ACM Multimedia Systems Journal, Special Issue on Video Surveillance. Accepted Jan 2004.

Motion Analysis for Event Detection and Tracking with a Mobile Omni-Directional Camera

Tarak Gandhi
Computer Vision and Robotics Research Laboratory
University of California – San Diego
tgandhi@ucsd.edu

Mohan M. Trivedi
Computer Vision and Robotics Research Laboratory
University of California – San Diego
mtrivedi@ucsd.edu

ABSTRACT

A mobile platform mounted with Omni-Directional Vision Sensor (ODVS) can be used to monitor large areas and detect interesting events such as independently moving persons and vehicles. To avoid false alarms due to extraneous features, the image motion induced by the moving platform should be compensated. This paper describes a formulation and application of parametric ego-motion compensation for an ODVS. Omni images give 360 degrees view of surroundings but undergo considerable image distortion. To account for these distortions, the parametric planar motion model is integrated with the transformations into omni image space. Prior knowledge of approximate camera calibration and camera speed are integrated with the estimation process using Bayesian approach. Iterative, coarse to fine, gradient based estimation is used to correct the motion parameters for vibrations and other inaccuracies in prior knowledge. Experiments with camera mounted on various types of mobile platforms demonstrate successful detection of moving persons and vehicles.

Keywords

Motion detection, Optical flow, Panoramic vision, Dynamic vision, Mobile robots, Intruder detection, Surveillance

1. INTRODUCTION AND MOTIVATION

Computer vision researchers have for long recognized the importance of visual surveillance related applications while pursuing some of the outstanding research issues in dynamic scene analysis, motion detection, feature extraction, pattern and activity analysis and biometric systems. Recent world events are demanding practical and robust deployment of video based solutions for a wide range of applications [6, 9, 25, 28]. Such wider acceptance of the need of the technology does not mean that these systems are indeed ready for deployment. There are many important and difficult research problems that are still outstanding. In this paper we present a research study focused on one of such challenging research problem, that of developing an autonomous system which can serve as a "Mobile Sentry" to perform the tasks that a person posted on a guard duty around a perimeter of a base gets assigned. The mobile sentry with video cameras should be able to detect "interesting" events, record and report the nature and location of the event in real-time for further processing by a human operator. This is indeed an ambitious goal and it requires resolution of several important problems from computer vision and intelligent robotics area. In this paper we are focused on the problem of how to detect and compensate for the ego motion of the mobile platform. One of the novel feature of our research is the omnidirectional video streams we use as the input to the vision system.

When the camera is stationary, background subtraction is often used to extract moving objects [16, 32]. However, when the camera is moving, the background also undergoes ego-motion, which should be compensated. To distinguish objects of interest from extraneous features on the ground, the ground is usually approximated by a planar surface, whose ego-motion is modelled using a projective transform [10, 24] or its linearized version. Using this model, the ego-motion of the ground can be compensated in order to separate the objects with independent motion or height. This approach has been widely used for object detection from moving platforms [5, 11, 24].

Omni-Directional Vision Sensors (ODVS) or omni-cameras which give a 360 degree field of view of the surroundings are very useful for applications in surveillance and robot navigation [3]. Motion estimation from moving ODVS cameras has recently been a topic of great interest. Rectilinear cameras usually have a smaller field of view, due to which the focus of expansion often lies outside the image, causing motion estimation to be sensitive to the camera orientation. Also, the motion field produced by translation along horizontal direction is similar to that due to rotation about vertical axis. As noted by Gluckman and Nayar [14], ODVS cameras avoid both these problems due to their wide field of view. They project the image motion on a spherical surface using Jacobians of transformations to determine ego-motion of a moving platform terms of translation and rotation of the camera. Vassalo et al. [34] use the spherical projection to determine ego-motion of a moving platform terms of translation and rotation of the camera. Experiments are performed using robotic platforms in indoor environment and the ego-

motion estimates are compared with those from odometry. Shakernia et al. [29] use the concept of back-projection flow, where the image motion is projected to a virtual curved surface in place of spherical surface and make the Jacobians of transformation simpler. Using this concept, they have adapted ego-motion algorithms designed to planar surface for use with ODVS sensors. Results using simulated image sequences show the basic feasibility of the approach.

In our own research, the emphasis is on robustness, efficiency, and applicability in outdoor environments encountered in surveillance and physical or base security. The main contribution of this paper is to perform detection of events such as independently moving persons and automobiles from ODVS image sequences, and apply it to video sequences obtained from a moving platform for surveillance applications.

2. EGO-MOTION COMPENSATION FRAMEWORK FOR ODVS VIDEO

Parametric motion estimation based on image gradients, also known as the "direct method" has been used for rectilinear cameras in for planar motion estimation, obstacle detection and motion segmentation [20, 23]. The advantage of the direct methods is that they can use motion information not only from corner-like features, but also edges, which are usually more numerous in an image. Here, the concept of direct method is extended for use with ODVS. This approach was also used for detecting surrounding vehicles from a moving car in [18]. An ODVS gives full 360 degree view of the surroundings, which reduces the motion ambiguities often present in the rectilinear cameras. However, the images undergo considerable distortion which should be accounted for during motion estimation.

The block diagram of the event detection system is shown in Figure 1. The initial estimates of the the ground plane motion parameters are obtained using the approximate knowledge about the camera calibration and speed. Using these parameters, one of the frames is warped towards another to compensate the motion of the ground plane. However, the motion of features having independent motion or height above the ground plane is not fully compensated. To detect these features, the normalized image difference between the two images is computed using temporal and spatial gradients. This suppresses the features on ground plane and enhances the interesting objects. Morphological and other post-processing is performed to further suppress the ground features due to any residual motion and get the positions of the objects. The detected objects are then tracked over frames to form events.

However, the calibration and speed of the camera may not be known accurately. Furthermore, the camera could vibrate during the motion. Due to this, the motion of the ground plane may not be fully compensated, leading to misses and false alarms. In order to improve the performance, motion parameters are iteratively corrected using the spatial and temporal gradients of the motion compensated images using optical flow constraint in coarse to fine framework. The motion information contained in these gradients is optimally combined with the prior knowledge of motion parameters using a Bayesian framework similar to [24]. Robust estimation is used to separate the ground plane features from other features. The following sections deal with the individual blocks described above, along with the appropriate formulation for ODVS.

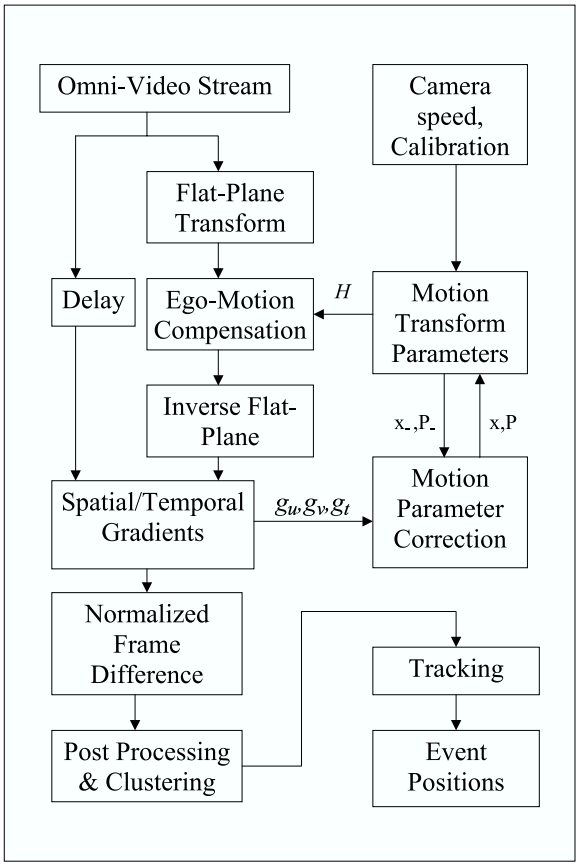


Figure 1: Event detection and recording system based on ego-motion compensation from a moving platform: Block Diagram

3. ODVS MOTION TRANSFORMATIONS

To compensate the motion of the omni-directional camera, the transformation due to ODVS should be combined with that due to motion. These transforms are discussed below:

3.1 Flat plane transformation

The ODVS used in this work consists of a hyperbolic mirror and a camera placed on its axis. It belongs to a class of cameras known as central panoramic catadioptric cameras [3]. These cameras have a single viewpoint that permits the image to be suitably transformed to obtain perspective views. Figure 2 (a) shows a photograph of an ODVS mirror. An image from a camera mounted with an ODVS mirror is shown in Figure 2 (b). It is seen that the camera covers a 360 degrees field of view around its center. However, the image it produces is distorted with straight lines transformed into curves. A flat plane transformation is applied to the image to produce a perspective view looking downwards as shown in Figure 2 (c), where the distortion is considerably reduced. Details of this transformation are discussed below.

(a)

(b)

(c)

Figure 2: (a) A typical image from the ODVS. (b) Transformation to a perspective plan view.

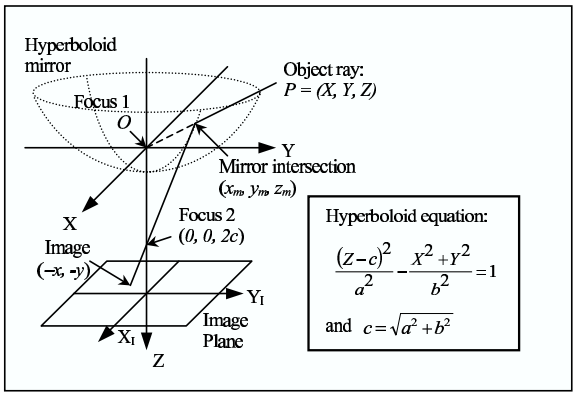


Figure 3: Omni-directional camera geometry

The geometry of a hyperbolic ODVS is shown in Figure 3. According to the mirror geometry, a the light ray from the object towards the viewpoint at the first focus O is reflected so that it passes through the second focus, where a conventional rectilinear camera is placed. The equation of the hyperboloid is given by:

$$\frac{(Z-c)^2}{a^2} - \frac{X^2+Y^2}{b^2} = 1 \tag{1}$$

where $c = \sqrt{a^2+b^2}$.

Let $P = (X, Y, Z)^t$ denote the homogenous coordinates of the perspective transform of any 3-D point λP on ray OP, where λ is the scale factor depending on the distance of the 3-D point from the origin. It can be shown [1, 19, 29] that the reflection in mirror gives the point $-p = (-x, -y)^t$ on the image plane of the camera using the flat-plane transform F:

$$F(P) = p = \begin{pmatrix} x \\ y \end{pmatrix} = \frac{q_1}{q_2 Z + q_3 \|P\|} \begin{pmatrix} X \\ Y \end{pmatrix} \tag{2}$$

where

$$q_1 = c^2 - a^2,\ q_2 = c^2 + a^2,\ q_3 = 2ac \tag{3}$$

$$\|P\| = \sqrt{X^2 + Y^2 + Z^2} \tag{4}$$

Note that the expression for image coordinates p is independent of the scale factor λ. The pixel coordinates $w = (u, v)^t$ are then obtained by using the calibration matrix K of the conventional camera composed of the focal lengths f_u, f_v, optical center coordinates $(u_0, v_0)^t$, and camera skew s.

$$\begin{pmatrix} w \\ 1 \end{pmatrix} = K \begin{pmatrix} p \\ 1 \end{pmatrix} \tag{5}$$

or

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{6}$$

This transform can be used to warp an omni image to a plan perspective view. To convert a perspective view back to omni view, the inverse flat-plane transform p can be used:

$$\begin{pmatrix} p \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{7}$$

$$F^{-1}(p) = P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} q_1 x \\ q_1 y \\ q_2 - q_3 \sqrt{x^2 + y^2 + 1} \end{pmatrix} \tag{8}$$

It should be noted that the transformation of omni to perspective view involves very different magnifications in different parts of the image. Due to this, the quality of the image deteriorates if the entire image is transformed at a time. Hence, it is desirable to perform motion estimation directly in the ODVS domain, but use the above transformations to map the locations to the perspective domain as required.

3.2 Planar motion transformation

To detect objects with motion or height, the motion of the ground is modeled using planar motion model [10, 22]. Let P_a and P_b denote the perspective transforms of a point on the ground plane in the homogenous coordinate systems corresponding to two positions A and B of the moving camera. These are related by:

$$\lambda_b P_b = \lambda_a R P_a + D = \lambda_a \left[R P_a + D/\lambda_a \right] \tag{9}$$

where R and D denote the rotation and translation between the camera positions, and λ_a, λ_b depend on the distance of the actual 3-D point. Let the ground plane satisfy the following equation at the camera position A:

$$\lambda_a K^t P_a = 1 \tag{10}$$

or

$$1/\lambda_a = K^t P_a \tag{11}$$

Substituting the value of $1/\lambda_a$ from equation (11) in equation (9), it is seen that P_a and P_b are related by a projective transform:

$$\lambda_b P_b = \lambda_a \left[R + D K^t \right] P_a = \lambda_a H P_a \tag{12}$$

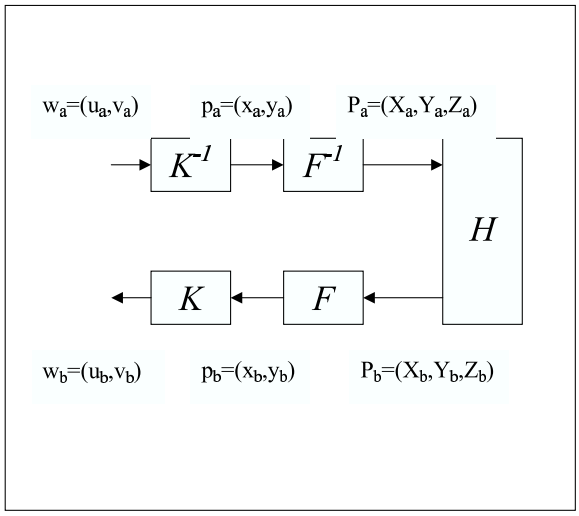


Figure 4: Transforming a pixel from omni image A to omni image B using (1) Inverse calibration matrix K^{-1}, (2) Inverse flat plane transform F^{-1} (3) Projective transform H for planar motion from A to B (4) Flat plane transform F (5) Calibration matrix K

or $P_b \equiv HP_a$ within a scale factor. This relation has been widely used to estimate planar motion for perspective cameras.

For performing motion compensation using omni-directional cameras, the above projective transform should be combined with the flat-plane transform as well as camera calibration matrix to warp every point in one image towards another. The complete transform for warping is shown in Figure 4.

4. PARAMETRIC MOTION ESTIMATION FOR ODVS

This section describes the main contribution of the paper. Direct methods based on image gradients have been applied for estimating the motion parameters for rectilinear cameras [4, 20]. Here, the direct method is generalized for ODVS cameras. Information from image gradients is combined with the a-priori known information about the camera motion and calibration in Bayesian framework to obtain optimal estimates of motion parameters for ego-motion compensation.

4.1 Use of optical flow constraint

Under favorable conditions, the spatial gradients (g_u, g_v), the temporal gradient g_t, and the residual image motion $(\Delta u, \Delta v)^t$ after current motion compensation satisfy the optical flow constraint [17].

$$g_u \Delta u + g_v \Delta v + g_t = 0 \tag{13}$$

However, there is only one equation between two unknowns for each point. Due to this, only the normal flow - i.e., flow in the direction of the gradient - can be determined using a single point. This is known as the aperture problem, illustrated in Figure 5 (a). To solve this problem, Lucas and Kanade [26] assumed that the image motion is approximately constant in a small window around every point. Using this constraint, more equations are obtained using the neighboring points, and the full optical flow can be estimated using least squares. Such an estimate is reliable near corner-like points where window has gradients in different directions. This is as seen in Figure 5 (b). This method has been used by Kanade, Lucas and Tomasi [30] to find and track corner-like features over an image. However, in case of ODVS the assumption of uniform optical flow needs to be modified due to the non-linear ODVS transform. Danilidis [7] has generalized the optical flow estimation to ODVS camera.

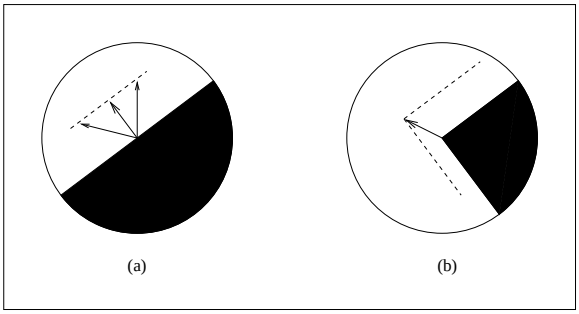


Figure 5: Aperture problem. (a) In the case of an edge, only the component of motion normal to the edge can be determined. (b) In the case of a corner, the aperture problem is avoided, and the motion can be uniquely determined.

However, this approach would use the motion information only at corner-like features. However, the edge features also have motion information. To use this information, the image gradients can be used directly to estimate the model parameters. This approach is known as direct method of motion estimation in literature and has been extensively used in obstacle detection using rectilinear cameras [4, 20]. Usually, a linearized version of projective transform is used:

$$\begin{aligned} \Delta u &= a_1 u + a_2 v + a_3 + a_7 u^2 + a_8 uv \\ \Delta v &= a_4 u + a_5 v + a_6 + a_7 uv + a_8 v^2 \end{aligned} \tag{14}$$

The expressions of image motion are substituted into the optical flow constraint in equation (13) to give:

$$g_u(a_1 u + a_2 v + a_3 + \ldots) + g_v(a_4 u + a_5 v + a_6 + \ldots) + g_t = 0 \tag{15}$$

This gives one equation for every point in 8 parameters, that can be solved using linear least squares. Since the quadratic parameters are more sensitive to noise, a 6-parameter affine model is also used.

4.2 Generalization for ODVS

To apply the motion estimation to ODVS camera, the non-linear flat-plane transform is used to go from omni to perspective domain and back. Since non-linearity has to be dealt with anyway, the projective transform H is used instead of a linear model so that large motions can be handled better. The motion parameters in the projective transform

are parameterized as:

$$\mathbf{h} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 \end{pmatrix}^t \quad (16)$$

with

$$\frac{H}{H_{33}} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \quad (17)$$

The optical flow constraint equation is satisfied only for small image displacements up to 1 or 2 pixels. To estimate larger motions, a coarse to fine pyramidal framework [21, 31] is used. In this framework, a multi-resolution Gaussian pyramid is constructed for adjacent images in the sequence. The motion parameters are first computed at the coarsest level, and the image points at the next finer level are warped using the computed motion parameters. The residual motion is computed at the finer level, and the process is repeated until the finest level. Even within each level, multiple iterations of warping and estimation can be performed.

Let $\bar{\mathbf{h}}$ be the actual value of the motion parameter vector, and $\hat{\mathbf{h}}$ be the current estimate. Using the current estimate, the second image B is warped towards the first image A to get the warped image B'. Then, the transformation between A and B' can be expressed approximately in terms of $\mathbf{\Delta h} = \bar{\mathbf{h}} - \hat{\mathbf{h}}$. Let $w_a = (u_a, v_a)^t$ be the projection of a point on the planar surface in image A. Then, the projection w_b of the same point in warped image B' is a function of w_a as well as $\mathbf{\Delta h}$, given using a composition of operations shown in Figure 4. The optical flow constraint between images A and B' is then given by:

$$\begin{pmatrix} g_u & g_v \end{pmatrix} [w_{b'} - w_a] = -g_t + \eta \quad (18)$$

where η accounts for the random noise in the temporal image gradient. For N points on the planar surface, the constraints can be expressed in a matrix form:

$$\mathbf{\Delta z} = \mathbf{c}(\mathbf{\Delta h}) + \mathbf{v} \quad (19)$$

where every row i of the equation represents the constraint for a single image point with

$$\begin{aligned} \mathbf{c}_i(\mathbf{\Delta h}) &= \begin{pmatrix} g_u & g_v \end{pmatrix}_i [w_{b'}(w_a; \Delta h) - w_a] \quad (20) \\ \mathbf{\Delta z}_i &= -(g_t)_i \,,\, \mathbf{v}_i = \eta_i \end{aligned}$$

Due to the flat plane and the projective transforms, the function $\mathbf{c}(\cdot)$ is a non-linear. Hence, state estimate $\hat{\mathbf{h}}$ and its covariance $\mathbf{P}$ are iteratively updated using the measurement update equations of the iterated extended Kalman filter [2], with $\mathbf{C}$ denoting the Jacobian matrix of $\mathbf{c}(\cdot)$.

$$\mathbf{P} \leftarrow \left[\gamma \mathbf{C}^t \mathbf{R}^{-1} \mathbf{C} + \mathbf{P}_-^{-1}\right]^{-1} \quad (21)$$

$$\hat{\mathbf{h}} \leftarrow \hat{\mathbf{h}} + \Delta\hat{\mathbf{h}} = \hat{\mathbf{h}} + \mathbf{P}\left[\gamma \mathbf{C}^t \mathbf{R}^{-1} \Delta\mathbf{z} - \mathbf{P}_-^{-1}(\hat{\mathbf{h}} - \mathbf{h}_-)\right] \quad (22)$$

where $\mathbf{R}$ is the covariance of the temporal gradient measurements, $\mathbf{h}_-$ is the prior value of state obtained from camera calibration and velocity, and $\mathbf{P}_-$ is the prior covariance. The matrix $\mathbf{R}$ is taken as a diagonal matrix to simplify calculations. However, this would mean assuming that the pixel gradients are independent, which may not really be the case since gradients are computed from multiple pixels. Hence, the factor $\gamma \leq 1$ is used to accommodate the interdependence of the gradient measurements.

To compute the Jacobian $\mathbf{C}$, each row $\mathbf{C}_i$ is expressed using chain rule:

$$\mathbf{C}_i = \left(\frac{\partial \mathbf{c}_i}{\partial \mathbf{h}}\right) = \left(\frac{\partial \mathbf{c}}{\partial w_b}\frac{\partial w_b}{\partial p_b}\frac{\partial p_b}{\partial P_b}\frac{\partial P_b}{\partial \mathbf{h}}\right)_i \quad (23)$$

where $P_b = (X_b, Y_b, Z_b)^t$, $p_b = (x_b, y_b)^t$ and $w_b = (u_b, v_b)^t$ are the coordinates of point i in in the mirror, image, and pixel coordinate systems for camera position B.

Differentiating equation (20) w.r.t. w_b gives:

$$\left(\frac{\partial \mathbf{c}}{\partial w_b}\right)_i = \begin{pmatrix} g_u & g_v \end{pmatrix}_i \quad (24)$$

The calibration equation (6) can be differentiated to obtain:

$$\left(\frac{\partial w_b}{\partial p_b}\right)_i = \begin{pmatrix} f_u & s \\ 0 & f_v \end{pmatrix} \quad (25)$$

The Jacobian of the flat plane transform is obtained by differentiating equation (2) at $P = P_b$ as:

$$\begin{aligned} \left(\frac{\partial p_b}{\partial P_b}\right)_i &= \begin{pmatrix} \frac{\partial x_b}{\partial X_b} & \frac{\partial x_b}{\partial Y_b} & \frac{\partial x_b}{\partial Z_b} \\ \frac{\partial y_b}{\partial X_b} & \frac{\partial y_b}{\partial Y_b} & \frac{\partial y_b}{\partial Z_b} \end{pmatrix} \quad (26) \\ &= \frac{1}{(q_2 Z_b + q_3 \|P_b\|)_i \, \|P_b\|_i} \\ &\quad \cdot \begin{pmatrix} q_3 x_b X_b - q_1 \|P_b\| & q_3 x_b Y_b & q_3 x_b Z_b \\ q_3 y_b X_b & q_3 y_b Y_b - q_1 \|P_b\| & q_3 y_b Z_b \end{pmatrix}_i \end{aligned}$$

Since the ODVS transforms giving p_a and p_b do not change if the homogenous coordinates P_a and P_b are changed by scale factor, we can scale the right hand side of equation (12) to give:

$$P_b = \frac{1}{H_{33}} H P_a = \begin{pmatrix} h_1 X_b + h_2 Y_b + h_3 Z_b \\ h_4 X_b + h_5 Y_b + h_6 Z_b \\ h_7 X_b + h_8 Y_b + Z_b \end{pmatrix} \quad (27)$$

Taking the Jacobian w.r.t. $\mathbf{h} = (h_1 \ldots h_8)$ gives:

$$\left(\frac{\partial P_b}{\partial \mathbf{h}}\right)_i = \begin{pmatrix} X_b & Y_b & Z_b & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_b & Y_b & Z_b & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & X_b & Y_b \end{pmatrix}_i \quad (28)$$

4.3 Outlier removal

The estimate given above is optimal only when all points really belong to the planar surface, and the underlying noise distributions are Gaussian. However, the estimation is highly sensitive to the presence of outliers, i.e. points not satisfying the ground motion model. These features should be separated using a robust method. To reduce the number of outliers, the region of interest of road is determined using calibration information, and the processing is done only in that region to avoid extraneous features. To detect outliers an approach similar to the data snooping approach discussed in [8] has been adapted for Bayesian estimation. In this approach, the error residual of each feature is compared with the expected residual covariance at every iteration, and the features are reclassified as inliers or outliers.

If a point $\mathbf{z}_i$ is not included in the estimation of $\hat{\mathbf{h}}$ – i.e. is currently classified as an outlier – then the covariance of its residual is:

$$\mathrm{Cov}\left[\mathbf{\Delta z}_i - \mathbf{C}_i \Delta\hat{\mathbf{h}}\right] \simeq \mathbf{R} + \mathbf{C}_i \mathbf{P} \mathbf{C}_i^t \quad (29)$$

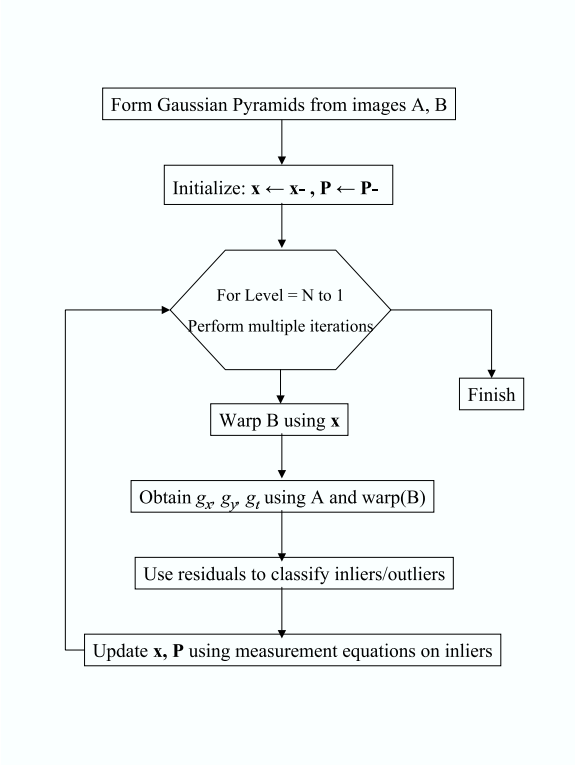


Figure 6: Hierarchical motion estimation algorithm.

However, if $\mathbf{z}_{in}$ is included in the estimation of $\hat{\mathbf{h}}$ – i.e. is currently classified as an inlier – then it can be shown that the covariance of its residual is given by:

$$\text{Cov}\left[\boldsymbol{\Delta}\mathbf{z}_i - \mathbf{C}_i\Delta\hat{\mathbf{h}}\right] \simeq \mathbf{R} - \mathbf{C}_i\mathbf{P}\mathbf{C}_i^t < \mathbf{R} \tag{30}$$

Hence, to classify in the next iteration, the Mahalanobis norm of the residual is compared with a threshold τ. For point currently classified as inlier the following condition is used:

$$\left[\boldsymbol{\Delta}\mathbf{z}_i - \mathbf{C}_i\Delta\hat{\mathbf{h}}\right]\left[\mathbf{R} + \mathbf{C}_i\mathbf{P}\mathbf{C}_i^t\right]^{-1}\left[\boldsymbol{\Delta}\mathbf{z}_i - \mathbf{C}_i\Delta\hat{\mathbf{h}}\right] < \tau \tag{31}$$

For point currently classified as inlier the covariance $\mathbf{R}$ is used in practice instead of $\mathbf{R} - \mathbf{C}_i\mathbf{P}\mathbf{C}_i^t$ in order to avoid non-positive definite covariance because of approximations due to non-linearities. This would somewhat increase the probability of classifying as an outlier instead of inlier, which is to be on safer side.

$$\left[\boldsymbol{\Delta}\mathbf{z}_i - \mathbf{C}_i\Delta\hat{\mathbf{h}}\right]\mathbf{R}^{-1}\left[\boldsymbol{\Delta}\mathbf{z}_i - \mathbf{C}_i\Delta\hat{\mathbf{h}}\right] < \tau \tag{32}$$

Note that this method is effective only when there is some prior knowledge about the motion parameters, otherwise the prior covariance $\mathbf{P}_-$ would become infinite. If there is no prior knowledge, robust estimators can be used as in [27]. The motion parameter estimation algorithm is shown in Figure 6.

5. DYNAMIC EVENT DETECTION AND TRACKING

After motion compensation, the features on the ground plane would be aligned between the two frames, whereas those due to stationary and moving objects would be misaligned. Image difference between the frames would therefore enhance the objects, and suppress the road features. However, the image difference depends on residual motion as well as the spatial gradients at that point. In highly textured regions, the image difference would be large even for small residual motion, and in less textured regions, the image difference would be small even for large residual motion. To compensate this effect, normalized frame difference [33] is used. This is given at each pixel by:

$$\frac{\sum g_t\sqrt{g_u^2 + g_v^2}}{k + \sum(g_u^2 + g_v^2)} \tag{33}$$

where g_u, g_v are spatial gradients, g_t is the temporal gradient. Constant k is used to suppress the effect of noise in highly uniform regions. The summation is performed over a $K \times K$ neighborhood of each pixel. In fact, the normalized difference is a smoothed version of the normal optical flow, and hence depends on the amount of motion near the point. Blobs corresponding to object features are obtained using morphological operations.

Nearby blobs are clustered into one, and the cluster centroids are tracked from frame to frame by an algorithm similar to [13]. For tracking, a list containing the frame number, unique ID, position, and velocity of each track is maintained. The list is empty in the beginning. The following steps are performed to associate the tracks with features:

- At each frame, associate each existing tracks with the nearest feature in a neighborhood window around the track position. Use Kalman filter [2] to update the track with the feature. If no feature is found in the neighborhood window, only time update is performed.
- For features not having a track in their neighborhood, create a new track out of the feature and update it in the next frame.
- To keep the number of tracks within bound, delete the weakest tracks when the number of tracks get too large.
- Merge the tracks that are very close to each other and have nearly the same velocity and are therefore assumed to be from same object.
- Display tracks which have sustained for a stipulated number of frames along with the track history.

For each track that survives over a minimum number of frames, the original ODVS image is used to generate a perspective view [19] of the event around the center of the bounding box.

6. EXPERIMENTAL VALIDATION AND RESULTS

A series of experimental trials were conducted to systematically evaluate the capabilities and performance of the mobile

Figure 7: An ODVS camera mounted on an electric cart for a mobile sentry experimental run.

sentry system for event detection and tracking. Three different types of ODVS mountings were utilized to examine the generality and functionalities of the mobile sentry system. The first trial involved a camera on an electric vehicle, the second was using a mobile robot, and the third involved a walking person with a helmet mounted ODVS. A related application applying a similar approach to an automobile mounted ODVS is also shown.

6.1 Camera on Electric Cart

The first experimental trial of the mobile sentry utilized the ODVS camera mounted on an electric cart as shown in Figure 7. The cart was driven on a campus road at speeds between 2 and 7 miles/hour (approx. 1 to 3 m/s). The approximate speed of the cart was determined using GPS and used as a-priori motion estimate. However, there were considerable vibrations of the camera. Using the parametric motion estimation process, the effect of these vibrations was suppressed. It was also observed that the ellipse corresponding to the entire FOV of the ODVS was oscillating, possibly due to relative vibrations between the camera and the mirror, or the automatic motion stabilization in the camera. These were suppressed by estimating the center of the FOV ellipse using a Hough transform, and translating it to a fixed position.

Figure 8 (a) shows an image from the ODVS video sequence. The estimated parametric motion is shown using red arrows. Figure 8 (b) shows the classification of points into inliers (gray), outliers (white), and unused (black) points. It should be noted that the outliers are usually identified when the edges are perpendicular to the motion. When an edge is parallel to the motion, aperture problem makes it difficult to identify it. The estimation is done only using the inlier points. Image with the normalized frame difference between the motion compensated frames is shown in Figure 8 (c). It is seen that the independently moving car and person stand out whereas the stationary features on ground are attenuated in spite of ego-motion. Figure 8 (d) shows the bounding boxes around the moving car and person after post-processing.

Table 1: Performance evaluation. The right column shows the ground truth number of relevant events in the image sequence. The other columns show the number and percentage of events detected by the system. The last three rows show the number of false alarms due to stationary objects and shadows. Ground truth is not relevant here.

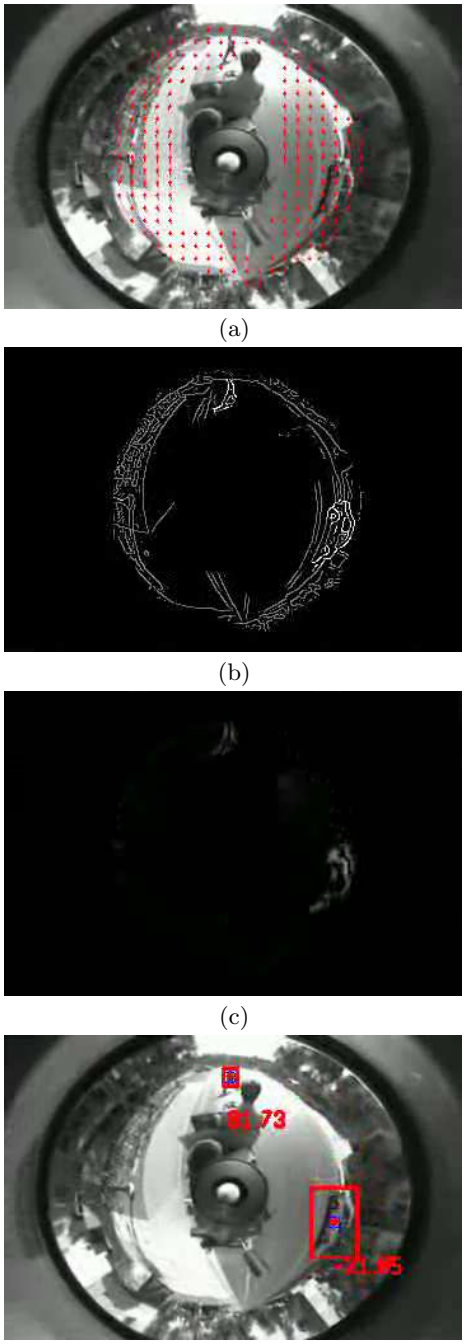
Minimum track length	15 frames	10 frames	Ground truth
Total Events	14 (74%)	17 (89%)	19
– Persons	9 (90%)	9 (90%)	10
– Vehicles	5 (55%)	8 (89%)	9
Total false alarms	3	4	N.A.
– Stationary objects	1	2	N.A.
– Shadows	2	2	N.A.

Since the algorithm uses a planar motion model, stationary objects above the ground induce motion parallax, and are detected if they are sufficiently close to the camera, and included in the region of interest. Figure 9 shows the detection of a stationary structure as well as a moving person. Only the parts within the region of interest are detected.

The centroids of the detected bounding boxes were tracked over time and the tracks that survived over 10 or more frames were identified. Typical snapshots from these tracks were taken, and the distortion due to ODVS was corrected to get the perspective view looking towards the track position as in [19]. Figure 10 show the snapshots from these tracks, detecting the events.

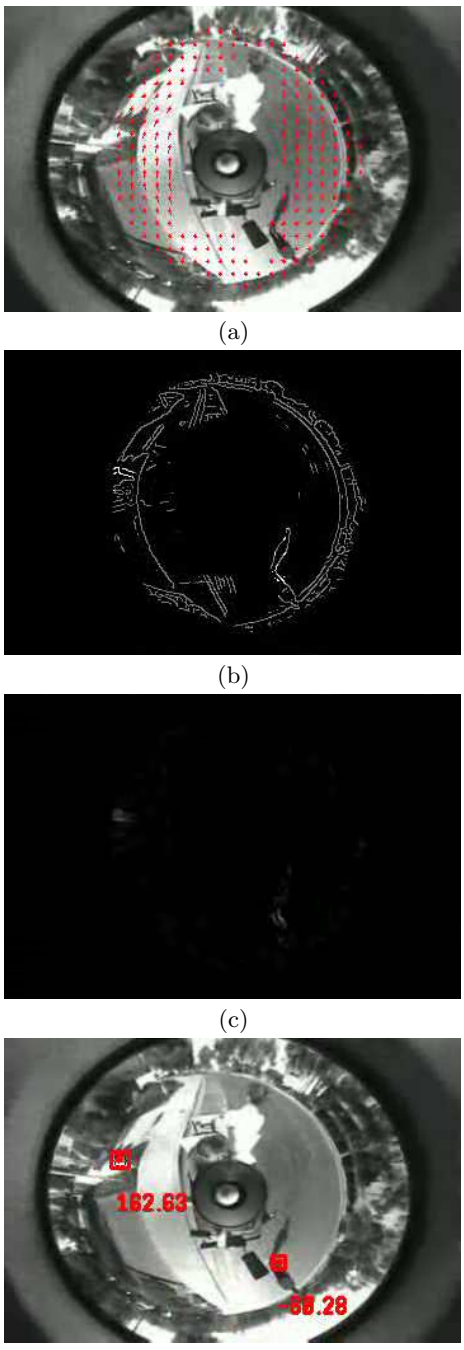
To evaluate the algorithm performance, the detection results were compared with ground truth obtained by manually observing the video sequence. The performance was compared for two different thresholds on the number of frames in which a track has to survive to be detected as an event. Table 1 shows the detection rate in terms of total number of events (ground truth), and the number of events actually detected. Note that stationary obstacles and shadows are classified as "false alarms", since they are currently not separated from independently moving objects. Lower threshold increases detection rate, but also increases false alarms. It was observed that an events corresponding to a moving person and cart were not detected at all due to the following reasons. The person and cart were quite far and especially the person had a small size in the image. Furthermore, the camera vehicle was turning, inducing considerable rotational ego-motion. Also, the objects were near the boundary of the region of interest that was analyzed. An image of this person is shown in Figure 11.

Attributes such as the time, duration and position of the events were extracted. The camera position at the event time was extracted from the onboard GPS. Assuming that the point nearest to the camera lies on the ground, the event location with reference to the camera could be computed. These were added to the camera position coordinates to obtain the event position. the approximate positions of the camera as well as the actual event were mapped as show in in Figure 12. Table 2 summarizes some of the events and their attributes.



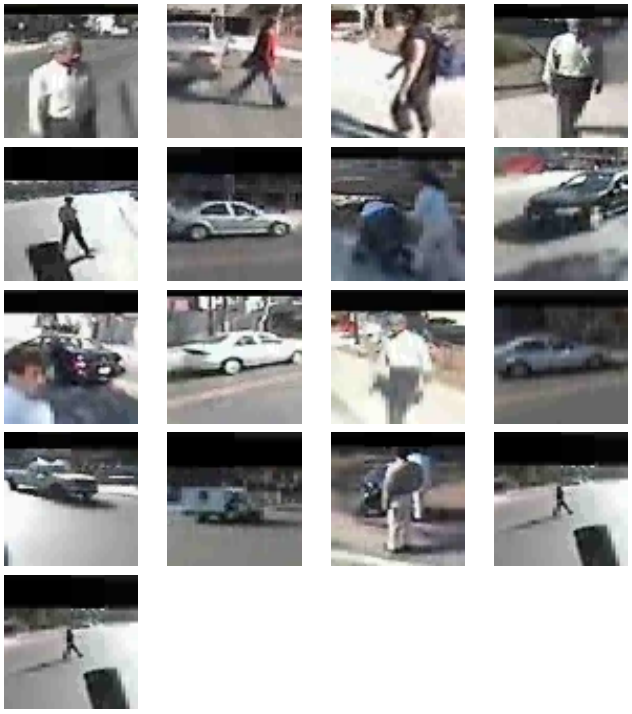
(a) (b) (c) (d)

Figure 8: Detection of moving objects from an ODVS mounted on an electric cart: (a) Estimated parametric motion of ground plane. Parts of the image corresponding to the cart as well as distant objects are excluded from motion estimation process. (b) Features used for estimation. Gray features are inliers, and white features are outliers. (c) Motion compensated difference image (d) Post-processed image showing detection of the moving car and person. The angle made with X-axis in degrees is also shown.



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

Figure 9: Detection of stationary structure in addition to a moving person. Note that only the part of the structure within the region of interest is detected.



(a)

(b)

Figure 10: Captured events with their IDs (a) Detected persons and vehicles. (b) Shadows and stationary obstacles currently considered false alarms.

Figure 11: Original image corresponding to a missed events. The moving person and vehicle were far from the camera and near the boundary of region of interest. Also, the camera vehicle was taking a turn, inducing considerable rotational ego-motion in the image.

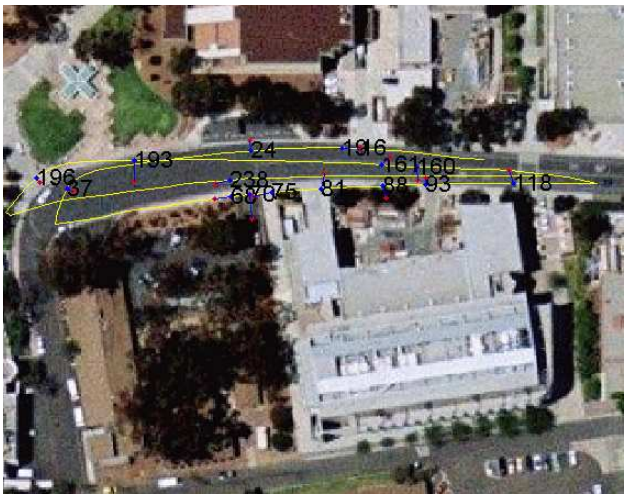
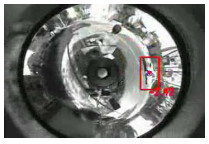


Figure 12: Map showing the position of the events in red and the camera position at that time in blue. The ego-vehicle track is marked as yellow line. The event IDs are labeled in black.

Table 2: Summarization of event attributes. The left image shows the original ODVS image at the time of the event, middle image shows the output of detection algorithm, and the right image shows the snapshot of the event corrected for ODVS distortion



```
Event ID: 65
Event time (snapshot):  16:01:08.3
Event duration [seconds]:  3.6
Event position [meters]:  (7.1, -6.6)
Camera position [meters]:  (7.8, -5.3)
```


```
Event ID: 109
Event time (snapshot):  16:01:40.0
Event duration [seconds]:  1.9
Event position [meters]:  (53.1, 1.8)
Camera position [meters]:  (56.0, -3.1)
```

Figure 13: Mobile Interactive Avatar: Semi-autonomous robotic system used for evaluating mobile sentry.

6.2 Camera on a Mobile Robot

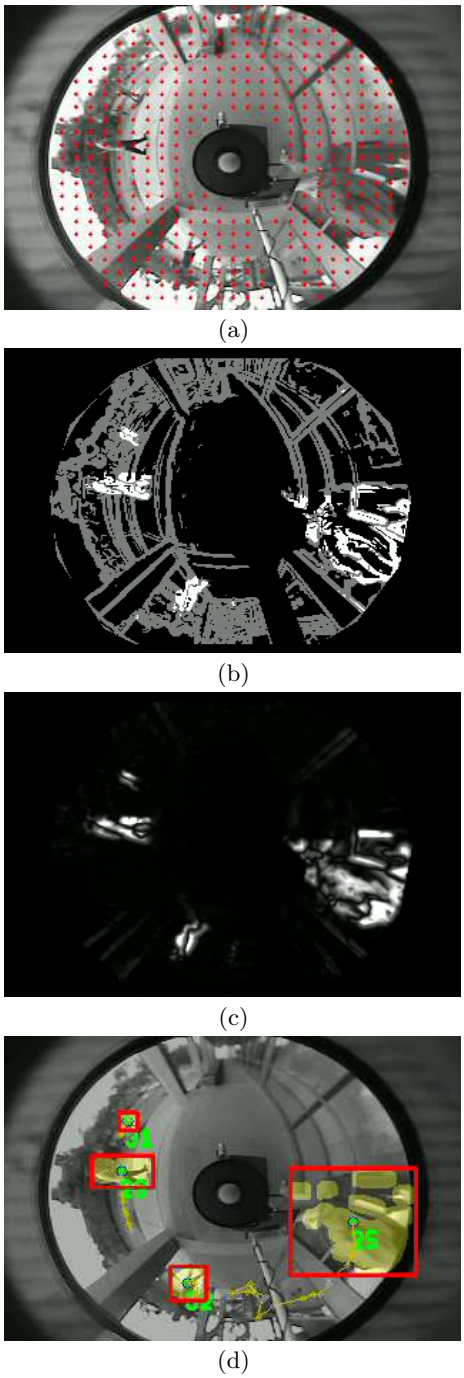
The second experimental configuration was a robotic platform designed in our laboratory. This platform is called Mobile Interactive Avatar (MIA) [15] in which cameras, displays can be mounted on semi-autonomous robot as shown in Figure 13 to interact with people at a distance. The robot was driven around the corridor of our building with people walking around it. Figure 14 shows the detection of moving persons in one of the frames. Snapshots of detected people shown in Figure 15. However, it was noted that the speed of the robot was much smaller than that of people, which would mean that simpler methods could also yield good results in this scenario. Also, the height of the robot was small, and people's faces could not be effectively captured.

6.3 Helmet mounted camera

The third experimental study for mobile sentry involved a person walking with an ODVS mounted on a helmet as shown in Figure 16. In this configuration, the camera height was approximately 2 meters, which enabled easy capture of people's faces. The speed of the person with helmet was comparable to that of other people. There was also a considerable camera rotation due to head and body movement, which helped to test the algorithm in presence of large rotational ego-motions. Figure 17 shows the detection of a moving person in one of the frames. To reduce estimation errors, the region of interest for motion analysis was truncated to remove the camera's own image, as well as the objects above the horizon. It is seen that there is significant rotational motion between two frames. In spite of this motion, the moving person is separated from background features such as the lines on the ground. However, if the rotational motion is too large, the detection often deteriorates and tracks get split into parts. Some of the snapshots of detected people shown are in Figure 18.

6.4 Vehicle mounted ODVS

In a related application [18], the event detection approach was applied to an omni camera mounted on an automobile. The actual vehicle speed, obtained from CAN bus was used for initial motion estimate. Algorithm similar to the above described one was used to detect vehicles on both sides of the car were detected as shown in Figure 19.



(a)

(b)

(c)

(d)

Figure 14: Detection of moving persons in an image sequence from a mobile robot. (a) Estimated parametric motion of ground plane. The part of the image in center, which images the camera itself is not used for estimation. (b) Features used for estimation. Gray features are inliers, and white features are outliers. (c) Normalized image difference after motion compensation. The moving person is detected but the lines on the ground are not detected. (d) Postprocessing and tracking output. The track of the detected person with the ID is shown with yellow line.

(b)

Figure 15: Some of the interesting events captured by the mobile robot.

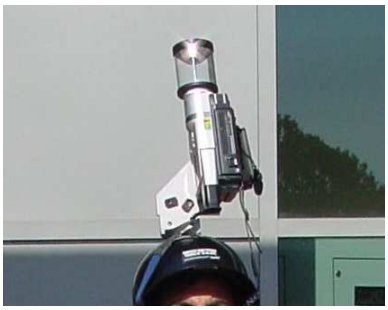
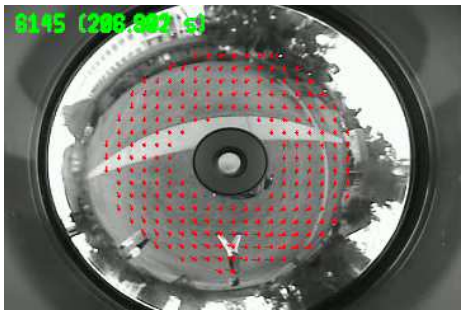


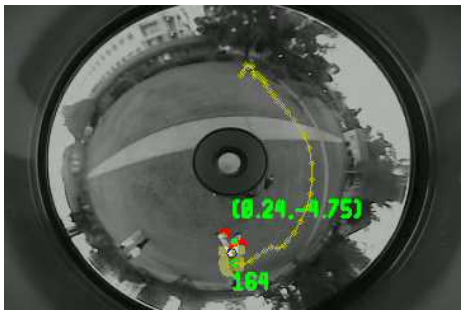
Figure 16: ODVS camera mounted on a helmet. This configuration enables acquisition of snapshots of surrounding people including their faces. However, there is considerable camera rotation due to movement of head and body which should be compensated.



(a)

(b)

(c)



(d)

Figure 17: Detection of a moving person in an image sequence from a helmet mounted camera. (a) Estimated parametric motion of ground plane. There is significant rotational motion that is estimated by the algorithm. (b) Features used for estimation. Gray features are inliers, and white features are outliers. The parts of the image in center, above horizon, and those having small image gradients are not used in estimation. (c) Normalized image difference after motion compensation. The moving person is detected but the lines on the ground are not detected. (d) Postprocessing and tracking output. The track of the detected person with the ID is shown with yellow line. The position coordinates are computed by assuming that the point on the blob nearest to the camera is on ground.

Figure 18: Some of the detected events from the helmet mounted camera.

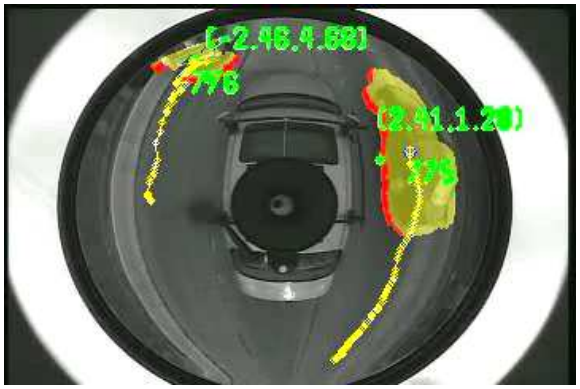


Figure 19: Detection of moving vehicles in an image sequence using omnidirectional camera mounted on a moving car. The track history of the vehicle over a number of frames is marked. The track id and the coordinates in road plane are also shown.

7. SUMMARY AND CONCLUDING REMARKS

This paper described an approach for event detection using ego-motion compensation from mobile omni-directional (ODVS) cameras. It applied the concept of direct motion estimation using image gradients to ODVS cameras. The motion of the ground was modeled as planar motion, and the features not obeying the motion model were separated as outliers. An iterative estimation framework was used for optimally fusing the motion information in image gradients with a-priori known information about the camera motion and calibration. Coarse to fine motion estimation was used and the motion between the frames was compensated at each iteration. A scheme based on data snooping was used to remove outliers. Experiments were performed by obtaining image sequences from various types of mobile platforms, and detecting events such as moving persons and automobiles.

The motion estimation approach described in the paper can be extended to use multiple motion models for estimation. In case of scenes with multiple stationary planar surfaces, the surfaces have the same parameters for rotation and translation, but different planar normals [12]. Hence, the ego-motion can be parameterized directly in terms of the linear and angular velocity of the camera, and the plane normals of each planar surface. An iterative estimation procedure which estimates each planar surface separately, but uses the estimates it obtains for rotation and translation as starting points for estimating other planar surfaces could make the process more robust to outliers. For example, if the scene consists of features far away from the camera, their ego-motion could be considered almost pure rotation, having only three degrees of freedom. These features could be used to estimate the approximate rotation [33]. This rotation could be used as an initial estimate for the parts of the scene containing ground plane in order to determine the full planar motion. This procedure could be combined with a robust motion segmentation method such as [27] to automatically separate multiple planar surfaces.

8. ACKNOWLEDGEMENTS

We are thankful for the grant awarded by the Technical Support Working Group (TSWG) of the US Department of Defense which provided the primary sponsorship of the reported research. We also thank our colleagues from the UCSD Computer Vision and Research Laboratory for their contributions and support.

9. REFERENCES

[1] O. Achler and M. M. Trivedi. Real-time traffic flow analysis using omnidirectional video network and flatplane transformation. In *World Congress on Intelligent Transportation Systems*, Chicago, IL, 2002.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. John Wiley and Sons, 2001.

[3] R. Benosman and S. B. Kang. *Panoramic Vision: Sensors, Theory, and Applications*. Springer, 2001.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[5] S. Carlsson and J. O. Eklundh. Object detection using model-based prediction and motion parallax. In *European Conference on Computer Vision*, pages 297–306, April 1990.

[6] E. Chang and Y.-F. Wang, editors. *First ACM International Workshop on Video Surveillance*, Berkeley, CA, November 2003.

[7] K. Daniilidis, A. Makadia, and T. Bulow. Image processing in catadioptric planes: Spatiotemporal derivatives and optical flow computation. In *IEEE Workshop on Omnidirectional Vision*, pages 3–12, June 2002.

[8] G. Danuser and M. Stricker. Parametric model fitting: From inlier characterization to outlier detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):263–280, March 1998.

[9] K. L. Dean. Smartcams take aim at terrorists. Wired News, June 2003. `http://cvrr.ucsd.edu/press/articles/Wired News Smartcams.html`.

[10] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[11] T. Gandhi, S. Devadiga, R. Kasturi, and O. Camps. Detection of obstacles using ego-motion compensation and tracking of significant features. *Image Vision and Computing*, 18(10):805–815, 2000.

[12] T. Gandhi and R. Kasturi. Application of planar motion segmentation for scene text extraction. In *Int. Conf. on Pattern Recognition*, volume 1, pages 445–449, 2000.

[13] T. Gandhi, M.-T. Yang, R. Kasturi, O. Camps, L. Coraor, and J. McCandless. Detection of obstacles in the flight path of an aircraft. *IEEE Transactions on Aerospace and Electronic Systems*, 39(1):176–191, January 2003.

[14] J. Gluckman and S. Nayar. Ego-motion and omnidirectional cameras. In *Proceedings of the International Conference on Computer Vision*, pages 999–1005, 1998.

[15] T. B. Hall and M. M. Trivedi. A novel interactivity environment for integrated intelligent transportation and telematic systems. In *5th International IEEE Conference on Intelligent Transportation Systems*, Singapore, September 2002.

[16] Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):809–830, Aug. 2000.

[17] B. Horn and B. Schunck. Determining optical flow. In *DARPA81*, pages 144–156, 1981.

[18] K. Huang, M. M. Trivedi, and T. Gandhi. Driver's view and vehicle surround estimation using omnidirectional video stream. In *IEEE Conference on Intelligent Vehicles*, Columbus, OH, June 2003.

[19] K. C. Huang and M. M. Trivedi. Video arrays for real-time tracking of persons, head and face in an intelligent room. *Machine Vision and Applications*, 14(2):103–111, 2003.

[20] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998.

[21] B. Jähne, H. Haußecker, and P. Geißler. *Handbook of Computer Vision and Applications*, volume 2, chapter 14, pages 397–422. Academic Press, San Diego, CA, 1999.

[22] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, Oxford, 1993.

[23] Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 390–397, June 2003.

[24] W. Kruger. Robust real time ground plane motion compensation from a moving vehicle. *Machine Vision and Applications*, 11:203–212, 1999.

[25] S. Lawson. Yes, you are being watched. PC World, December 27, 2002. `http://www.pcworld.com/news/article/0,aid,108121,00.asp`.

[26] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[27] J. M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66:143–145, 1998.

[28] D. Ramsey. Researchers work with public agencies to enhance super bowl security, February 15 2003. `http://www.calit2.net/news/2003/2-4_superbowl.html`.

[29] O. Shakernia, R. Vidal, and S. Sastry. Omnidirectional egomotion estimation from back-projection flow. In *IEEE Workshop on Omnidirectional Vision (in conjunction with CVPR)*, June 2003.

[30] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[31] E. P. Simoncelli. Coarse-to-fine estimation of visual motion. In *Proc. Eighth Workshop on Image and Multidimensional Signal Processing*, pages 128–129, Cannes, France, 1993.

[32] C. Stauffer and W. E. L. Grimson. Adaptive background mixture model for real-time tracking. In *Proceedings of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[33] E. Trucco and A. Verri. *Computer vision and applications: A guide for students and practitioners*. Prentice Hall, March 1998.

[34] R. F. Vassallo, J. Santos-Victor, and H. J. Schneebeli. A general approach for egomotion estimation with omnidirectional images. In *IEEE Workshop on Omnidirectional Vision*, pages 97–103, June 2002.