.

4. Maintaining detection of objects that become stationary
   In our application, it is important to continue to detect a target vehicle for as long as possible even if it comes to a complete stop.

An approach based on the kernel density estimation technique presented in [9] can satisfy most of the requirements specified above. The basic idea behind this technique is that the underlying pdf of any distribution can be approximated by a weighted average of a set of kernel functions defined around sample data points taken from the distribution.

In this technique, we let $x_s(\mathbf{q})$ be an intensity value at location $\mathbf{q}$, and time s, that takes values from the set {0,…,255}. Then, we can estimate the probability that a new pixel at time t, has intensity $x_t(\mathbf{q})$ if it belongs to the background ($\mathbf{B}$) by,

$$\Pr(x_t \mid x_t \in \mathbf{B}) \approx \frac{1}{N} \sum_{s \in S_N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x_t - x_s)} \tag{1}$$

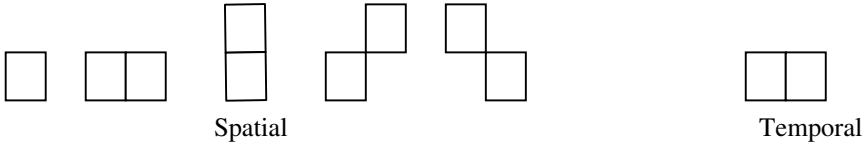

Fig. 2. Spatial and temporal clique shapes

where S_N is a set of N time instances prior to the current time t. Note that the pixel location **q** is omitted for clarity. Here, the kernel function is assumed to be a Gaussian with width σ. A suitable kernel width can be estimated from the sample data in the background pixels [9].

If the estimated probability is greater than a threshold, then the pixel can be labeled as a background pixel. Otherwise, it can be assumed that it belongs to a moving target.

3.1 Spatial and Temporal Constraints

We use a three-dimensional Markov random field (or equivalently, Gibbs random field) approach, previously used in image segmentation [17,18], to further refine the foreground segmentation. In this approach, each pixel in the image is modeled as belonging to two regions- background ($X_t = B$) and foreground ($X_t = B'$). Then, by Bayes theorem, the *a posteriori* probability density that a given pixel, X_t, is in the background can be expressed using the *a priori* density of the background process as:

$$p(X_t \mid x_t) \propto p(x_t \mid X_t) \cdot p(X_t) \tag{2}$$

where x_t is the intensity of the pixel. We have already shown how the density $p(x_t \mid X_t = B)$ can be found using kernel density estimation. The *a priori* density, $p(X_t)$, can be found by modeling the background region using a 3D Gibbs random field. This is done by assuming that the region process satisfies the Markov property. That is, if $\mathbf{N}_t(\mathbf{s})$ is the spatio-temporal neighborhood of a pixel in location **s** at time *t*, then

$$p[X_t(\mathbf{s}) \mid X_r(\mathbf{q}),\ \text{all}\,(\mathbf{q},r) \neq (\mathbf{s},t)] = p[X_t(\mathbf{s}) \mid X_r(\mathbf{q}),\ (\mathbf{q},r) \in \mathbf{N_t}(\mathbf{s})] \tag{3}$$

If this property is satisfied, the Gibbs density for the process can be expressed as

$$p(X_t) = \frac{1}{Z} e^{-\sum_C V_C(X_t)} \tag{4}$$

where Z is a normalizing constant, and $V_c(X_t)$ is the clique potential for a given clique C. We only use two-point cliques (spatial and temporal) and assume that all

Fig. 3. Some results of background subtraction algorithm. Vehicle traveling at 20mph.

one-point cliques have an equal potential of zero. The clique shapes are shown in figure 2. This amounts to an assumption that the probability of classification of the pixel depends only on the immediate (3x3 pixel) spatial neighborhood of the pixel, and temporally only on the previous and next pixel at the same location.

The two-point spatial and temporal clique potentials are defined such that for any two points **s** and **q** in a clique *C*, and for $\beta > 0$,

$$V_C(X) = \begin{cases} -\beta, & \text{if } X(\mathbf{s}) = X(\mathbf{q}), \quad \mathbf{s},\mathbf{q} \in C \\ +\beta, & \text{if } X(\mathbf{s}) \neq X(\mathbf{q}), \quad \mathbf{s},\mathbf{q} \in \mathrm{C} \end{cases} \tag{5}$$

3.2 Implementation

An important step in the model generation process is that of updating the background model. In our application, we wished to continue to detect a target even when it becomes stationary. We solve that problem by only updating the background pixels that do not belong to the detected foreground object.

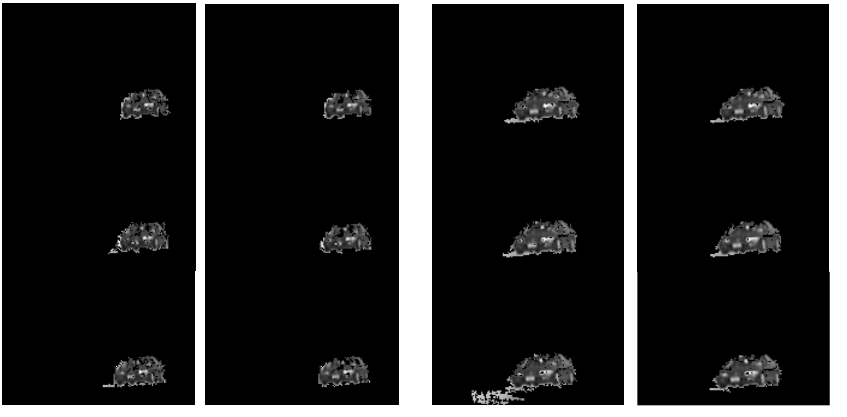

Fig. 4. Sequence of background frames before and after applying MRF

As input to the algorithm, we use the previously classified frame (background and foreground) and obtain the new classifications for the current and future frame based on the non-parametric model without GRF constraints. Then, a new classification for the current and future frame is found by adding spatial and temporal constraints as specified above. This is iterated until the number of pixels whose classification is changed over a new iteration is below a given threshold. The newly classified frame is now fixed and is used as input for the next iteration of the algorithm.

Figures 3 and 4 show some results of the background subtraction algorithm. In figure 4, we show the improvements made by including spatial and temporal constraints based on Markov random fields.

4 Feature Point Detection and Matching

This method uses a Harris detector [12] for the initial detection of affine invariant feature points. The Harris feature point detector attempts to detect points of interest within the image around which the image intensities change two-dimensionally. The image intensity variation is represented by the second moment matrix, $\boldsymbol{\mu}$, which is calculated using image gradient statistics over a neighborhood of each point.

$$\boldsymbol{\mu}(\mathbf{x}) = \int_{\mathbf{q}\in\mathbf{I}} g(\mathbf{x}-\mathbf{q},\sigma_s)\cdot\nabla(\mathbf{q},\sigma_d)\cdot\nabla(\mathbf{q},\sigma_d)^T d\mathbf{q} \tag{6}$$

where $g()$ is a Gaussian window with a scale of σ_s, $\mathbf{I}$ is the image intensity function, and

$$\nabla(\mathbf{q},\sigma_d) = g(\sigma_d) * \mathbf{L}(\mathbf{q}) \tag{7}$$

where $g(\sigma_d)$ is a Gaussian and $\mathbf{L(q)}$ is the image gradient function evaluated at $\mathbf{q}$. The Gaussian function is used to smooth the noise in the original image.

It has been shown in [12] that we can define a corner strength measure, C(**x**), which represents a point whose neighborhood exhibits significant intensity variations in both dimensions as,

$$C(\mathbf{x}) = \det(\boldsymbol{\mu}(\mathbf{x})) - k\cdot trace^2(\boldsymbol{\mu}(\mathbf{x})) \tag{8}$$

where k is an empirically determined constant. Points with corner strengths above a given threshold could be considered to be interest points.

We can determine the best feature points in the image by choosing the points that give the maximum corner strength according to the Harris measure over all integration and derivation scales. However, since the error in the localization of a feature point is increased with increasing scale, we localize the detected interest point in the smallest scale using a method similar to that proposed in [19].

4.1 Affine Gaussian Scale Space

Affine Gaussian scale space is presented in [14], as a framework within which to solve the wide baseline correspondence problem. An important assumption in using this method is that locally smooth regions of the image of an object will only undergo an affine transformation when viewed from different viewpoints.

The difference between affine Gaussian and linear scale space is that in the former, the Gaussian functions used for convolution of the image prior to finding the second moment matrix will not be rotationally symmetric. Therefore, the scale parameter for an affine Gaussian window will be a covariance matrix instead of a scalar variance.

Then, the second moment matrix of a point in affine Gaussian scale space is,

$$\boldsymbol{\mu}(\mathbf{x};\boldsymbol{\Sigma}_d;\boldsymbol{\Sigma}_s) = \int_{\mathbf{q}\in\mathbf{I}} g(\mathbf{x}-\mathbf{q},\boldsymbol{\Sigma}_s)\cdot\nabla(\mathbf{q},\boldsymbol{\Sigma}_d)\cdot\nabla(\mathbf{q},\boldsymbol{\Sigma}_d)^T d\mathbf{q} \tag{9}$$

where, $\boldsymbol{\Sigma}_d$, and $\boldsymbol{\Sigma}_s$ are the covariance matrices associated with the scales of derivation and integration.

Assume that the second moment matrix in affine Gaussian scale space of a given image $\mathbf{L}$, is shown to be $\mathbf{M}_\mathbf{L}$, and

$$\boldsymbol{\Sigma}_{\mathbf{d,L}} = t\mathbf{M}_\mathbf{L}^{-\mathbf{1}} \text{ and } \boldsymbol{\Sigma}_{\mathbf{s,L}} = s\mathbf{M}_\mathbf{L}^{-\mathbf{1}} \tag{10}$$

Then, if $\mathbf{q}_\mathbf{R}$ is a point in a transformed image, $\mathbf{R}$, such that $\mathbf{q}_\mathbf{R} = \mathbf{A}\mathbf{q}_\mathbf{L}$ and, $\boldsymbol{\mu}_\mathbf{R} = \mathbf{M}_\mathbf{R}$, it can be shown [13] that,

$$\boldsymbol{\Sigma}_{d,\mathbf{R}} = t\mathbf{M}_\mathbf{R}^{-\mathbf{1}}, \text{ and } \boldsymbol{\Sigma}_{s,\mathbf{R}} = s\mathbf{M}_\mathbf{R}^{-\mathbf{1}} \tag{11}$$

Therefore, the fixed point conditions are preserved under linear transformations. Moreover, it is shown in [14] that if we define $\mathbf{L}'$ to be a *square root transformed* image of $\mathbf{L}$, such that $\mathbf{L}'(\mathbf{x}) = \mathbf{L}(\mathbf{M}_\mathbf{L}^{-\frac{\mathbf{1}}{\mathbf{2}}}\cdot\mathbf{x})$, then

$$\boldsymbol{\mu}'_\mathbf{L}(\mathbf{q}'_\mathbf{L};t\mathbf{I},s\mathbf{I}) = \mathbf{I} \tag{12}$$

Since the same would be true for images $\mathbf{R}$ and $\mathbf{R}'$, and assuming that the affine transformation from $\mathbf{L}'$ to $\mathbf{R}'$ can be written as $\mathbf{A}'$, we get,

$$\boldsymbol{\mu}'_\mathbf{L}(\mathbf{q}'_\mathbf{L};t\mathbf{I},s\mathbf{I}) = \mathbf{I} = \mathbf{A}'^{\mathbf{T}}\boldsymbol{\mu}'_\mathbf{R}(\mathbf{q}'_\mathbf{R};t\mathbf{I},s\mathbf{I})\mathbf{A}' = \mathbf{A}'^{\mathbf{T}}\mathbf{A}' = \mathbf{I} \tag{13}$$

Therefore, $\mathbf{A}'$ is a rotation matrix. This implies that, if we are given two images where one is a linear transformation of the other, and we can find the fixed points for each image, then the square root transformed versions of the two images will be related by a simple rotation.

In the wide baseline matching application, we can obtain the local neighborhoods of points detected by the multi-scale Harris feature detector, and find their corresponding square root transformed image neighborhoods. Then, we can use conventional rotation invariant descriptors to represent the transformed images and match them using the minimum distance between such descriptors.

4.2 Implementation

In our implementation, we calculated the corner strength of each point in the image at multiple values of σ_s and σ_d. The values of σ_d were kept proportional to σ_s, and σ_s was chosen to be in the range [2.0, 16.0]. The points with the maximum corner strengths across all possible scales were considered to be feature points of the image.

Then, the goal was to transform the local neighborhood, **L**, around each feature point, **x**, to a fixed point and to find its square root form. This transformation, **A**, is accomplished by iterating through the following basic steps.

1) Set $\mathbf{A}^{(0)} = \mathbf{I}$.
2) Set $\mathbf{L}'(\mathbf{q}) = \mathbf{L}(\mathbf{A}^{(k)} \cdot \mathbf{q})$.
3) Find $\boldsymbol{\mu}'_{\mathbf{L}}(\mathbf{x}; t\mathbf{I}, s\mathbf{I})$ where *t,s* are kept constant and equal to the characteristic scales found by the multi-scale Harris detector.
4) Normalize $\boldsymbol{\mu}'_{\mathbf{L}}$ to have a unit determinant.
5) If $\boldsymbol{\mu}'_{\mathbf{L}} \neq \mathbf{I}$, then set $\mathbf{A}^{(k+1)} = (\boldsymbol{\mu}'_{\mathbf{L}})^{-\frac{1}{2}} \cdot \mathbf{A}^{(k)}$.
6) Normalize $\mathbf{A}^{(k)}$ by its largest eigenvalue.
7) If $\boldsymbol{\mu}'_{\mathbf{L}} = \mathbf{I}$, then stop. Otherwise, return to step 2.

The normalization of the second moment matrix in step 2 amounts to a rescaling of the local intensity values in the neighborhood of the pixel and the normalization of the transformation matrix in step 6 will ensure that the original image will not be under-sampled.

4.3 Feature Point Correspondence

Before matching the transformed image neighborhoods around the feature points obtained using the above method, they need to be made invariant to changes in intensity since they are viewed by different cameras from different viewpoints. We have used a simple approach, which consists of normalizing each pixel in the feature neighborhood by the maximum intensity value for the neighborhood.

Then, since the images obtained from the two viewpoints will be similar only up to a rotation, the next step is to obtain rotation invariant feature descriptors for each of the images. The descriptors we use are based on a method suggested in [14] and consist of a vector of 15 elements, which correspond to higher order derivatives of the image.

Once the descriptors are obtained we match them using the minimum Mahalanobis distance between each two descriptors taken from different viewpoints. If a point in one image is close to multiple points in another image with a larger spatial variance, then we discard the point since the matching is too ambiguous. Also, if a point does not have any points in the other image which are within a threshold distance, that point is discarded since it may not exist at all in the other image.

4.4 Results

The feature point matching algorithm is successful if the difference in viewpoint between two images is within reasonable limits. For example, the matching algorithm detected 20 corresponding points between figure 5(a) and figure 5(b). Of them, 17 were correct matches. However, of the 17 detected correspondences between 5(a)

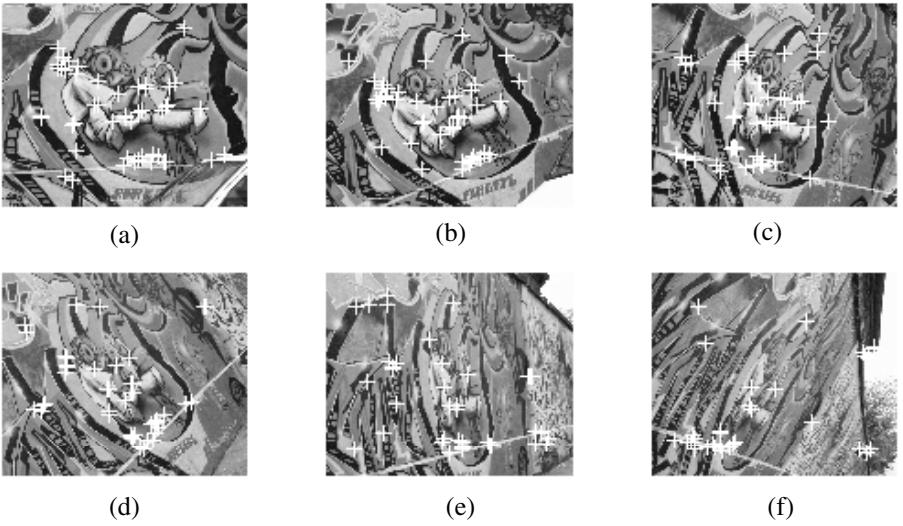

Fig. 5. Results of corner detection. White crosses indicate position of detected corners.

and 5(c), only 9 were correct matches. This shows that, in order to be used for self-calibration, we will need to take images of the target from cameras that have relatively similar viewpoints. On the other hand, if the cameras are already calibrated, then we can use the epipolar constraint to find better correspondences, and a relatively low number of correct correspondences will be sufficient to perform feature point based tracking.

5 Camera Calibration

The sensors need to be calibrated for their intrinsic and extrinsic parameters. Intrinsic camera parameters describe image formation, and they are focal length, aspect ratio, principal point and skew. Extrinsic camera parameters describe position and orientation of the cameras relative to some reference frame and they are described in terms of translation and rotation.

We assume that the relationship between the world coordinates, $[x\ y\ z]$, and the pixel coordinates, $[u\ v]$, is linear projective. This allows for use of projective geometry, which greatly simplifies mathematical representation. In the new generation of cameras distortion is reasonably small, and this model is a good approximation.

$$\begin{bmatrix} u & v & s \end{bmatrix}' = \mathbf{P}\begin{bmatrix} x & y & z & 1 \end{bmatrix}', \ \mathbf{P} = \mathbf{A}[\mathbf{R} \mid \mathbf{t}] \tag{14}$$

Here **A** is the intrinsic parameter matrix and **R, t**, describe the rotation and translation parameters.

In addition, we can use several more simplifying assumptions about the camera model that will ease our calibration task and will not seriously degrade the accuracy of reconstruction. Skew can be assumed to be equal to zero, $\theta=\pi/2$, (reasonable for new generations of cameras), and the principal point, $[u_0, v_0]$, can be assumed to be at the center of the image. It is well known that variation in location of the principal point of several pixels does not affect the reconstruction in a great manner [15].

$$\mathbf{A} = \begin{bmatrix} fk_u & fk_u \cot\theta & u_0 \\ & \dfrac{fk_v}{\sin\theta} & v_0 \\ & & 1 \end{bmatrix} \rightarrow \mathbf{A} = \begin{bmatrix} fk_u & & \\ & fk_v & \\ & & 1 \end{bmatrix} \tag{15}$$

If aspect ratio is known in advance (from manufacturers specifications,) and if we have a good guess for the focal lengths of each camera, we are then able to reconstruct the scene from one snapshot of the stereo pair.

Scene reconstruction involves obtaining pixel correspondences between a pair of images, which are used to estimate the fundamental matrix, **F** [20]. The fundamental matrix defines an epipolar constraint between images in terms of pixels. Since the estimation of the fundamental matrix is very sensitive to errors in feature point correspondences, and our Harris feature based matcher can produce some false matches, we use the random sample and consensus algorithm (RANSAC) [21] using the epipolar constraint as a criterion to detect false matches and eliminate outliers. For corresponding points $\mathbf{m}_2$ and $\mathbf{m}_1$ in two images, the epipolar constraint is expressed as,

$$\mathbf{m}_2^{\mathrm{T}}\mathbf{F}\mathbf{m}_1 = 0 \tag{16}$$

F is calculated using a normalized eight-point algorithm [22].

Knowing the intrinsic parameters and the fundamental matrix we can calculate the essential matrix, which can further be decomposed into rotational and translational components to obtain initial guesses for the extrinsic parameters [23]. We then optimize the results in terms of the discrepancy from the epipolar constraint by solving a nonlinear least squares problem [15].

$$\min\left[\sum_{i=1}^{m}\left((\mathbf{m}_{2i})^{\mathrm{T}}\mathbf{A}_2^{-\mathrm{T}}\mathbf{T}\mathbf{R}\mathbf{A}_1^{-1}\mathbf{m}_{1i}\right)^2\right] \tag{17}$$

Here **T** is a skew symmetric matrix made from the translation vector, $\mathbf{A}_2$, and $\mathbf{A}_1$ are the intrinsic matrices of the two cameras, and $\mathbf{m}_2$, $\mathbf{m}_1$ are corresponding points.

If we do not have an accurate guess for the focal lengths, we can obtain them by self-calibrating the cameras. Since we have only one unknown intrinsic parameter for each camera, we need only one synchronized snapshot from each camera to be able to solve for the focal lengths. To self-calibrate cameras, we need to solve the set of Kruppa equations. Kruppa equations require the fundamental matrix to be known and they relate the correspondence of epipolar lines tangent to a dual image of the absolute conic [24].

In the more general case, if we do not know the aspect ratio in advance, then we can still self-calibrate the cameras by using two snapshots of a moving target taken from each camera. Then, if we only obtain correspondence points detected within the target, we can equate the motion of the target to a motion of the stereo rig. It has been shown in [25] that this provides enough additional constraints to solve for the unknown intrinsic parameters.

Since with this approach we can only reconstruct the scene up to an unknown scale factor, we need some external information to perform the metric reconstruction. For example, if the cameras are equipped with GPS device, then we can obtain the scale factor by calculating the baseline distance between the cameras. Figure 6 shows results of the camera calibration algorithm and metric reconstruction procedure.

Fig.6. Original images and reconstructed scene using two cameras

6 Tracking Results

For the tracking experiment, we used a sequence of images taken from two cameras in a wide baseline setup. The images were taken with a resolution of 1024x768 pixels and they consisted of two moving objects in an indoor environment. We assumed that the focal lengths of the cameras were known and that aspect ratios were equal to one.

After the cameras were calibrated, the detected feature points on the target were tracked over the entire sequence. The tracking was performed using a Kalman filter. We assume a linear constant velocity dynamic model for the Kalman filter. Figure 7 shows an example of a tracked point as the object moves in the field of view of both cameras.

Fig. 7. Tracked Feature Point. Top: right camera view, bottom: left camera view

The position and velocity plots of the point are shown in figure 8. The position of the point is shown relative to the XZ plane in the camera coordinate system. This corresponds to viewing the trajectory of the point from above. There are some missing points in the position plot that correspond to frames in which the feature points could not be extracted with sufficient certainty. There are also a couple of outliers that are caused by false point correspondences between the images. The velocity plot shows some deviation from the ground truth due to errors in the metric reconstruction.

7 Conclusions

We have concluded that computer vision based target tracking is a viable approach for a wide-baseline configuration involving multiple cameras. Feature point based tracking algorithms enable real time operation, and also reduce communication requirements between sensors. The main difficulty in this approach is establishing wide-baseline feature point correspondences from uncalibrated camera views for the purpose of camera calibration. We plan to further investigate this topic in the future.

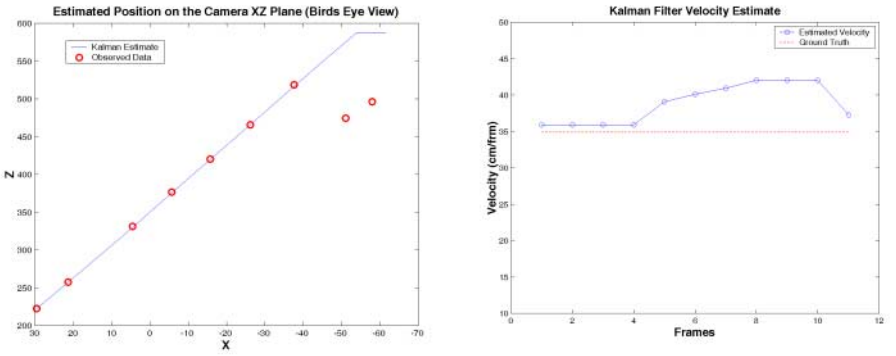

Fig. 8. Position and velocity estimates of tracked point.

Acknowledgements. This project has been funded in part by the Defense Advanced Research Projects Agency (DARPA) through the Sensor Information Technology Program (SensIT).

References

1. G. D. Sullivan, "Visual Interpretation of Known Objects in Constrained Scenes," *Philosophical Transactions B of the Royal Society*, vol. 337, pp. 361–370, 1992.
2. D. Koller, K. Daniilidis, and H.-H. Nagel, "Model-based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *International Journal of Computer Vision*, vol. 10, pp. 257–281, 1993.
3. J. M. Ferryman, S. J. Maybank, and A. D. Worall, "Visual Surveillance for Moving Vehicles," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 187–197, 2000.
4. D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," *Proceedings of the European Conference on Computer Vision*, vol. 1, pp. 189–196, 1994.
5. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, "Towards Robust Automatic Traffic Scene Analysis in Real-Time," *Proceedings of the 12th International Conference on Pattern Recognition*, pp. 126–131, 1994.
6. D. Beymer, P. McLauchlan, B. Coifmann, J. Malik, "A Real-Time Computer Vision System for Measuring Traffic Parameters," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 495–501, 1997.
7. T. Aach and A. Kaup, "Statistical Model-based Change Detection in Moving Video," *Signal Processing*, vol. 31, pp. 165–180, 1993.
8. C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 22–29, 1999.
9. A. Elgammal, R. Duraiswamy, D. Harwood, and L. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, July 2002.

10. D. G. Lowe, "Object Recognition from Scale Invariant Features," *Proceedings of the International Conference on Computer Vision*, pp. 1150–1157, 1999.
11. K. Mikolajczyk, and C. Schmid, "Indexing Based on Scale Invariant Interest Points," *Proceedings of the 8th International Conference on Computer Vision*, pp. 525–531, 2001.
12. C. Harris, and M. Stephens, "A Combined Corner and Edge Detector," *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988.
13. T. Lindeberg, and J. Gårding, "Shape-Adapted Smoothing in Estimation of 3-D Shape Cues from Affine Distortions of Local 2-D Brightness Structure," *Image and Vision Computing*, vol. 15, no. 6, pp. 415–434, 1997.
14. A. Baumberg, "Reliable Feature Matching Across Widely Separated Views," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 774–781, 2000.
15. Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 11, November 2000.
16. M. Pollefeys, R. Kock, and L. Van Gool, "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Intrinsic Camera Parameters," *International Journal of Computer Vision*, vol. 32, no. 1, pp. 7–25, Aug 1999.
17. T. Pappas, "An Adaptive Clustering Algorithm for Image Segmentation," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, April 1992.
18. R. O. Hinds, T. N. Pappas, "An Adaptive Clustering Algorithm for Segmentation of Video Sequences," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. 2427–2430, May 1995.
19. A. P. Witkin, "Scale-Space Filtering," *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pp. 1019–1022, 1983.
20. Q.T. Luong, O. Faugeras, "The Fundamental Matrix: Theory, Algorithms and Stability Analysis,"*The International Journal of Computer Vision*, vol.17, pp 43–76, 1996.
21. M. A. Fischler, R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of the ACM*, Vol 24, pp 381–395, 1981.
22. R. Hartley, "In Defence of the 8-Point Algorithm," *ICCV*, pp. 1064–1070, 1996.
23. O. Faugeras, Q.T. Luong, "The Geometry of Multiple Images," The MIT Press, 2001.
24. R. Hartly, A. Zissereman, "Multiple View geometry", Cambridge University Press, 2000.
25. Z. Zhang, Q.-T.Luong, and O. Faugeras, "Motion of an Uncalibrated Stereo Rig: Self-Calibration and Metric Reconstruction," *IEEE Trans. on Robotics and Automation,* vol. 12, no. 1, pp. 103–113, 1996.