Real-time detection of moving objects in a dynamic scene from moving robotic vehicles

A. Talukder, S. Goldberg, L. Matthies, A. Ansar

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA. Tel. (818)354-1000 – Fax (818)393-3302
Email: [ashit.talukder, steve.goldberg, larry.h.matthies, adnan.iansar]@.jpl.nasa.gov

Abstract

Dynamic scene perception is currently limited to detection of moving objects from a static platform. Very few inroads have been made into the problem of dynamic scene perception from moving robotic vehicles. We discuss novel methods to segment moving objects in the motion field formed by a moving camera/robotic platform in real time. Our solution does not require any egomotion knowledge, thereby making the solution applicable to a large class of mobile outdoor robot problems where no IMU information is available. We address two of the toughest problems in dynamic scene perception on the move, using only 2D monocular grayscale images, and secondly where 3D range information from stereo is also available. Our solution involves optical flow computations, followed by optical flow field preprocessing to highlight moving object boundaries. In the case where range data from stereo is computed, a 3D optical flow field is estimated by combining range information with 2D optical flow estimates, followed by a similar 3D flow field preprocessing step. A segmentation of the flow field using fast flood filling then identifies every moving object in the scene with a unique label. This novel algorithm is expected to be the critical first step in robust recognition of moving vehicles and people from mobile outdoor robots, and therefore offers a general solution to the field of dynamic scene perception. It is envisioned that our algorithm will benefit robot scene perception in urban environments for scientific, commercial and defense applications. Results of our real-time algorithm on a mobile robot in a scene with a single moving vehicle are presented.

Keywords: Computer vision, dynamic scenes, moving object detection, optical flow, robotics, segmentation, scene understanding.

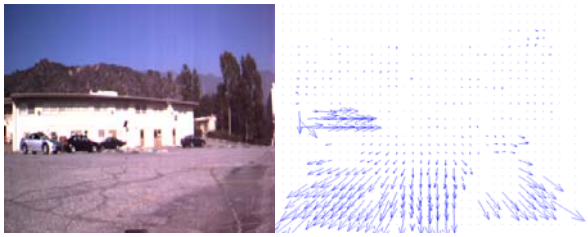
1. Introduction

The robotics community to date has mostly focused on autonomous robot operation in static scenes [1], or highly constrained dynamic scenes [2]. Autonomous operation in urban scenes however realistically involves operation in the presence of moving objects, either to avoid hitting or being hit by moving vehicles/people, or detect, track, and approach moving people and vehicles in the scene. Moving object detection from fixed cameras using optical flow algorithms that employ region-based correspondence measures [3], or feature-based matching [3] have been discussed extensively. However, 2D optical flow information by itself is insufficient to locate moving objects on the move due to the effective motion of background pixels, as shown in Figure 1b. It is unrealistic, and uneconomical to stop a robot frequently to enable it to localize moving objects. Therefore, we propose new techniques that will allow moving object detection on the move in real time.

We classify the problem of moving object detection into four categories:

1. Detection of moving objects from a static camera
2. Detection of moving objects from a moving camera with known egomotion (from IMU, etc.), and knowledge of 3D/depth/range information (from stereo)
3. Detection of moving objects from moving camera without any knowledge of egomotion, and knowledge of 3D/depth/range information (from stereo)
4. Detection of moving objects from moving camera without any knowledge of egomotion, and no knowledge of 3D/depth/range information

As mentioned previously, prior work on moving object detection has mostly concentrated on the first category

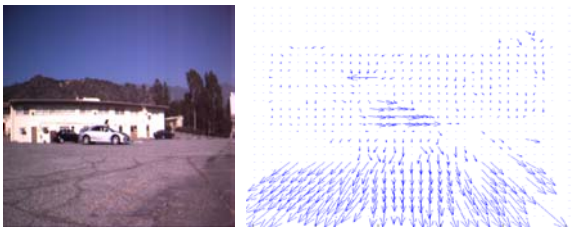


(a) Input observed image (b) 2D Optical flow vectors

Figure 1*: Typical estimated 2D optical flow image of a moving car observed from a moving robot platform*

[3]. The last two scenarios where no egomotion information is available are the toughest problems to solve, and have not been addressed much in previous research efforts. In [4], the expected image motion was computed using vehicle odometry. In [5], a quadratic motion model

is used to compute car motion on flat road surfaces and outliers are detected using robust multiresolution techniques. Such a solution is elegant, but requires expensive computing power and does not account for vehicle vibration on uneven surfaces. Vibration and abrupt motion of the robot vehicle on uneven surfaces, coupled with errors in optical flow estimates causes spurious optical flow vectors, as shown in Figure 2. These issues significantly complicate the problem of robustly detecting moving objects from mobile platforms.



(a) Moving car (b) Spurious flow (vibrating platform)

***Figure 2**: Moving car in scene and spurious computed optical flow caused by vibrating robot vehicle.*

We present a general solution for the problem of object detection in dynamic scenes where no knowledge of robot egomotion is available. We address this problem for two cases: (a) a stereo-camera robot system, where range information is available, and (b) a monocular camera platform with no range information. While our proposed algorithms for dynamic object detection should be sufficiently robust to require no prior knowledge of robot egomotion, they are required to run in real-time on a mobile robot platform moving at up to 0.75 m/s with a camera frame rate of up to 15 Hz. These system constraints pose a formidable challenge algorithmic requirements

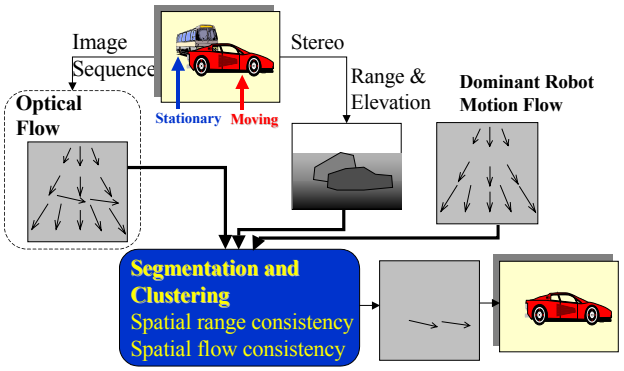


***Figure 3**: Proposed general solution for real-time detection of moving objects in dynamic scenes from moving robotic platforms*

The paper is organized as follows. In Section 2, we discuss our real-time optical flow algorithm that is used to estimate temporal-spatial motion patterns in dynamic scenes. In Section 3, we detail our real-time segmentation algorithm to segment moving objects in the presence of robot egomotion using monocular cameras, and in Section 4, we discuss moving object segmentation on the move with stereo-cameras. Results are presented in Section 5.

2. Real-time Optical Flow

Optical flow computation involves estimation of image motion fields from temporal variations of spatial image data. Optical flow fields correspond to 2D projections of 3D movements of surfaces in a scene, either due to dynamic objects in the scene being imaged, or due to egomotion of the robotic platform on which the camera is placed. Optical flow traditionally involves feature or region tracking [3], coupled with velocity smoothness constraints. Feature tracking solutions, such as the LK tracker, are fast, but yield sparse flow maps that may be insufficient for moving object detection on the move; therefore, we use a region-based flow technique. A smoothing constraint is not used to avoid artificially smoothing the flow field, which could disrupt moving object segmentation. We briefly discuss our implementation of the real-time flow algorithm below.

Our real-time optical flow algorithm is based on the sum of absolute difference (SAD) and is implemented using a vectorized sliding sum method. Each correlation score is generated by computing the SAD of a window in the current image with a window of the same size in the previous image. The resulting unsigned value is our correlation score. To compute multiple scores we start with the window in the upper left corner of our flow search window. A sliding correlation window is used, moving left-right, top-down over the entire search area. This left to right, top to bottom progression aids in the cache efficiency of our algorithm as well as allows the position of the minimum score to be stored as an index into this search space. The minimum score is computed by comparing vectors of correlation scores from each position in the search window. The index of the minimum correlation score is then used in a lookup table to determine the pixel offsets for the best matching correlation window.

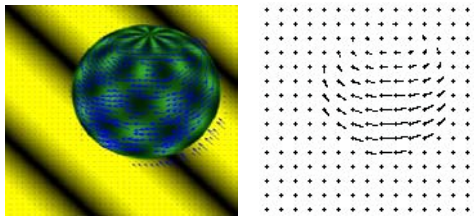


***Figure 4**: Real-time optical flow (blue arrows) estimates on rotating sphere and true flow vectors.*

A sub-pixel flow estimate at each pixel is generated by computing a quadratic fit on the center and neighboring correlation scores. This provides a fixed point estimate of the flow vectors. Each estimate is represented as two eight bit values allowing us to search a 15x15 pixel flow window and limiting the precision of our estimate to 1/16 of a pixel. Figure 4 shows computed optical flow using the JPL real-time implementation. The true flow field is also shown for comparison.

3. Monocular Moving Object Detection on the move

Monocular camera-based moving object detection on the move involves extraction of signatures of moving objects from the 2D optical flow field in the presence of 2D motion flow caused by robot egomotion. This is significantly tougher than detection of moving objects from static cameras, where optical flow magnitude can be used to segment the moving objects in the scene. Moving objects are characterized by discontinuities in the 2D robot motion flow field. Algorithms that inherently use region homogeneity criteria and locate such discontinuities are therefore useful for segmenting moving objects from a optical flow field.

Markov random field models are popular methods for segmenting spatial fields based on region homogeneities/texture; however, they use slow, iterative relaxation procedures such as EM or simulated annealing to converge to the best segmentation solution. Quadtree schemes for segmentation of color images have been used, but are not very effective in handling discontinuities in small regions, and yields "blocky" segmented images. Manjunath et. al. [6] use Gabor texture feature gradient orientation to identify region boundaries. Robust image segmentation methods using watershed [10] implementations that run at 0.1 secs for 128x128 images have been discussed [9]. However, it consumes too much memory and computational resources to be run on our real-time robot platform that also computes stereo and the optical flow vector field at the same time. We propose the use of fast, yet robust, segmentation techniques that will efficiently segment out moving objects in the presence of robot egomotion. We preprocess the computed optical flow image to highlight boundary regions between moving objects and background pixels. This preprocessed image is then segmented to yield labeled moving object regions. Details of these steps are provided below.

3.1. Monocular Optical Flow Image Preprocessing

Moving objects in an image are typically characterized

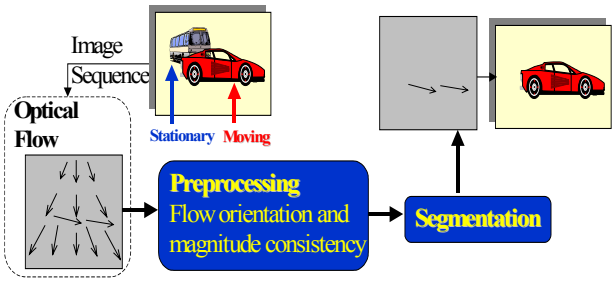
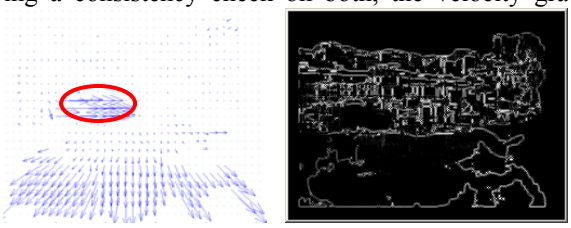


***Figure 5**: Real-time monocular moving object segmentation concept*

by a discontinuity in the orientation of the 2-D optical flow at the object pixel, and a change in magnitude of the 2-D flow at the object pixel, relative to background pixels. A change in flow magnitude is observed since typically the moving object moves in a different direction and with a different velocity than perceived motion in background pixels due to robot egomotion. However, the flow magnitude gradient is not completely sufficient, since the presence of a distinct static vertical background structures closeby, with neighboring (2D adjacent) distant background pixels will yield large 2D optical flow gradients because of the smaller flow vectors on distant pixels, relative to the larger displacement of nearer pixels. Therefore, regions that have large changes in flow orientation coupled with large flow gradients are labelled as potential moving objects. This amounts to doing a consistency check on both, the velocity gradient



(a) Flow with moving car (marked) (b) Laplacian of the orientation flow map

***Figure 6**: Laplacian of orientation map from flow field*

and orientation gradient at each point. Pixels that do not satisfy both criteria are discarded.

The orientation map of the optical flow field $\underline{F}_2 = (u_{xy}, v_{xy})$ at pixel (x,y) is defined as $O_{x,y} = \tan^{-1}(v_{x,y} / u_{x,y})$. High-pass filtering the orientation map with a 3x3 Laplacian filter highlights region boundaries with sharp variations in flow orientation. We denote this image as $DO_{x,y}$. The Laplacian of an orientation map (Figure 6b) for a car moving horizontally forward (marked with a circle in Figure 6a) as the robot moves is shown in Figure 6.

The gradient of a vector field can be computed in several ways. The divergence of a vector field $\underline{F}_2 = (u_{xy}, v_{xy})$ at (x,y), $\nabla . \underline{F}_2 = \partial u/\partial x + \partial v/\partial y$ measures the ratio of the incoming vectors to outgoing vectors at that point. In other words, it measures the net outflow at every point in the vector field. The curl of a vector, $\nabla \times \underline{F}_2$, measures the rotational component at every point in the vector field. Both these definitions are inadequate to extract the gradient information for moving object detection. We compute the gradient in the following manner:

$$\nabla \underline{F}_2 = \begin{bmatrix} \partial u/\partial x & \partial u/\partial y \\ \partial v/\partial x & \partial v/\partial y \end{bmatrix}$$

The determinant of this matrix measures the volume spanned by the gradient vectors. However, if one of the vectors has zero components, the resultant volume of the gradient matrix will be zero; therefore, the determinant is not a suitable measure. We use the 1-norm of a matrix $|\nabla \underline{F}_2|_1 = \max(|\partial u/\partial x| + |\partial v/\partial x|, |\partial v/\partial x| + |\partial v/\partial y|)$. The gradients, $\partial u/\partial x$, $\partial v/\partial x$, etc. are computed using a Sobel operator on each of the horizontal and vertical flow component fields. The gradient of the flow field in Figure 6a is shown in Figure 7a.

Since boundaries of moving objects are marked by sharp changes in velocity and orientation gradient, the orientation flow field is used to mask the gradient flow field by clippings its values when the orientation gradient is low, to generate a moving object boundary image *M* in the following manner:

$$M_{x,y} = |\nabla \underline{F}_2|_1 \quad \text{IF } DO_{x,y} > T$$
$$= 0 \qquad \text{Otherwise}$$

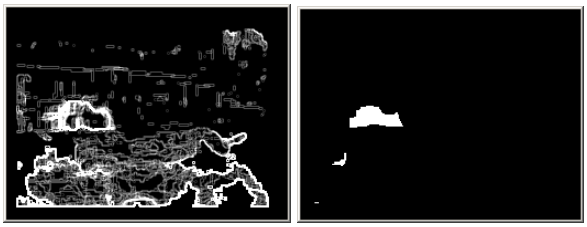
3.2. Monocular moving object labelling

After the preprocessing steps, moving object boundaries have high values whereas static background and the inside of moving objects are suppressed. Therefore, regions inside closed boundaries would correspond to moving objects. While adaptive thresholding followed by closed-contour detection or a binary/grey watershed are feasible solutions, they are computationally expensive.

Flood filling of the moving object boundary image *M,* using neighborhood pixel changes as a stopping criterion is a robust solution. The flood filling procedure used to fill a neighboring pixel is:

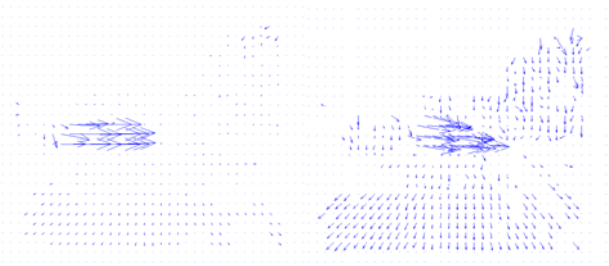
$$M(x',y') - T_M <= M(x,y) <= M(x',y') + T_M, \quad .$$

where M(x',y') is the value of one of pixel neighbors, and T_M is the threshold that should be exceeded in order for the flood filling to stop. Therefore, to be added to the



(a) Gradient of flow field (b) Segmented image

Figure 7*: (a) Flow field gradient of the image in Figure 4a,b and (b) monocular segmented moving object*



(a) (b)

Figure 8*: 3D velocity fields with (a) right-moving car (Figure 1a) for smooth robot motion; (b) moving car (shown in Figure 2a) with vibrating robot motion*

connected component, a pixel should have at least one neighbour with similar value.

Flood filling from within a moving object would fill the inside of that object. However, this would require a seed inside every moving object. Not only would this require a-priori knowledge about the number of moving objects and the approximate location of the center of each moving object, but additionally N independent flood filling runs would need to be done, thereby increasing computation time. Instead, flood-filling of the moving object boundary image *M* starting from a background pixel would fill every background pixel, and yields a labeled image where the insides of every moving object have not been filled. Figure 7b shows the final segmented moving object regions for the scene shown in Figure 4a after flood-filling. A flow consistency check on every unfilled region is then used to discard false background regions, and retain true moving objects.

4. Stereo-based moving object detection on the move

4.1. 3D velocity estimation

Before discussing the stereo-based algorithm for dynamic scene detection, we need to define the notation for 3D scene velocity, and 2D image velocity. Optical flow typically estimates 2D pixel motion, but when combined with range/depth information can yield estimates of true 3D pixel velocity. The 2D flow (2D velocity) at pixel *(x,y)* in an image can written as $\underline{F}_2 = (u_{xy}, v_{xy})$. The corresponding 3D velocity of the pixel in 3D space is denoted in caps: $\underline{F}_3 = (U_{XYZ}, V_{XYZ})$. We will ignore the subscripts (spatial dependencies) for 2D and 3D velocities in the discussion for ease of notation, and refer to the 2D velocities at (x,y) as *(u,v)* and 3D velocity at (X,Y,Z) as *(U,V)* respectively. Note that the relation between a point *(X,Y,Z)* in 3D space and its projection *(x,y)* in 2D image space is given by:

$$x = k\,X / Z, \quad y = k\,Y / Z \qquad (1)$$

where k is dependent on camera optics. Ignoring k, we can denote the 3D velocity along the X and Y axes as:

$$U = dX/dt = d(x \,.\, Z)/dt \;;\; U = Z \,.\, dx/dt + x \,.\, dZ/dt$$
$$V = dY/dt = d(y \,.\, Z)/dt \;;\; V = Z \,.\, dy/dt + y \,.\, dZ/dt \qquad (2)$$

If we assume that the stereo range estimates are coarse such that a change in depth between two consecutive frames at a single pixel cannot be detected for now (i.e. dZ/dt=0), then the 3D velocities simplify to the following:

$$U = Z \,.\, u$$
$$V = Z \,.\, v \qquad (3)$$

In other words, scaling the 2D flow velocities at each pixel by its depth/range yields an approximation to the 3D flow velocity at that pixel. Figure 8a shows the 3D estimated 3D velocity vectors for the moving car in Figure 6a. Note that the background motion vectors due to the moving camera/robot tend to have similar 3D velocity components, compared to the 2D velocity vectors in Figure 6a where closer objects have larger 2D velocity magnitudes. Therefore, segmenting the moving car from a moving robot now becomes much easier using 3D velocity estimates. Figure 8b shows the 3D velocity field for a vibrating downwards robot motion shown in Figure 2. The 3D velocities of most of the background pixels now uniformly reflect the downwards robot motion, independent of range.

4.2. 3D flow-based segmentation of moving objects

Knowledge of 3D velocity at every pixel greatly aids moving object segmentation on the move, as we now discuss. For a moving robot platform, the dominant motion in the scene would be due to egomotion, if much of the scene was occupied by static background pixels (i.e. moving objects in the scene occupy relatively fewer pixels than background). Therefore, estimation of the dominant motion, followed by detection of regions with outlier velocities would assist robust segmentation of moving objects on the move.

Several outlier detection methods [7] can be used, including least-mean squares, or least-median squares which is more robust than mean-square estimates, K-means clustering, robust estimation methods [5], or advanced clustering techniques and new deviation measures [8]. Many of these solutions, while being powerful, are not suitable for real-time outlier detection with limited computation and memory. Therefore, we use a fast Gaussian-model based outlier detection method, where we assume a Gaussian distribution for the estimated robot 3D velocity. If the true robot velocity vector in the X,Y directions is given as (V_m, U_m), the estimated velocity vector at a background pixel (x,y) is modeled as $(V_x, U_y) = (V_m, U_m) + N(0, \Sigma_{x,y})$, where N( ) is a zero mean, normally distributed random variable with variance $\Sigma_{x,y}$ (2x2 matrix). Assuming uncorrelated noise, the estimated background velocity at (x,y)simplifies to:

$$(V_x, U_y) = (V_m, U_m) + N(0, \sigma_{x,y}),$$

The variance $\sigma_{x,y}$ is generally independent of robot velocity, but is a function of depth; pixels at a greater distance could have larger uncertainties in depth estimates than closer pixels. We assume uniform uncertainty at all depths, which further simplifies the expression for the 3D velocity of a background pixel due to robot egomotion:

$$(V_x, U_y) = (V_m, U_m) + N(0, \sigma),$$

We use a 95% confidence interval test to generate a hypothesis for a valid background pixel as:

$$(x,y) = \text{Background} \quad \textbf{IF } (V_x, U_y) \in (V_m, U_m) \pm 2\sigma,$$
$$(x,y) = \text{Outlier} \quad \text{Otherwise} \qquad (4)$$

Labeling regions with outlier 3D velocities as moving objects may yield incorrect results, due to the inaccuracies in the outlier estimation process (background pixels could be classified as outliers if a conservative outlier threshold is chosen), and range estimation errors created during the stereo-matching process. Therefore, we also apply a flow consistency-based segmentation to these outlier regions, similar to the monocular case, that rejects regions with smooth 3D velocity field gradients thereby ensuring that false background regions are correctly assigned as background pixels. The preprocessing steps involve velocity field orientation estimation and velocity field gradient computations (Section 3.1) followed by flood filling, as discussed in Section 3.2.

5. Results

We present results of our real-time optical flow algorithm and real-time moving object segmentation technique. The optical flow algorithm was developed and implemented in-house at JPL, and currently runs at 6.8 Hz on a 320x240 image on a 2.1GHz PC platform. The monocular and stereo-based optical flow segmentation algorithms were implemented in C++, and they use the Intel OpenCV library primitives. The segmentation algorithms currently run at 10 Hz on a 900 MHz PC, and it is expected that they can be sped up after further optimization.

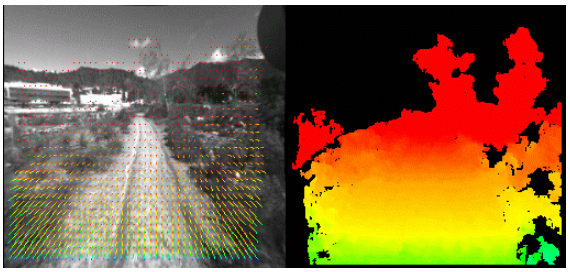


***Figure 9**: Optical flow field (superimposed) and range data from a moving robot in a static scene*

Figure 9 shows results of a moving robot in a static scene. The monocular and stereo-based moving object segmentation algorithms correctly label all pixels in the scene as background pixels.

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

***Figure 10**: (a) Smooth robot motion with moving car and 2 several stationary cars (b) 2D flow field and (c) monocular segmented image; (d) 3D flow field from stereo and (e) stereo-based segmented moving object.*

We then tested our optical flow algorithm and the moving object segmentation algorithms on scenes with forward motion of the robot and one moving object with translational motion. To analyze the accuracy of the algorithms, tests were done when the robot had smooth forward motion, and also when it was subjected to vibrational upward and downward motion, caused by unevenness on the road surface.

Figure 10a shows a moving car with forward smooth robot motion on an even road surface. The 2D optical flow image is shown in Figure 10b and the monocular

segmentation algorithm (Figure 10c) detects the moving car and discards the stationary ones in the background. The false alarm regions can be discarded based on region size and optical flow consistency measures. Figure 10d shows the estimated 3D velocity field from stereo-range, where the background pixels have similar 3D velocity, which simplifies moving object detection. The 3D stereo-based segmented car is shown in Figure 10e. Note the absence of any false alarms, compared to the monocular segmentation case.

Figure 11a illustrates the performance of our algorithms on an uneven road surface where the camera undergoes downwards vibrational motion. The monocular segmentation algorithm (Figure 11b) locates the moving car but also falsely detects ground regions due to spurious 2D flow vectors caused by camera motion on the uneven surface. The stereo-based segmentation algorithm handles the robot vibrations much better, as shown in Figure 11e.

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

Figure 11*: (a) Vibrational robot motion on uneven road with moving car (b) 2D flow field and (c) monocular segmented image; (d) 3D flow field from stereo and (e) stereo-based segmented moving object.*

6. Conclusions and Future Work

We have presented a real-time algorithm to detect multiple moving objects as the robot undergoes egomotion. Our technique does not require any knowledge about robot egomotion from IMU and handles camera/robot vibrations on uneven surfaces, thereby making the solution very general and applicable to various dynamic perception problems. This technique represents a clear improvement to traditional dynamic perception procedures. Initial tests with single and multiple moving objects in the scene show excellent results. Several improvements can be done to improve the generality and robustness of the system. Better outlier detection methods can be employed for monocular and 3D-stereo based optical flow to better locate moving objects in the presence of robot vibrations during motion. Better 3D velocity estimates by considering the velocity in the Z-direction can significantly improve moving object detection accuracy. A better flood-filling procedure (such as the watershed) could improve performance. Additionally, incorporating IMU information, if it is available, into the segmentation scheme will definitely result in better detection of moving objects on the move. We are also exploring the possibility of estimating the focus of expansion to determine the direction of dominant motion (caused by robot motion), which will better assist in locating objects with outlier velocities.

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-ITO 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] L. Pedersen, "Autonomous characterization of unknown environments" IEEE ICRA, Vol. 1, pp. 277 – 284, May, 2001,

[2] M. Veloso, et. al., "Playing soccer with legged robots", Proc. IEEE IROS, pp. 437-442 vol.1, Oct 1998.

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

[4] W. Enkelmann. "Obstacle detection by evaluation of optical flow field from image sequences", Image and Vision Computing, 9:160–168, 1991.

[5] Gildas Lefaix, Eric Marchand, Patrick Bouthemy, "Motion-based Obstacle Detection and Tracking for Car Driving Assistance", IAPR/ICPR'2002, Vol. 4, pages 74–77, 2002.

[6] W. Y. Ma and B. S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation," *Proc. IEEE CVPR*, June 1997, pp. 744-49.

[7] Peter J. Rousseeuw, Annick M. Leroy, "Robust Regression and Outlier Detection", Wiley Publishing, ISBN: 0-471-85233-3, October 1987

[8] Andreas Arning, Rakesh Agrawal and Prabhakar Raghavan, "A Linear Method for Deviation Detection in Large Databases", *Knowledge Discovery and Data Mining*, pp. 164-169, 1996.

[9] P. De Smet, R. Pires, D. Vleeschauwer, I. Bruyland, "Activity Driven Non-linear Diffusion for Color Image Watershed Segmentation", Journal of Electronic Imaging, Vol 8 (3), 1999.

[10] Ashit Talukder, et. al, "Modified Binary Watershed Transform for Segmentation of Agricultural Products", *Proc. SPIE* , Nov 1998.