

# Real-time Detection of Moving Objects from Moving Vehicles using Dense Stereo and Optical Flow

Ashit Talukder and Larry Matthies

Jet Propulsion Laboratory, 4800 Oak Grove Drive, MS 300-123, Pasadena, CA 91109

Email: [Ashit.Talukder@jpl.nasa.gov](mailto:Ashit.Talukder@jpl.nasa.gov), [Larry.Matthies@jpl.nasa.gov](mailto:Larry.Matthies@jpl.nasa.gov)

## Abstract

*Dynamic scene perception is very important for autonomous vehicles operating around other moving vehicles and humans. Most work on real-time object tracking from moving platforms has used sparse features or assumed flat scene structures. We have recently extended a real-time, dense stereo system to include real-time, dense optical flow, enabling more comprehensive dynamic scene analysis. We describe algorithms to robustly estimate 6-DOF robot egomotion in the presence of moving objects using dense flow and dense stereo. We then use dense stereo and egomotion estimates to identify other moving objects while the robot itself is moving. We present results showing accurate egomotion estimation and detection of moving people and vehicles under general 6-DOF motion of the robot and independently moving objects. The system runs at 18.3 Hz on a 1.4 GHz Pentium M laptop, computing 160x120 disparity maps and optical flow fields, egomotion, and moving object segmentation. We believe this is a significant step toward general unconstrained dynamic scene analysis for mobile robots, as well as for improved position estimation where GPS is unavailable.*

**Keywords:** Dynamic scene analysis, egomotion, moving object detection, object tracking, optical flow, visual odometry.

## 1. Introduction

Detection and tracking of moving objects in imagery is important in many applications, including autonomous navigation of unmanned vehicles, intelligent passenger vehicles, human-robot interaction, video surveillance, and a variety of other video compression, editing, and related tasks. Our focus is primarily on intelligent manned or unmanned vehicle applications. As such, we seek motion analysis algorithms that can operate in real-time (i.e. many frames/sec) with CPU and memory resources that are practical for mobile systems, including man-portable robotic vehicles.

The literature on dynamic scene analysis is vast; here we can only cite a few representatives of key threads of related work. A recent review of vision technology for intelligent vehicles [1] has noted that while several solutions have been proposed to the problem of dynamic object detection in real-time, a clear feasible system has not emerged so far. This is primarily due to the fact that

algorithms with superior performance require huge amounts of computing resources, whereas current real-time implementations make restrictive assumptions about object or robot motion or scene structure. Dynamic scene research from moving vehicles primarily is divided into background subtraction methods, sparse feature tracking methods, background modelling techniques, and robot motion models. We discuss these solutions briefly below.

A great deal of moving object work has been done with stationary cameras, particularly for surveillance. Such work often begins with background subtraction for moving object detection using grayscale/color images [2, 3] and stereo disparity background models [4] for people tracking. This clearly is not sufficient by itself when the camera is also in motion. Solutions with adaptive background update algorithms have been built to handle illumination variations, background changes over time, and/or slow camera motion [4]. However, they fail when the camera motion is fast and the scene is complex.

Another line of work detects and tracks moving objects on the ground from moving cameras on relatively high altitude aircraft, where the stationary background surface can be modelled as essentially planar and subtracted out by affine registration of the entire background [5]. This is not adequate where the moving camera is on a ground vehicle and the environment has significant variations in depth. Ground-based vision systems have also used affine models to detect cars on flat roads [6]. However, such assumptions fail when the background is not flat or has complex 3D structure.

A family of solutions in dynamic object detection from mobile platforms assumes restricted camera/robot motion in the attempt to simplify the egomotion estimation and object detection problem. In [7], the author assumes purely forward camera motion and detects moving objects as outliers that violate this motion model. Effects of rotational egomotion are cancelled using a-priori trained rotational motion templates. The motion model is less restrictive in [8], where 3 DOF robot motion is estimated.

For moving cameras at ground level, motion analysis has often been done by tracking point features. This includes work that assumes moving camera(s) and a static scene, doing either monocular structure from motion [9] or stereo structure and egomotion [10, 11]. Multi-body structure from motion has been addressed by using factorization to batch process points tracked in long, monocular image sequences [12]. Kalman filter-based algorithms have also been used to locate moving cars in

stereo image sequences acquired from a moving car using sparse feature points [13]. These approaches have achieved considerable success, though the recovered world model is necessarily limited by the sparseness of the point features that are tracked. Monocular methods, such as [9] compute motion estimates only up to a scale factor. Batch algorithms [12], while potentially robust, are slow and not suited for real-time systems. Prior Kalman filter-based tracking [13] solutions assume translational motion and was designed to find only one moving object in the scene. Additionally, it only shows segmentation results on 6-10 manually selected features; the computational limitations of the technique on dense feature sets is not discussed.

Our solution combines dense stereo with dense optical flow and yields an estimate of object/background motion at every pixel; this increases the likelihood of detecting small / distant objects or those with low texture where feature selection schemes might fail. Considerable work was done in the 1980s on multi-body structure and motion estimation from monocular and binocular dense optical flow fields [14-16], but with no aspiration to real-time performance. Techniques that employ dense optical flow include dominant motion detection schemes, but such methods fail when scene structure is complex. In [17], egomotion information was obtained from an odometer and moving objects were located by clustering the dense optical flow field into regions of similar motion; however, the same limitations for complex scene structure apply here since 3D depth information is not used. Depth from stereo could overcome these limitations. Waxman's seminal paper [15] derived the relation between stereo and dense optical flow for planar scene patches. Our work builds on the basic principles derived in that paper, but does not need the planar scene patch assumption, and extends it to real-time robust dynamic scene analysis by combining dense stereo and optical flow.

Several research groups and companies now have the ability to produce dense stereo disparity fields at or near video rates with compact computer systems, using variants on area-based matching algorithms. We have adapted such algorithms to compute optical flow in real-time [18]. In this paper, we extend this line of work to a more comprehensive approach to moving object detection on-the-move by using stereo disparity fields and optical flow fields to estimate egomotion, and using predicted and observed flow and disparity to detect moving objects. The novelty of our work is in the fact that our solution enables detection of moving objects in real-time without any constraints on object or robot motion or on the scene structure, in contrast with prior approaches that constrain camera/robot motion [7, 8], make flat scene assumptions [5, 6], or work only when camera motion is slow or non-existent [4]. Section 2 outlines our approach in greater detail and reviews our results on fast optical flow field estimation. Section 3 describes how we use disparity and flow to estimate egomotion. Section 4 gives the extension

of the formulation to detect moving objects. Section 5 presents quantitative and qualitative experimental results for egomotion estimation and shows results for moving object detection on-the-move with several image sequences. This whole process runs at 18.3Hz (54.6 ms/frame) on a Pentium M 1.4 GHz machine with 750 MB RAM, using 160x120 disparity and flow fields. We draw conclusions in Section 6.

## 2. Overview of Approach

The crux issue in moving object detection on-the-move is to distinguish the apparent motion of the static background from that of the moving objects. If we have a depth map, which current stereo systems provide, and if we know the motion of the cameras, then in principle we can predict the optical flow and the change in depth that the camera motion will induce for the background, difference that from the measured optical flow and change in depth, and flag large non-zero areas as potential moving objects. This reduces the crux issue to the

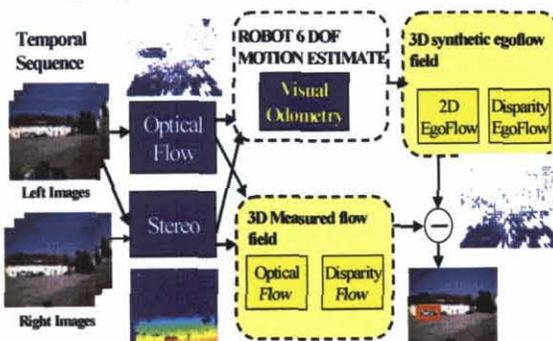


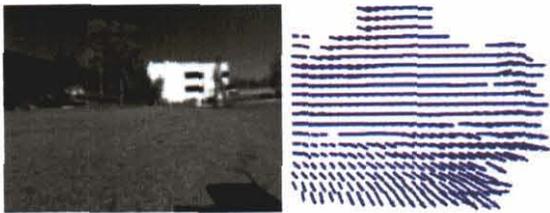
Figure 1: Algorithmic architecture for moving object detection on-the-move

following question: can we estimate the camera motion, depth map, and optical flow field well enough and fast enough to make this scheme practical for intelligent vehicles? We show that the answer to this is yes. Note that the scheme does not depend on having fully dense depth or optical flow data, because it can work with however many pixels have such data. Figure 1 illustrates this approach. In the figure, we refer to the optical flow and change in depth induced just by the camera motion as *egoflow*, to distinguish it from the measured flow and depth change that could include separate object motion. The *3D egoflow field* refers to image plane (x,y) components of flow plus the associated temporal disparity change, not to 3D coordinates in the X-Y-Z sense.

Solid-state (MEMS-based) inertial measurement units (IMUs) are becoming small, cheap, and accurate enough, as well as essential enough, that virtually all robotic vehicles (and in the future all passenger vehicles) can be assumed to have an IMU. This will provide information about camera motion, as will other state sensors, like wheel encoders. However, for a variety of reasons this is not sufficient, because problems like wheel slip, IMU saturation, and calibration errors between the IMU and the cameras may cause inaccurate motion estimation.

Thus, it is essential to also estimate the camera motion directly from the imagery; ultimately, this will be fused with other state sensors to produce a more accurate and reliable joint estimate of camera/vehicle motion. This will augment the egomotion box shown in figure 1. Several groups have reported stereo egomotion estimation (also called visual odometry) based on tracking point features that is reliable and accurate to 1-3% of distance travelled and a few degrees of heading over hundreds of frames [10, 11]. Thus, the basic feasibility of visual egomotion estimation is established, and the remaining issues in that department are engineering the speed, accuracy, and reliability of the competing approaches to the problem including reliability with other moving objects in the image.

Both egomotion estimation and moving object detection require some form of low-level image motion estimation. Where speed is an issue, this has generally been done by extracting and tracking feature points. For long-term generality, we believe it is desirable to extend this to dense optical flow fields. Moreover, it turns out that current real-time, dense stereo algorithms are readily adaptable to computing useable optical flow estimates quite quickly. Conceptually, instead of searching a 1-D disparity range that lies completely within one scanline, the disparity search is broken into segments on adjacent scanlines in a rectangle that defines the maximum tolerable 2-D optical flow. We suppress the details of the implementation for brevity, but the algorithmic change in the correlation stage from doing stereo is quite small. Our implementation is based on the SAD similarity measure applied to bandpass or highpass filtered imagery, as is relatively common in real-time stereo. Subpixel flow estimates are essential; for speed, we implement this by fitting 1-D parabolas separately to the horizontal and vertical components of flow. We currently search a 15x15 pixel window centered on the source pixel; for reasons of speed, we encode the flow components in one byte, so subpixel resolution is quantized to 1/16 of a pixel. Bad matches do occur; currently, we filter those just by removing small disconnected blobs in the estimated flow



**Figure 2:** Typical optical flow result; see text for discussion.

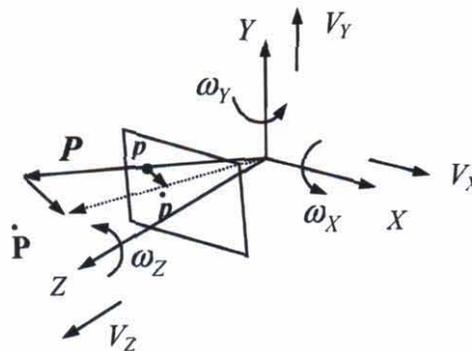
field. Some additional refinement of that is still desirable to remove wild estimates that are connected to areas of good estimates. On a 1.4 GHz Pentium M laptop, using 11x11 SAD templates and a 15x15 search, we can compute 320x240 flow fields (not including image decimation and rectification) at 6 Hz; the speed on the same machines for 160x120 flow fields is 33.2 Hz.

Figure 2 shows a sample optical flow result, computed with imagery from a robot with cameras about 1 foot above the ground, driving in a curving arc to the left on a paved lot with a person moving in the field of view. Observe that the flow field is quite dense and is in qualitative agreement with the known camera motion. The moving person causes some discordant flow in the upper left portion of the image. Some wild points are present near the shadow in the lower right; these tend to be filtered in subsequent stage of the process shown in figure 1, but do represent an issue for further work. Note that dense flow is obtained on the relatively low contrast pavement. Our emphasis in this whole approach has been to produce useable flow fields in a fast implementation; thus, the large literature on more sophisticated flow algorithms [19] is germane but requires much more computation. Even with our approach, optical flow is still much slower than stereo, because a 15x15 search region is 225 disparities, whereas for 320x240 stereo it is typical to compute on the order of 40 disparities.

### 3. Egomotion Estimation

The terms egomotion estimation and visual odometry are both in common use for essentially the same function; in this paper we use the term egomotion. Although various prior authors have assumed restricted robot motions and/or restricted scene geometries to simplify or constrain this problem, such as 3 degree-of-freedom (DOF) motion on a planar terrain, we make no such assumptions. This is warranted by the fact that implementations already exist that use stereo point tracking to obtain accurate 6 DOF egomotion [10, 11]. Our goal here is to formulate the motion estimator in terms of our observables (disparity and optical flow) in such a way as to obtain fast, accurate estimates in the presence of egomotion and independent scene motion.

Since we have disparity and optical flow, we formulate the motion estimator with the classic equation relating



**Figure 3:** 3D motion and projection to 2D image.

instantaneous velocities to range and optical flow [20]. This is valid for the small motions compatible with the tracking range of our optical flow algorithm. Coordinate frame conventions and notation for this are illustrated in figure 3. The origin of the coordinate system is at the projection center of the camera, with X left, Y up, and Z

forward for a right-handed system.  $(x,y)$  are image coordinates,  $d$  is disparity,  $\delta$  is disparity change between frames, and  $f$  and  $b$  are focal length and stereo baseline, respectively.

### 3.1. Optical Flow and 3D motion

The equation for the general motion of a point  $P = (X,Y,Z)$  in 3D space about the center of the camera projection point is given by [20] :

$$dP/dt = -(V + \Omega \times P) \quad (1)$$

where  $\Omega = [\omega_x, \omega_y, \omega_z]$  is the rotational velocity and  $V = [V_x, V_y, V_z]$  is the translational velocity. For perspective projection cameras,

$$x = \frac{fX}{Z}, y = \frac{fY}{Z} \quad (2)$$

Optical flow  $(u,v)$  is the time derivative of this:

$$u = \frac{1}{Z} \left( f \frac{dX}{dt} - x \frac{dZ}{dt} \right), \quad (3)$$

$$v = \frac{1}{Z} \left( f \frac{dY}{dt} - y \frac{dZ}{dt} \right).$$

Equations (1) to (3) yield the relation between 2D optical flow, image coordinates,  $Z$ , and 3D motion:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \frac{1}{f} \begin{bmatrix} xy & -(f^2 + x^2) & fy \\ (f^2 + y^2) & -xy & -fx \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (4)$$

Dividing out  $f$  one more time normalizes  $x,y,u,v$  and makes  $f$  not explicitly visible in this equation. Since  $u,v$  and  $d = fb/Z$  are measured, this is a linear least squares problem for estimating the motion parameters.

Most current approaches to stereo-based egomotion estimation formulate the problem by first triangulating feature points to get 3D coordinates, then solving the resulting 3D to 3D pose estimation problem. To obtain satisfactory results, this requires including a 3D Gaussian uncertainty model for the 3D coordinates, which ultimately requires an iterative solution. It also introduces a small bias, because the uncertainty on the 3D coordinates is not truly Gaussian, but has a slight asymmetric distribution in depth. Also, solving for motion as a 3D to 3D problem requires not using points with zero disparity; however, these points can contribute significantly to the attitude estimate. Such points can be used with equation (4). Finally, 3D to 3D formulations require full floating point implementations, whereas keeping the problem in image and disparity space allows more use of shorter word sizes and fixed point arithmetic, hence enables a faster implementation. Thus, for several reasons, the image space formulation of equation (4) has advantages over the more typical 3D to 3D formulation.

### 3.2. Robust Solution for Dynamic Scenes

Independent scene motion, as well as inevitable disparity and optical flow errors, require a robust solution to (4) that copes with outliers. In a system context, an IMU and other state sensors can also contribute to solving this problem; however, we do not explore such details here. The dense disparity and optical flow fields do yield a highly over-determined system of equations from which the egomotion can be estimated accurately, even in the presence of moving objects.

There is a large and well known literature on robust estimation techniques, which we do not survey here. Since our immediate goal is to do proof-of-concept testing of this approach in situations where independent scene motion and other outliers affect a modest fraction of the pixels, it is sufficient at present to use an iterative least mean-square error (LMSE) estimation technique where we estimate the motion parameters using all data points, reject outlier data points based on the estimated motion parameters, and re-estimate the motion parameters with LMSE using only the inlier points. The initial LMSE estimates using all data points are first computed, from which the difference between measured and predicted 2D optical flow  $\{u-u_{est}, v-v_{est}\}$  at every point can be stored. The sample standard deviation  $\sigma(\{u-u_{est}\})$ , etc. for each flow component is computed, and points within  $\pm\sigma$  are retained as inliers; these correspond to static background pixels. These inliers are then used to re-estimate the correct motion parameters using LMSE.

## 4. Moving Object Detection Under General Motions

### 4.1. Predicting optical and disparity flow

As part of the outlier detection process described above, we predict the optical flow at each pixel, given the estimated 6 DOF egomotion parameters and the estimated disparity at each pixel. These predictions are differenced from the observed optical flow to form residuals for outlier detection. Objects that are not rigidly fixed to the stationary background will generate optical flow that results in such outliers, modulo noise thresholds. For a given object velocity, however, the induced optical flow will be larger if the object is moving parallel to the image plane than if it is moving parallel to the optical axis (looming); hence, looming motions will be harder to detect. Up to now we have not discussed using predicted versus observed change in disparity (i disparity flow) to assist with outlier or moving object detection. Since this is particularly valuable for detecting objects moving along the optical axis, we discuss that now.

Following [15], from the relations  $d = fb/Z$  and  $\delta = dd/dt$ , we obtain:

$$\delta = \frac{-dZ}{dt} \frac{fb}{Z^2} = \frac{-dZ}{dt} \frac{d^2}{fb} \quad (5)$$

From equation (1),

$$\frac{dZ}{dt} = -V_Z - \omega_X y + \omega_Y x \quad (6)$$

Combining (4) and (5) gives the following relation between disparity, disparity flow, and egomotion:

$$\delta = \frac{V_Z d^2}{fb} + \frac{d}{f} (\omega_X y - \omega_Y x) \quad (7)$$

Thus, once we have estimated disparity and egomotion at a given frame, equation (7) gives the change in disparity that the scene point at each pixel will experience due to egomotion. Differencing this from measured change in disparity helps detect outliers and independent scene motion, particularly for looming motions. Note that implementing this requires using the predicted optical flow vector at time  $t_{i-1}$  to index into the disparity map time  $t$  to get observed disparity change for each pixel. Since the optical flow vectors are estimated to subpixel precision, we do bilinear interpolation of the new disparity field in this process. If any of the four neighbor disparities are missing, we omit this test at that pixel.

#### 4.2. Postprocessing and Moving Object Segmentation

So far, we have discussed our approach to estimate 6 DOF robot motion and predict the 3D flow field (temporal derivatives in the  $(x,y)$  monocular image and change in stereo disparity space) due to the robot egomotion. The difference between predicted robot image/disparity egoflow  $[u^R \ v^R \ \delta^R]$  and computed image/disparity flow  $[u^c \ v^c \ \delta^c]$  yields residual image optical and disparity flow fields where moving objects are highlighted. This involves a 3D vector difference:

$$[u^M \ v^M \ \delta^M] = [u^R \ v^R \ \delta^R] - [u^c \ v^c \ \delta^c] \quad (7)$$

that gives a 3D flow field attributed purely to moving objects. This operation in effect cancels out the effects of temporal inter-frame changes caused by robot motion, and ideally yields zero-valued flow vectors on static background pixels. Therefore, thresholding the flow and disparity residual fields at every pixel will yield a binary map that potentially contains moving objects as binary blobs or groups of moving pixels.

A few false alarms may be present in the binary map due to errors in disparity estimates and/or egomotion estimation errors. Advanced post processing that uses range measures and clusters groups of pixels based on consistency in range and velocity will assist in false alarm removal. Possible solutions include Markov Random Fields that model velocity and spatial similarities, or the Expectation Maximization algorithm to cluster points in an unsupervised manner. However, such advanced image processing and clustering cannot run in currently available processors at near-video rate frame that is needed for fast motion detection.

We do a fast 3D blob labeling and motion outlier process where we take 3D measures to reject false blobs, and merge blobs that are adjacent in 3D space. While true 3D blob labeling will involve 26-connectedness of a 3D

matrix (containing 3D X,Y,Z coordinates), it is not suitable for real-time implementation. We use a simpler approach, which is computationally effective, yet robust at rejecting false alarms. We separate the binary motion residual image into depth planes, where we find a list of possible moving pixels at each depth plane. In our experiments, we used a resolution of 5 meters for each depth plane. We then do a 2D blob labelling to cluster the pixels at each depth plane. This is extremely fast, since efficient 2D blob coloring algorithms exist. A 2D blob area threshold is used to remove blobs with small 2D areas. The 3D distance  $X_D, Y_D$  between neighboring blobs with centers  $(x_0, y_0)$  and  $(x_1, y_1)$  are estimated using the following approximations:

$$X_D = (x_0 - x_1) \frac{Z_{PLANE}}{f_y}, \quad Y_D = (y_0 - y_1) \frac{Z_{PLANE}}{f_y}.$$

Disconnected blobs that are spatially adjacent are merged into one. For a given depth plane at range  $Z_{PLANE}$ , the 3D height and width of each blob in the depth plane is also estimated using the perspective projection relations:

$$H_{3D} = H_{2D} \frac{Z_{PLANE}}{f_y}.$$

A similar technique is used to estimate 3D width. Blobs with very large or small 3D width and height are rejected as false alarms. These steps are repeated at every depth plane. The binary mask formed by the union of postprocessed binary blobs at each depth plane is the final segmented moving object mask. This 3D blob segmentation was found to effectively remove outlier regions due to errors in flow and range estimation.

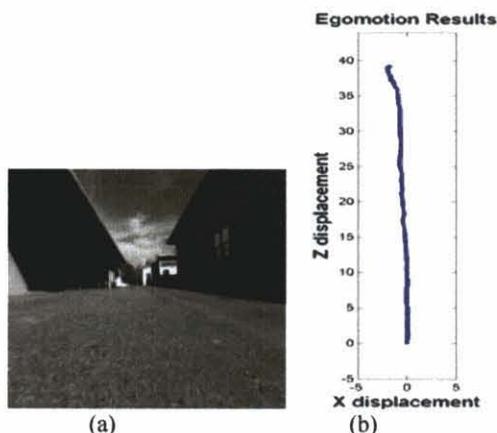
## 5. Results

We present results from two main components of our algorithm. Results of our real-time egomotion algorithm are presented in Section 5.1, and our real-time moving object segmentation results from moving robotic platforms are shown in Section 5.2.

### 5.1. Visual Odometry

Our 6-DOF robot egomotion estimation algorithm is computationally efficient since it involves two iterations of LMS estimates with an outlier rejection step. It was observed to run at 11.8 ms/frame (85 Hz) on a 1.4 GHz Pentium M machine, which makes it well suited for real-time systems. We evaluated the egomotion algorithm on three image sequences. The first sequence involves accurately measured translational robot motion on a translation stage with transverse object motion. The next sequence involves roughly transverse robot motion over an extended traverse in a static scene. The last sequence involves transverse and rotational robot egomotion estimation in the presence of a moving object. This helps us evaluate the outlier rejection capabilities of the egomotion algorithm when the robot undergoes rotational and translational motion.

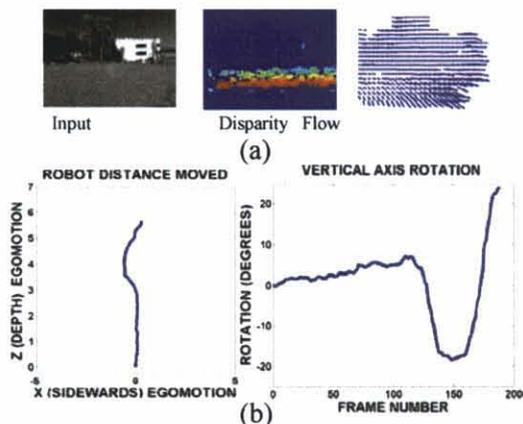
In the first sequence, the camera was placed on a translation stage and a car moved transversely parallel to the image plane at about 15 m (see Fig. 6 for representative image and Section 5.2 for description of



**Fig 4:** Robot egomotion estimates (b) from straight line motion along an alley, showing (a) first frame

the data). The true motion per frame was 20 mm/frame. The average estimated egomotion (with outlier estimation to reject pixels on the moving car) along the Z-axis per frame was 18.7 mm/frame, corresponding to a translation displacement error of 6.5%. The standard deviation of the displacement estimate in the Z-axis was 0.35mm/frame, significantly lesser than the mean error. This hints at the possibility of a consistent bias in the egomotion estimates that could be due to a miscalibration of the stereo cameras.

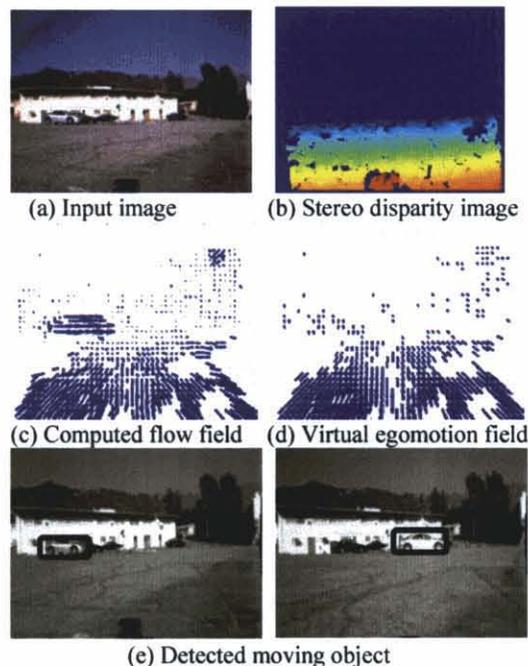
In the static scene, the robot moved down an alley, in a relatively straight path (Z-axis) with a small, steady displacement to the left side (X axis). The first and last



**Fig 5:** (a) Rotational motion sequence with moving human as shown by optical flow and stereo disparity, and (b) robot egomotion displacement and rotation results.

image in the sequence are shown in Fig. 4(a). The robot travelled a total distance of 42 metres along the alley. The estimated robot motion is shown in Fig. 4(b) as a function

of motion along X (sideways) and along Z (depth). The total estimated motion was 39.6 metres, which translates to a 5.7% error. Since the same calibration files were



**Fig 6:** Results of moving object detection (e) from moving robotic platform combining stereo-disparity (b), flow (c), and visual odometry (d).

used as in the previous test, we feel that the egomotion estimation error could be reduced with better camera calibration.

The third sequence (Figure 5) was the toughest, with a moving object, and rotational and translational robot motion. In this sequence, the robot moved forward, while rotating from left to right and back along the vertical axis. The outlier rejection successfully estimated the robot rotational motion, and displacement along the Z-axis (Figure 5b, bottom), while a moving object moved parallel to the image plane, as shown in the top row of Figure 5. The robot rotational motion for one frame is revealed in the 2D optical flow field (figure 5a); note the flow due to the moving human in the right portion of the image.

## 5.2. Detection of Objects with General Motions

We tested our moving object segmentation algorithm for moving cameras on several real test cases, with a variety of motions, ranging from controlled and precisely measured robot motion to combinations of rotational and translational robot displacement. We tested dynamic scenes with transverse and radially moving objects.

In the first case, the cameras were placed on a translation stage and had well-calibrated interframe motion along the Z-axis. A car moved from left to right parallel to the image plane across the entire sequence. The object tracking result for one frame pair is shown in

Figure 6. The corresponding disparity and optical flow images are shown in Figures 6(b) and 6(c). The virtual egoflow field (computed from our 6-DOF visual odometry algorithm) caused by robot motion is shown in Figure 6(d). The difference of the computed flow and the virtual egoflow fields highlights the moving object in the

Section 5.1. The moving object segmentation results are shown as a bounding box for various frames in the sequence in Figure 8. False alarms were obtained in 15 frames out of a total of 188 frames. The false alarms were caused by errors in flow and range estimates. False alarms could be reduced by introducing temporal filtering to

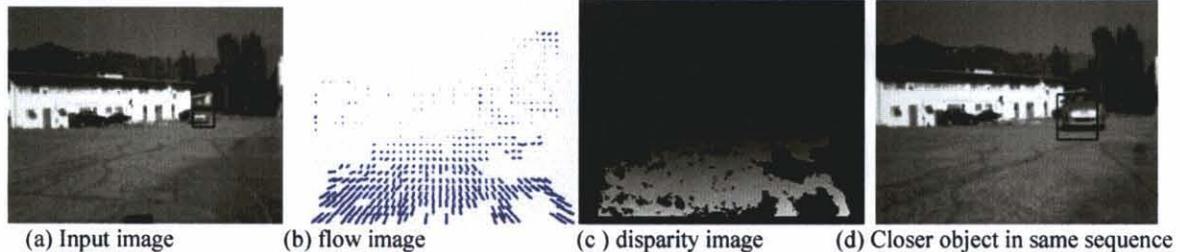


Fig 7: Results of looming car sequence as robot moves forward

scene, which is then thresholded to segment the moving object (Figure 6(e)).

In our second test, the object moved radially towards the camera (looming motion) as the camera moved forward. Due to object and camera motion along the Z-axis, the 2D image optical flow field has less information to segment the approaching moving object, as seen in Figure 7(b). Discriminating object motion along the Z-axis can be derived from inter-frame changes in the disparity flow field (or disparity flow), as discussed in Equation 6. Figure 7(c) shows the disparity image for the frame in Figure 7(a). The rectangular bounding box in Figure 7(a) for the first frame in the sequence shows the segmented moving object, in this case at a distance of 19 m. Figure 7(d) shows the last frame of the sequence, where the car was 16 m from the camera.

We then tested the moving object detection algorithm with unconstrained robot motion by placing the cameras on a Packbot robot (see Figure 5). This is a difficult case

retain consistent tracked objects and reject noisy regions in intermittent frames.

Our moving object segmentation algorithm, including stereo, optical flow, and robot egomotion estimation, was implemented in C/C++. The final blob segmentation module is implemented in MATLAB and not included on the run-time numbers. Initial tests of our algorithm indicate that the entire algorithm (end to end), including image capture, processing and display, runs at 54.6 ms/frame (18.3 Hz) on a 1.4 GHz Pentium M. This real-time demonstration of unconstrained moving object detection in complex scenes under general 6 DOF robot motion by fusing dense stereo and optical flow is, to our knowledge, the first of its kind in the computer vision and robotics community. Further speedup is expected after optimisation of the egomotion estimation code.

## 6. Conclusions and Future Work

We have discussed a new methodology to detect and

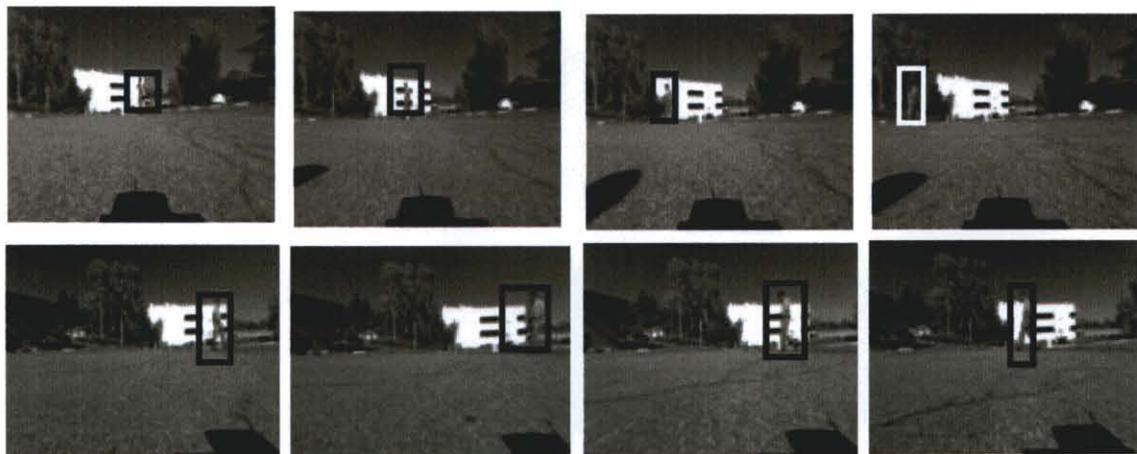


Fig 8: Tracking a moving human under general rotational and translational robot motion.

since the contrast of the moving object relative to the background is poor, as can be seen in Figure 8. The robot egomotion estimates for this sequence was discussed in

segment moving objects in the presence of general camera/robot motion in real-time by combining range from stereo with dense optical flow. A robust robot

egomotion algorithm is discussed that estimates all the 6-DOF robot motion. This egomotion estimate is then used to remove flow caused by robot motion. A novel scheme for detecting 3D object motion by evaluating 2D optical flow in conjunction with disparity flow is detailed. Our results show the potential of the real-time algorithm for egomotion in complex, cluttered scenes, and its use in detecting moving objects with unconstrained motion. Our algorithm has potential applications in dynamic scene analysis for automated transportation systems, unmanned air and ground vehicles, and also for navigation and position/pose estimation indoors/outdoors during loss of GPS and IMU failure.

Potential improvements to our algorithm include temporal tracking of multiple moving objects, handling of occlusions, and combining sparse feature tracking with dense optical flow to further improve tracking accuracy and speed of the algorithm

### Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the DARPA-IPTO Mobile Autonomous Robot Software (MARS) Robotics Vision 2020 Program through an agreement with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

### 7. References

[1] Dickmanns, E.D. "The development of machine vision for road vehicles in the last decade". in *IEEE Intelligent Vehicles Symposium*. 2002.

[2] Rosales, R. and S. Sclaroff. "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions". in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,. 1999. Fort Collins, Colorado: p. 117-123.

[3] Collins, R.T., et al., "Algorithms for cooperative multisensor surveillance". *Proceedings of the IEEE*, 2001, **89**(10): p. 1456-77.

[4] Eveland, C., K. Konolige, and R.C. Bolles. "Background modeling for segmentation of video-rate stereo sequences". in *IEEE Conference on Computer Vision and Pattern Recognition*. 1998. Santa Barbara.

[5] Cutler, R. and L. Davis. "Monitoring human and vehicle activities using airborne video". in *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*. 1999: p. 146-153.

[6] Stein, G.P., O. Mano, and A. Shashua. "Vision-based ACC with a Single Camera: Bounds on Range and Range Rate Accuracy". in *IEEE Intelligent Vehicles Symposium 2003*. 2003. Columbus, OH.

[7] Franke, U. and S. Heinrich, "Fast Obstacle Detection for Urban Traffic Situations". *IEEE Trans. Intelligent Transportation Systems*, 2002, **3**(3): p. 173-181.

[8] Ke, Q. and T. Kanade. "Transforming Camera Geometry to A Virtual Downward-Looking Camera: Robust Ego-Motion Estimation and Ground-Layer Detection". in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2003)*. 2003. Madison: p. 390-397.

[9] Nister, D. "An efficient solution to the five-point relative pose problem". in *IEEE Conference on Computer Vision and Pattern Recognition*. 2003. Madison, WI, USA: p. 195-202.

[10] Olson, C., et al. "Stereo Ego-motion Improvements for Robust Rover Navigation". in *Intl. IEEE Conference on Robotics and Automation*. 2001: p. 1099-1104.

[11] Mallet, A., S. Lacroix, and L. Gallo. "Position estimation in outdoor environments using pixel tracking and stereovision". in *IEEE International Conference on Robotics and Automation*. 2000.

[12] Costeira, J. and T. Kanade. "A Multi-Body Factorization Method for Motion Analysis". in *International Conference on Computer Vision*. 1995.

[13] Dang, T., C. Hoffmann, and C. Stiller. "Fusing optical flow and stereo disparity for object tracking". in *IEEE Intl. Conf. Intelligent Transportation Systems*. 2002: p. 112-116.

[14] Enkelmann, W., "Interpretation of traffic scenes by evaluation of optical flow fields from image sequences". *Control, Computers, Communications in Transportation*, 1990: p. 43-50.

[15] Waxman, A.M. and J.H. Duncan, "Binocular image flows: steps towards stereo-motion fusion". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. **PAMI-8**(6): p. 715-729.

[16] Adiv, G., "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1985. **7**(4).

[17] Willersinn, D. and W. Enkelmann. "Robust obstacle detection and tracking by motion analysis". in *Proc. IEEE Conf. Intelligent Transportation Systems*. 1998: p. 717 - 722.

[18] Talukder, A., et al. "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles". in *IEEE IROS*. 2003. Las Vegas, NV.

[19] Barron, J.L., D.J. Fleet, and S. Beauchemin. "Performance of optical flow techniques". *International Journal of Computer Vision*, 1994. **12**(1): p. 43-77.

[20] Longuet-Higgins, H.C. and K. Prazdny, "The Interpretation of a Moving Retinal Image". *Proceedings of the Royal Society of London B*, 1980. **208**: p. 385-397.