Ground Plane Segmentation Using Optical Flow Templates

Alexander Lambert, Krunal Chande, Aaron Bobick

***Abstract*—The study demonstrates a subspace-based approach for ground plane segmentation using optical flow in the context of mobile robotics. This method does not explicitly depend upon odometry for expected flow estimation, but relies on previously derived templates, or basis flows. For strictly planar egomotion on a flat surface, the optic flow is quadratic and of a specific structure. The ground-plane flow can then be expressed as a linear combination of independent basis flows, allowing it to be recovered and differentiated from potential obstacles. The approach is verified in simulation, as well as on a real mobile platform, and is shown to be a viable strategy for the intended purpose. However, limitations are evident and possible improvements are outlined.**

I. Introduction

Knowledge of the surrounding environment and its geometry is time-critical in many mobile robot applications. Not only is it necessary for obstacle avoidance, such as in navigation of autonomous vehicles, but it can also be a key factor for detecting expected objects in a time-sensitive manner. An example could consist of a quad-copter drone deployed to clear a building of suspicious packages, or spot immobile individuals at a disaster site.

Many approaches which use 3D point clouds for segmenting objects in a scene are not practical for real-time navigation and obstacle avoidance. They require further processing and are computationally expensive. Additionally, many methods of pixel labeling and map generation typically rely on image features, which might be subject to variables such as lighting, pose, clutter in the background, etc. Laser scanners and LIDAR are fiscally expensive, heavy, and consume a relatively large amount of power. Wheel odometry and inertial-measurement units (IMUs) both suffer from large amounts of noise, and are prone to error accumulation unless corrected periodically using more reliable, but slower, sensors.

Optical flow, on the other hand, is directly an effect of scene geometry. Also, it is typically computed per frame without an associated history, and is computationally inexpensive and can be performed on a basic monocular camera. Given certain assumptions on the constraints of the motion and expected scene geometry, it has been shown that certain aspects, such as the ground plane, which have a statistical regularity over time, can be used to extract possible obstacles [1], [2], [3], [4].In [**?**], the authors manage to use expected optical flow structure to better estimate the location of obstacles and the flow itself. Roberts et. al.[3] use probabilistic PCA to learn optical flow subspaces which occur on a temporally regular basis. The authors also compare optical flow in superpixels against flow templates in order to label and distinguish obstacles from ground and distant features[3]. The concept of using motion constraints to better derive the ground plane can be used not only for obstacle avoidance, but object detection as well (given that enough geometric information can be extracted). Although even dense optical flow may offer only coarse geometric information about the scene, it could be used for drawing a mobile platforms attention to a potentially interesting feature. It can then slow down to inspect the feature and/or allocate more computational resources for more reliable object detection methods to be used.

The present study aims to evaluate the possibility of utilizing optical basis flows obtained from a camera on a robotic platform in order to differentiate between objects and the ground plane. We also make the assumption that the odometry data from the robot is inherently noisy and demonstrate an approach which uses only perceived optic flow to recover robot egomotion under some constraints. This expected robot motion is used to generate a motion model of the optic flow and we do further analysis to segment out the ground plane, i.e. the parts of the observed flow field which match our hypothesis, vs the Obstacles, i.e. the areas in the Flow Field that deviate from our hypothesis by a certain margin.

The paper is organized as follows: Section 2 describes the related work in the area of motion based image segmentation and Optic Flow. Section 3 goes in depth about the method of Optic Flow we are using. Section 4 outlines our approach and in Section 5 we show our results.We demonstrate our results using a dynamic model of a robotic platform in simulation as well as on a real platform.

II. Related Work

A. Dense Optical Flow

A dense optical flow method is preferred for segmentation of surfaces, as all pixels in the image contribute to the flow-field. The current study uses the Brox [5], [6] algorithm for computation of optical flow. The scheme was selected based on its performance using the Middleburry dataset [7]. The authors compute an energy functional which combines three basic assumptions: a brightness constancy assumption, which is similar to what we have already seen in Lucas and Kanade [8]; a gradient constancy assumption; and a discontinuity preserving spatio-temporal smoothness constraint. They get rid of any linearizations done and provide a theoretical basis for warping. We describe their assumptions in further detail and outline some of the math involved in them

- Brightness Constancy Assumption
  They assume that the brightness of a particular pixel does not change due to displacement. This can be formulated as
  $$I(x, y, t) = I(x + u, y + v, t + 1) \tag{1}$$
  The linearized version of this constraint yields the constraint
  $$I_x u + I_y v + I_t = 0 \tag{2}$$
  This however is valid under the assumption that the image changes linearly along the displacement, which is generally not the case. Hence they choose not to use it.
- Gradient Constancy Assumption
  The Brightness Constancy assumtion breaks down as soon as there is a slight change in the brightness of the scene. Instead they also incorporate the Gradient of the Gray value which is assumed to be invariant under displacement.
  $$\Delta I(x, y, t) = \Delta I(x + u, y + v, t + 1) \tag{3}$$
  where, $\Delta = (\delta_x, \delta_y)$ is the spatial gradient.
- Smoothness Assumption
  This assumption takes into account the interactions between neighbouring pixels. This constraint can be applied solely to the spatial domain or the spatio-temporal domain based on the frames available. Since there are expected to be some discontinuities at the boundaries of objects, it is sensible to generalize the assumtion by demanding a piecewise smooth flow field.

They then derive an energy functional when penalizes deviations from these assumptions. Let $x := (x, y, t)^T$ and $w := (u, v, 1)^T$.

$$\begin{aligned} E_{data}(u, v) = \int_\Omega \psi\Big(\big|I(x + w) - I(x)\big|^2 + \\ \gamma\big|\Delta I(x + w) - \Delta I(x)\big|^2\Big) dx \end{aligned} \tag{4}$$

And adding the smoothness constraint.

$$E_{smooth}(u, v) = \int_\Omega \psi\Big(\big|\Delta_3 u\big|^2 + \big|\Delta_3 v\big|^2\Big) dx \tag{5}$$

where the term γ is a weight between the two assumptions, $\psi(s^2)$ is an increasing concave function which makes it more robust[9][10]. And Δ_3 is the spatio-temporal gradient given by $\Delta_3 := (\delta_x, \delta_y, \delta_t)^T$

The total energy is the weighted sum between the data term and smoothness term.

$$E(u, v) = E_{data} + \alpha E_{Smooth} \tag{6}$$

Where α is some regularization parameter and we want to minimize (6).

III. Approach

A. Flow-Field Derivation

The homography for the optical flow of the ground plane can be greatly simplified if certain assumptions are made about the constraints on ego-motion. In the current approach, we assume that the mobile platform is moving on a flat surface, that the translational velocity is parallel to the ground and the angular velocity is perpendicular to the ground plane. The 2-D flow-field of a plane for a calibrated projective camera is then given by [1]

$$\begin{aligned} v_x &= c_{13}x^2 + c_{23}xy + (c_{33} - c_{11})fx - c_{21}f_y - c_{31}f^2 \\ v_y &= c_{13}xy + c_{23}y^2 + (c_{33} - c_{22})f_y - c_{12}f_x - c_{32}f^2 \end{aligned} \tag{7}$$

where,

$$c_{ij} = \frac{1}{fd(t)}(d(t)\hat{\omega} \cdot \hat{e_j} \times \hat{e_i} - \gamma_i V_j) \tag{8}$$

where v_x, v_y are the velocities of the vehicle in the vehicle frame, f is the focal length of the camera, x, y are the 2-D pixel coordinates, $d(t)$ is the distance between the viewed plane and the optical center, and (e_1, e_2, e_3) are the body reference frame axis.

The above equation can be re-written as a linear combination of two linearly independent basis flows:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \alpha X \begin{bmatrix} a_0 \\ \vdots \\ a_7 \end{bmatrix} + \beta X \begin{bmatrix} b_0 \\ \vdots \\ b_7 \end{bmatrix} \tag{9}$$

$$X = \begin{bmatrix} 1 & x & y & x^2 & xy & 0 & 0 & 0 \\ 0 & 0 & 0 & xy & y^2 & 1 & x & y \end{bmatrix} \tag{10}$$

where $[a_0 \dots a_7]^T, [b_0 \dots b_7]^T$ are the coefficient vectors corresponding to different basis flows, and α, β are scaling factors. Therefore, given a pair of ground-plane flow templates for two motion primitives, i.e. forward translational displacement and yaw, a given flow can be reconstructed as a combination of these two basic flow homographies. This would correspond to the *expected* flow of a ground plane for a particular translational and rotational velocity.

B. Analytic Solution

The previous equation can be simplified further if we assume that the translational velocity V is always parallel to the optical axis, and that the image plane is orthogonal to the ground plane[1]. The expected flow field can then be written as:

$$\begin{aligned} v_x &= \frac{\omega}{f}x^2 + \frac{V}{hf}xy + \omega f \\ v_y &= \frac{\omega}{f}xy + \frac{V}{hf}y^2 \end{aligned} \tag{11}$$

where ω is the angular velocity and h is the height of the camera from the ground, which is constant. Using the odometry of our system, we can obtain an estimate for V and ω and find our expected ground plane flow.

C. Segmentation

In order to obtain the basis flows, the coefficient vectors in 9 were found either by using the analytic solution described above, or by performing an iterative inlier segmentation (RANSAC) on basis flows obtained from simulation as well as from experiment using an actual mobile platform. In both these

cases, the flow homographies were calculated from a flowfield while undergoing a purely translational forward motion or a rotational motion. This was done in a scene with a largely visible ground-plane, where only the flow on the lower half of the image was considered as the camera was assumed to be parallel to the ground.

The coefficients of the known flow-templates were then used to extract points which corresponded to a ground plane in a flow-field from an observed scene. This was done using a similar iterative scheme as that used for basis derivation. However instead of solving for homography coefficients, the scaling factors α and β were found.

IV. Evaluation

Here we present the results of our approach. We first test the system in simulation using the open-source software, Gazebo[11], which includes an approximate dynamical model of our mobile platform, as shown in Fig.1a. The real system, aptly named *Jeeves*, is a modified Segway platform, developed by the Cognitive Robotics Lab at Georgia Tech. Support casters have been added to the platform such that active balancing is not needed. The robot boasts an array of fancy sensors, although the most relevant to this study is the original Microsoft Kinect (not so fancy). The RGB camera of the Kinect was used to gather empirical data for the experiments. The implementation uses the Brox optical flow scheme[5], [6] included in the OpenCV library[12].

We start of by demonstrating the output of our Optic Flow Pipeline in Figure 2. You have sparse arrows marking the direction and magnitude of flow at equally spaced intervals and a hue of red-green denoting magnitude and direction (green -¿ Left, red-¿Right), of the observed optic flow at each point.

In Figure 3 we apply a naive scheme of thresholding the Sum of Squared Differences of the (Observed $-$ Analytic Optic Flow). We can see that in simulation it does an excellent job of segmenting out the object. But we have to tune the Thresholding parameter for accurate results.

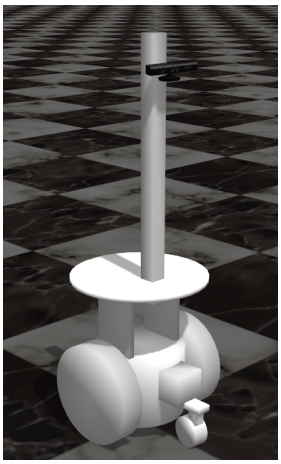
In Figure 4 We generate basis sets in simulation and find the results of segmenting the Ground plane of the same image using that basis set. We first do it by using an Analytic Form, and then learn using the observed flow model.

In Figure 5 we test our system in a real environment. We use the previously generated Basis sets from simulation to segment out the ground plane.

In Figure 6 we learn new basis sets from real data.

In Figure 7 we show the results of our implementation on real data using all three basis sets.

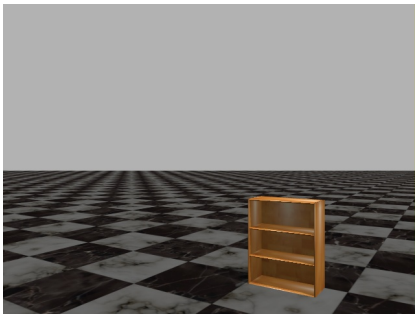
In Figure 8 we test the system in another environment.



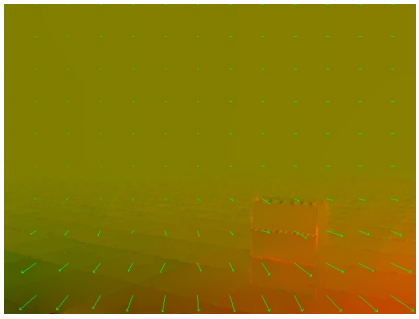
(a) Simulated Model of Jeeves in Gazebo

(b) Jeeves

Fig. 1: Mobile platforms used both in simulation and experiment.

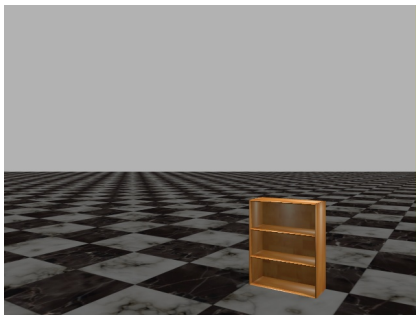


(a) Output of camera on Simulation Jeeves



(b) Resulting Brox Optic Flow

Fig. 2: Observed Optic Flow calculated using Brox Algorithm

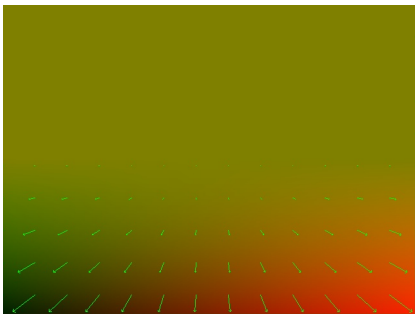


(a) Simulation Environment

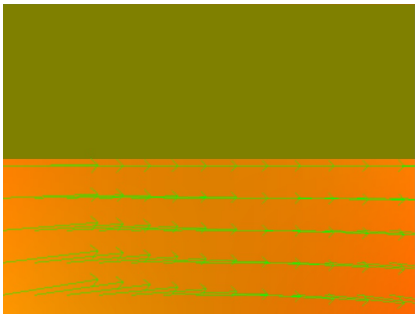
(b) SSD based on Analytic Flow

(c) Result from Thresholding

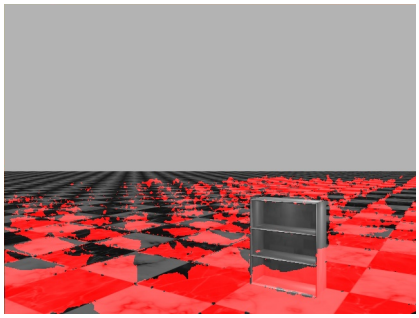
Fig. 3: Observed Optic Flow calculated using Brox Algorithm



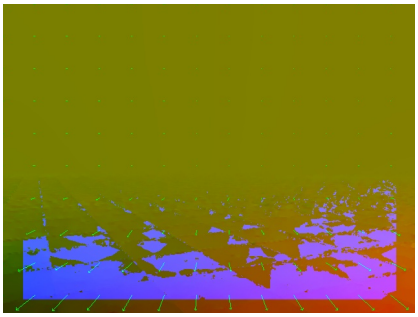
(a) Optic Flow Analytic Basis Set - Translation



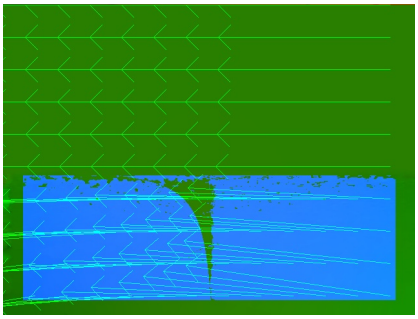
(b) Optic Flow Analytic Basis Set - Rotation



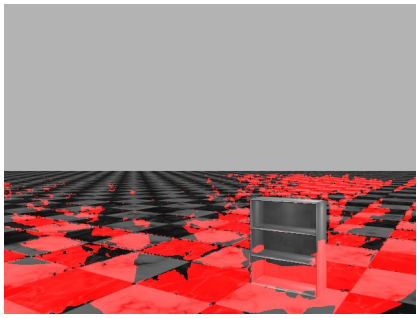
(c) Segmented Ground Plane using Analytic Basis Sets



(d) Optic Flow Learned Basis Set - Translation



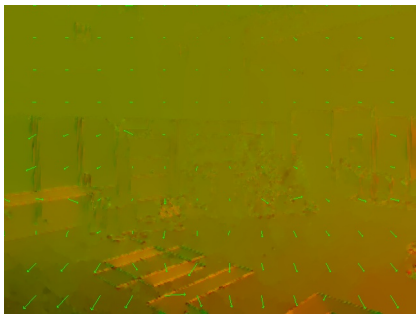
(e) Optic Flow Learned Basis Set - Rotation



(f) Segmented Ground Plane using Learned Basis Sets

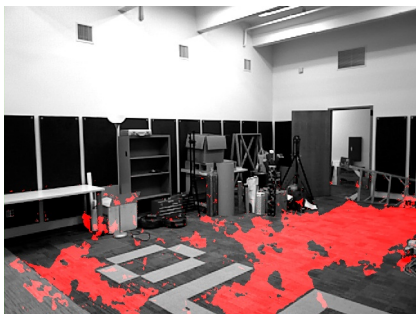
Fig. 4: Results for Simulation

(a) Test environment set-up



(b) Resulting Optic flow

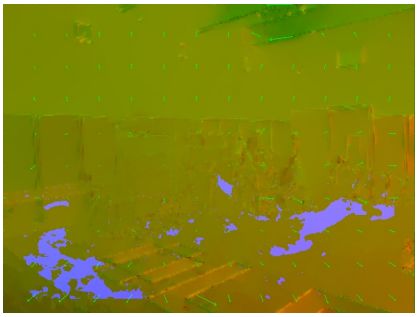
(c) Ground Plane Extracted using Analytic Basis Set



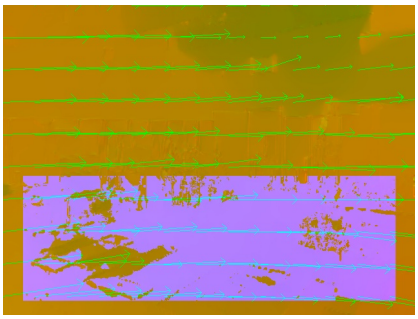
(d) Ground Plane Extracted using Simulated Basis Set

Fig. 5:

(a) Test Environment



(b) Observed Basis Flow - Translation



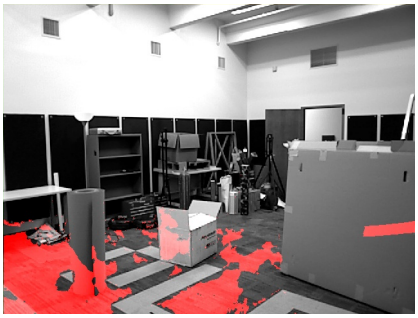
(c) Observed Basis Flow - Rotation

Fig. 6:

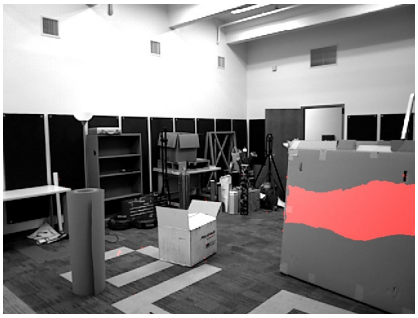
(a) Test Environment

(b) Observed Optic Flow

(c) Extracted Ground Plane - Analytic Basis Set



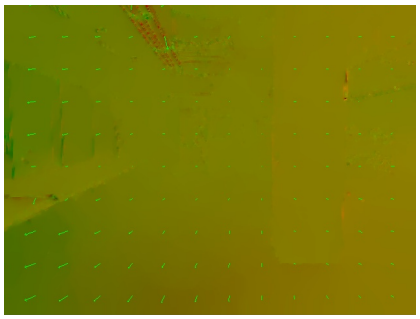
(d) Extracted Ground Plane - Simulated Basis Set



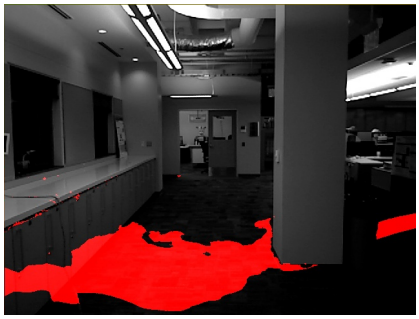
(e) Extracted Ground Plane - Observed Basis Set

Fig. 7:

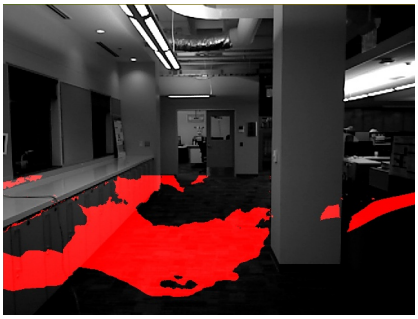
(a) Test Environment B



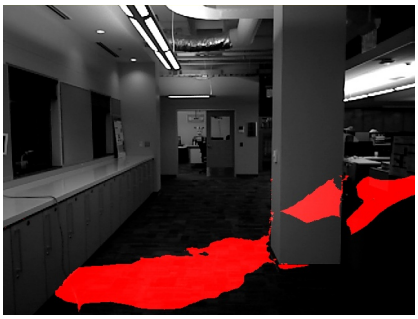
(b) Observed Optic Flow



(c) Extracted Ground Plane - Analytic Basis Set



(d) Extracted Ground Plane - Simulated Basis Set



(e) Extracted Ground Plane - Observed Basis Set

Fig. 8:

V. Discussion

The obtained results indicate that the approach to ground plane segmentation covered in this study achieves limited success. Overall, it is apparent that the basis flows derived from the analytical solution are the most effective, followed by the simulation-based templates, with the empirically-derived flows being the least successful. This is fairly unsurprising, given the platform instabilities present in both the simulated and real cases. Although the basis flows were derived specifically for a given fundamental motion (i.e. translation, rotation), the platform often underwent perturbations which resulted in the violation of key assumptions (such as translational velocity being parallel to the ground plane). This produced coefficients which were affected by such "noise", and were biased accordingly when applied to a given scene. This effect is particularly evident in the derivation of basis flows from a real scene, as in Fig.**??**. Few of the ground plane points belong to the consensus set, and it is not guaranteed that these correspond to a strictly planar motion.

We also note a few interesting things. For example in figure 7 we see that lack of texture in the letters causes the optic flow algorithm to incorrectly estimate the flow in that region. This severely affects the ability to segment that part as the ground plane, resulting in a distinct boundary around this pattern.

In the figure 4 the base of the object is not detected because of its proximity to the ground and the height of the camera. The base has similar optic flow characteristics as the ground plane itself. Hence, obstacle segmentation may be improved by lowering the optical frame, thereby creating a larger distinction in the flow-field for surfaces near the ground. Although this would create much larger motions in the image plane, the optical flow method used incorporates a descriptor-matching method which is claimed to be robust against such large displacements [6]. It should be noted that this would come at the cost of limiting how far the camera can see reliably, although in practice, additional sensors can be mounted for this purpose.

As mentioned above, the camera in the physical system undergoes frequent jerks and vibrations during movement, and lacks any explicitly designed damping mechanism. This shows up in certain optic flow estimates, such as when one of the casters becomes misaligned, and removes any possibility of extracting information from that frame. A shock compensation term could be included in the algorithm, as in [1], to mitigate these effects.

Scenes with poor illumination (such as the right side of Fig.8) make it difficult for the optical flow scheme to register intensity gradients properly.

VI. Future Work

Instead of using real data as training data, we can use data obtained from simulation to do the same. This would make it more robust. Also right now due to lack of time, we couldn't train over an extended training set in different scenarios, hence the system was not very robust to completely new environments, though we did try it with a few. Instead of having just two basis set modelling rotation and translation, we can have multiple basis sets depending upon our prior knowledge of scene geometry for example corridors and walls. Currently, due to the rigidity of the platform, a lot of jerky motion of the robot is transferred over to the camera, this causes issues with optic flow, especially with high frame-rate cameras and the only straightforward solution is to downsample the image frequency which is of course not very appealing because we increase on in reaction time which might be dangerous. What we need to do for this problem is to have a system of shock compensation in place which can add some sluggishness to the response acting like a low pass filter based on our physical system model, similar to a suspension, but in software. There is a lot of information that can be obtained about the scene geometry apart from just the obstacles. We can in theory actually segment out different types of obstacles based on the optic flow patterns it generates. This is incredibly useful in doing a first pass to quickly find objects of interests.

References

[1] Andrea Giachetti, Marco Campani, and Vincent Torre. The use of optical flow for road navigation. *Robotics and Automation, IEEE Transactions on*, 14(1):34–48, 1998.

[2] David J Fleet, Michael J Black, Yaser Yacoob, and Allan D Jepson. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3):171–193, 2000.

[3] Richard Roberts, Christian Potthast, and Frank Dellaert. Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 57–64. IEEE, 2009.

[4] Richard Roberts and Frank Dellaert. Optical flow templates for superpixel labeling in autonomous robot navigation. In *5th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV13)*, 2013.

[5] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer, 2004.

[6] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500–513, 2011.

[7] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[8] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[9] Etienne Memin and Patrick Perez. A multigrid approach for hierarchical motion estimation. In *Computer Vision, 1998. Sixth International Conference on*, pages 933–938. IEEE, 1998.

[10] Michael J Black and P Anandan. Robust dynamic motion estimation over time. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 296–302. IEEE, 1991.

[11] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[12] G. Bradski. *Dr. Dobb's Journal of Software Tools*.