

Autonomous Robot Navigation-A study using Optical Flow and log-polar image representation

Jörg Rett¹ and Jorge Dias

Institute of Systems and Robotics, University of Coimbra - Pole II

Pinhal de Marrocos, 3030-290 Coimbra, PORTUGAL

e-mail: {jrett, jorge}@isr.uc.pt

¹*This work is partially supported by FCT-Fundação para a Ciência e a Tecnologia Grant #12956/2003 to J. Rett*

Keywords: Navigation, Obstacle Detection, Homography, Logpolar, Optical Flow

This paper describes a methodology for autonomous robot navigation, based on log-polar transform of images and optical flow. The navigation task for robots involves the detection of obstacles in the traversable path. This is considered as basic capability for mobility that includes the measurement of the height of objects to classify them as to be avoided or to be ignored. The vanishing point in the image corresponds to the Focus of Expansion (FOE) since we are assuming that the mobile robot moves with a translational velocity parallel to the ground plane. The FOE is determined from the optical flow field using a phase-based approach. From the FOE in the images and assuming the robot moves in the levelled ground, the planar homography H , is recovered and any object on the floor can be detected. In this paper we prove that it is not necessary to recover the homography H explicitly but sufficient to evaluate the displacement of tracked points along epipolar lines in the image. This article describes how these epipolar lines are computed and their relation with the FOE, when the robot moves with translational velocity.

1 Introduction

Robotics field is facing the challenge to develop robots that share an environment with humans. The two basic skills social robots need to have is to interact with the persons and to navigate in the world. To study possible solutions and feasible techniques we started the development of the robot guide *Nicole*. Nicole will guide visitors through the Institute of Systems and Robotics (ISR), talk about the research and react on gestures performed by persons recognized as "god-fathers". The interaction part as well as the navigation part will strongly rely on visual cues. This paper is concerned with the navigation part.

Navigation based on vision needs to segment the traversable path and distinguish it from objects that need to be avoided. The method we propose will solve the problem of obstacle detection. Nicole starts from her initial position facing along the corridor with an initial movement straight along the corridor, while capturing images from the corridor. From the images we compute the optical flow which is further used to initialize the Focus of Expansion (FOE). To get an initial guess for the FOE we also tested an approach using the intersection of the horizon line and lines parallel to the corridor. The benefit of the latter method was that it did not require any initial translational movement.

As Nicole moves with constant speed straight along the corridor we start detecting corners and interesting points in the images. Local features have been shown to be well suited to tracking as they are robust to occlusion, background clutter and other content changes [1]. The main idea is to distinguish between features belonging to the traversable ground plane and features that belong to obstacles to be avoided. We calculate their spatial position from the cartesian coordinates using the initial FOE as the center. The corners are tracked and their trajectory in cartesian coordinates can be used to update the FOE. Their trajectory in spatial coordinates will provide us with the information we are searching for: ground plane or obstacle to avoid.

In the field of feature detection the recent research is aiming on obtaining invariance to viewing conditions like in the work of Mikolajczyk and Schmid [1]. In the field of navigation using vision Liang et al. [2] use a reciprocal-polar representation of images with the origin on the FOE. Assuming a 3D motion parallel to the ground plane the corresponding image motion of a set of co-planar points is a pure shift. The interpretation of the shift signal allows the estimation of the ground plane homography. This estimation used to calculate the affine height of non-ground plane pixels (obstacles).

In section 2 the geometric model is discussed. The section 3 deals with log polar transformation and how its properties are used in this article. Section 4 presents the implementation of the vision based obstacle detection and describes the algorithm for the calculation of the optical flow. Section 5 shows the results and Section 6 closes with a discussion and conclusions.

2 Geometric Models

The task of navigation and avoiding obstacles using vision can be described as the recovery of the spatial layout of a scene to a certain extend. This implies the application of a basic conceptual approach for two view reconstruction as presented in [3]:

1. Compute the fundamental matrix from point correspondences.
2. Compute the camera matrices P from the fundamental matrix.
3. For each point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ compute the point in space \mathbf{X}_i that projects to these two image points.

Due to "noisy" images the implementation of such a reconstruction method lacks in practice of robustness.

Recent approaches [2] renounce the reconstruction of the scene and concentrate on the detection of points on the ground plane as the first step. As shown in [3] there is a planar homography \mathbf{H}_π that maps an homogenous image point \mathbf{x}_1 to an homogenous point \mathbf{x}_2 in the other image.

$$\mathbf{x}_1 = \mathbf{H}_\pi \mathbf{x}_2 \quad (1)$$

Liang et al. further simplified the recovery of \mathbf{H} by creating the multiple views by a pure translational movement. In [4] it is shown the \mathbf{H} matrix can be recovered from the vanishing point, the horizon line and a corresponding point pair.

Our approach sets the same constraint on the movement but goes one step further by renouncing the calculation of the \mathbf{H} matrix explicitly. From Fig. 1 it can be seen that the line passing by the image centers $\{\mathbf{C}_1\}$ and $\{\mathbf{C}_2\}$ usually baseline, intersect the images \mathbf{I}_1 and \mathbf{I}_2 in two points \mathbf{e}_1 and \mathbf{e}_2 named epipoles. These points also represent:

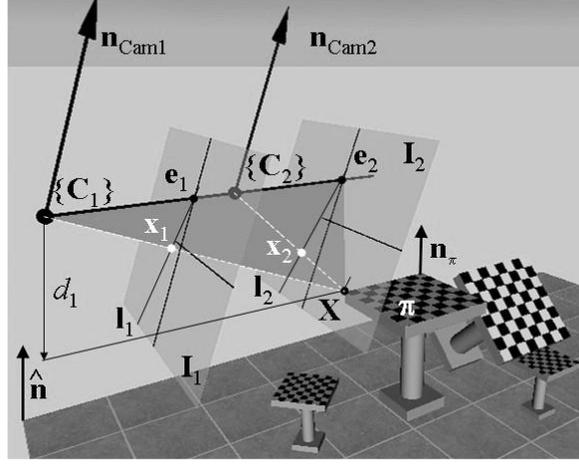


Figure 1: Robot in pure translational motion.

- Vanishing points of the ground plane, since the translational motion is parallel to the ground.
- Vanishing points that lie on a plane π parallel to the ground plane
- The Focus of Expansion (FOE) generated by the translational motion

The baseline, the 3D point \mathbf{X} and its images \mathbf{x}_1 and \mathbf{x}_2 define a plane. This plane contains the epipoles \mathbf{e}_1 and \mathbf{e}_2 and it is named epipolar plane. The intersection of that epipolar plane with the image planes \mathbf{I}_1 and \mathbf{I}_2 define the epipolar lines \mathbf{l}_1 and \mathbf{l}_2 . As the direction of the translation lies on the epipolar plane, the correspondent epipolar lines will coincide in the image plane and:

$$\mathbf{l}_1 = \mathbf{l}_2 = \mathbf{l} \quad (2)$$

Thus, along the trajectory of the robot the correspondent images of the 3D point \mathbf{X} move along the epipolar line \mathbf{l} , as long as the robot maintains a translational motion.

We can express the central projection \mathbf{P}_1 with the (unknown) intrinsic camera calibration matrix \mathbf{K} and the normalized image point \mathbf{x}_1 as $\lambda \mathbf{x}_1 = \mathbf{P}_1 \mathbf{X} = \mathbf{K}[\mathbf{I}|0]\mathbf{X}$.

The plane restriction is $[\mathbf{n}^\top d_1] \cdot \mathbf{X} = 0$ with π being a plane not containing the cameras optical centers and defined by its normal vector \mathbf{n} and perpendicular distance d_1 to the optical center of camera 1 $\{\mathbf{C}_1\}$.

$$\mathbf{X} = \lambda \begin{bmatrix} \mathbf{I} \\ \frac{\mathbf{n}^\top}{d_1} \end{bmatrix} \mathbf{K}^{-1} \cdot \mathbf{x}_1 \quad (3)$$

The central projection \mathbf{P}_2 is $\mathbf{x}_2 = \lambda(x, y, 1)^\top = \mathbf{P}_2 \mathbf{X} = \mathbf{K}[\mathbf{I}|\mathbf{t}]\mathbf{X}$ with the translation vector \mathbf{t} . It follows that the mapping from an image point \mathbf{x}_1 to an image point \mathbf{x}_2 is:

$$\mathbf{x}_2 = \lambda \mathbf{K} \cdot [\mathbf{I} \quad \mathbf{t}] \cdot \begin{bmatrix} \mathbf{I} \\ \frac{\mathbf{n}^\top}{d_1} \end{bmatrix} \cdot \mathbf{K}^{-1} \cdot \mathbf{x}_1 \quad (4)$$

Expressing the movement as with the scalar factor λ :

$$\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{K} \left(\lambda \mathbf{t} \cdot \frac{\mathbf{n}^\top}{d_1} \right) \mathbf{K}^{-1} \cdot \mathbf{x}_1 \quad (5)$$

The projection of the translational vector \mathbf{t} (by multiplication with \mathbf{K}) on the image plane is the epipole \mathbf{e} .

$$\mathbf{x}_2 = \mathbf{x}_1 + \lambda \frac{1}{d_1} \mathbf{e} (\mathbf{n}^\top \mathbf{K}^{-1} \mathbf{x}_1) \quad (6)$$

As the multiplication of \mathbf{n}^\top , \mathbf{K}^{-1} and \mathbf{x}_1 also results in a scalar we invent the scalar λ' and write:

$$\mathbf{x}(\lambda') = \mathbf{x}_1 + \lambda' \frac{1}{d_1} \mathbf{e} \quad (7)$$

The equations describes the movement of a point \mathbf{x} in the image. Starting at point \mathbf{x}_1 it will move along the line defined by \mathbf{x}_1 and the epipole \mathbf{e} . The magnitude of the movement is proportional to the magnitude of the velocity and has a reciprocal relationship to the perpendicular distance d_1 to the optical center of the camera in the first frame.

Fig. 2 shows the movement of \mathbf{x} starting at point \mathbf{x}_1 along the epipolar line \mathbf{l}_2 . As known

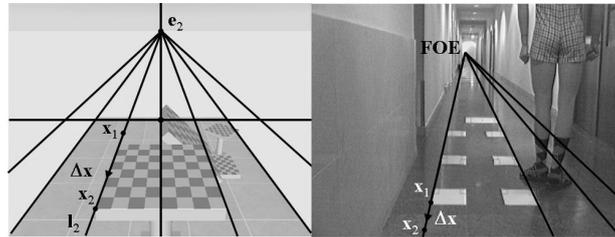


Figure 2: Motion of \mathbf{x} along the epipolar lines. a) in the synthetic image b) in the real sequence

from [3] we have a pencil of epipolar lines radiating from the FOE. As shown in Fig. 2 the images of the 3D points move along the those lines. Now we only need to track the features along the pencil of epipolar lines. We can further speed up the tracking process by transforming the rays to parallel lines. A technique which will transform an image in a way that makes this possible is the Log-Polar transform shown in section 3.

3 Space Variant Methods - the Log-Polar transform

The implementation of visual behaviors in artificial systems is strongly related to the performance of the image analysis algorithms used[5]. To achieve a good performance of the system it is necessary to develop fast algorithms for image processing and control, but exhibiting stability and robustness, to fit the goals of the vision system activity. This section describes algorithms based on the principles of log-polar transformation its properties and potentialities.

One interesting feature of the human visual system is the topological transformation [6][7] of the retinal image into its cortical projection. In our own human vision system, as well as in those of animals, it has been found that the excitation of the cortex can be approximated by a log-polar mapping of the eye's retinal image. In other words, the real world projected in the retinas of our eyes is reconfigured onto the cortex by a process similar to log-polar mapping before it is examined by our brain[7].

In the human visual system the cortical mapping is performed through a space-variant sampling strategy, with the sampling period increasing almost linearly with the distance from the fovea. Within the fovea the sampling period becomes almost constant. This retino-cortical mapping can be described through a transformation from the *retinal plane* (ρ, θ) onto the *cortical plane* $(\log(\rho), \theta)$. This transformation is applied just on the *non-foveal* part of a retinal image.

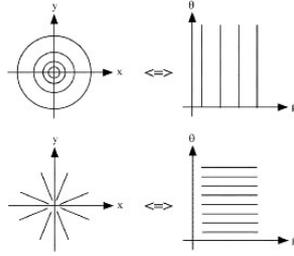


Figure 3: The log-polar mapping applied to regular patterns. Concentric circles are mapped to vertical lines, radial lines to horizontal lines.

3.1 Log-Polar Mapping and its Properties

3.1.1 Log-Polar Mapping

Log-polar mapping can be performed from regular image sensors by using different space-variant sampling structures [8],[9]. The spatial variant geometry of the sampling points is obtained through a tessellation and a sampling grid formed by concentric circles with N_{ang} samples over each circle [5].

3.1.2 Log-Polar Properties

The log-polar mapping has number of important properties that make it useful as a sampling structure. The mapping of two regular patterns as shown in Fig. 3 results in similarly regular patterns in the other domain. From the Fig. 3(a) the concentric circles in the image plane become vertical lines in the *cortical* plane. A single circle maps to a single vertical line since the constant radius r at all angles θ of the circle gives a constant ρ_c coordinate for all θ_c coordinates. Similarly an image of radial lines which have constant angle but variable radius, result in a map of horizontal lines.

These mapping characteristics are fundamental for some properties such as rotation and scaling invariance. Rotation and scaling result in shifts along the θ_c and ρ_c axis, respectively. For rotation invariance notice that all possible angular orientations of a point at given radius will map to the same vertical line.

Scaling invariance is another characteristic of this log-polar mapping. From the Fig. 3(b) we seen that as point moves out from the origin along a radial line, its mapping stays on the same horizontal line moving from the left to the right. The mappings of the concentric circles remain vertical lines and only move horizontally as the circles change in size.

Another property is related with projection of the images when the sensor translates. The images of Fig. 4 show the mapping of the optic flow vectors for different types of translational motion of the sensor. Notice that when the sensor translates in same direction as the optical axis the optical flow generated appears as vectors diverging from the image center. The effect in the cortical plane is a set of lines with vectors with the same orientation, as illustrated in Fig. 4 (a).

Fig. 5 a) shows the centering of the cartesian image to the FOE and the sampling structure formed by 256 samples over each circle and a radius related with the fovea of 2 pixels. Fig. 5 b) shows the Log-Polar image with the epipolar lines superimposed to support the above mentioned properties.

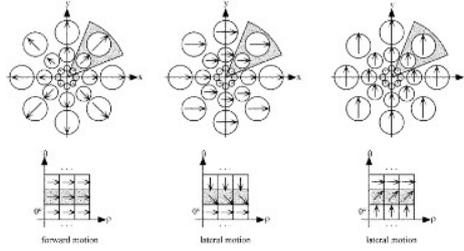


Figure 4: The optical flow vectors for different types of translational motion. (a) Forward motion. (b)+(c) Lateral motion

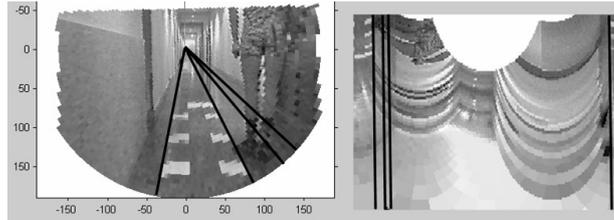


Figure 5: LogPolar transformation of the image. a) Logpolar sampling b) Logpolar representation

4 Vision-based Obstacle Detection

The task of vision-based obstacle detection may be split in those for a (unstructured) outdoor environments and indoor environments. Solutions for the former have been reported by Snorrason et al. [10] using a color-contrast approach and 3D data for NASA's Mars rovers as well as by Lorigo et al. [11] using three independent vision modules based on brightness gradients, RGB (Red, Green, Blue) color, and HSV (Hue, Saturation, Value) color, respectively.

For indoor navigation we can expect a planar surface (ground plane) with obstacles above ground level. Thus, we are able to solve the problem of obstacle detection by detecting features which do not lie on the ground plane.

To detect features above ground plane we will track them along the time while performing a translational movement on the ground plane. As shown in section 2 we are able to distinguish ground plane features and non ground plane features by their trajectory along the epipolar lines. To construct the epipolar lines we need to specify their center, the FOE. We tried two techniques to find the position of the FOE. The first one used the well known Canny edge detector, a line segmentation algorithm and discriminator for those lines parallel to the corridor. In the second approach we used the optical flow field to search for the common intersection of the flow vectors. This technique is described in more detail in the following section.

4.1 Optical Flow Field and FOE

Solving the task to compute 2D component velocities, Fleet and Jepson showed that phase contours are more robust with respect to smooth shading and lighting variations, and more stable with respect to small deviations from image translations [12]. To calculate the optical flow field we used the phase-based approach suggested by Gautama and Van Hulle [13] who made their Matlab implementation publicly available.

(http://www.mathworks.com/matlabcentral/files/2422/optical_flow.tar.gz)

4.1.1 Principle of the algorithm

In their approach the image sequence is spatially filtered using a bank of quadrature pairs of Gabor filters which are characterized by their center frequencies, (f_x, f_y) , and the width of the enveloping (spatial) radially symmetric Gaussian, σ . Using constant bandwidths of β octaves, results in a spatial width of

$$\sigma = \frac{2^\beta + 1}{(2^\beta - 1)2\pi\sqrt{f_x^2 + f_y^2}} \quad (8)$$

and the temporal phase gradient is computed, yielding estimates of the velocity component in directions orthogonal to the filter pairs' orientations. The component velocity v_c is computed by

$$v_c = \text{proj}_{\phi_x^n}(v)\phi_x^n = \frac{-\phi_t(x)}{2\pi(f_x^2 + f_y^2)}(f_x, f_y) \quad (9)$$

where the spatial phase gradient ϕ_x^n denotes the normalized version of vector ϕ_x and $\phi_t(x)$ the temporal phase gradient.

A component velocity is rejected if the corresponding filter pair's phase information is not linear over a given time span. Finally, the remaining component velocities at a single spatial location are combined and a recurrent neural network is used to derive the full velocity.

Gautama and Van Hulle used the estimation of the phase nonlinearity ϵ_l as the confidence measure. Exceeding a certain threshold (nonlinearity criterion τ_l) leads to rejections of the component velocities. The 2D full velocity is determined by several component velocities using a bank of eleven spatial quadrature filter pairs (Gabor). The 2D velocities were computed only if N_{min} valid component velocities were available.

4.1.2 Performance of the algorithm

The performance of the algorithm depends mainly on four parameters:

- The nonlinearity criterion τ_l
- The minimum number of valid component velocities N_{min}
- The time span of the image sequence
- The speed of the image sequence

In the ideal case the algorithm should produce a dense flow field with flow vectors which intersect with the horizon at one point: The Focus of Expansion (FOE). In reality we have intersections all along the horizon line in the image with a center of gravity close to the FOE. We also have a certain number of flow vectors that have no valid intersection with the horizon line within the image. Those will be rejected and not used for the FOE computation.

Starting from the values suggested by Gautama and Van Hulle for τ_l and N_{min} we could certify their results on the density of the flow field. The density increased considerably as the nonlinearity criterion was relaxed or the minimal number of valid component velocities was decreased. On the other hand the rejection rate grew with the number of calculated flow vectors. For densities around 95% we had a rejection rate of 20%. For densities below 50% our rejection rate did decrease in all cases. This suggests that values around 6 and 7 for N_{min} and 0.01 and 0.005 for τ_l are a good choice. They breakdown in performance for the filter bank for speeds higher 3.5 pixels per frame mentioned by Gautama and Van Hulle was also observed in our tests.

4.1.3 Tracking

The tracking of the corner features is done in the Log-polar image. From Fig. 5 it can be seen that features that lie along the epipolar line can be easily found along vertical (or horizontal) scan lines in the Log-polar image. For the algorithm itself we use the normalized cross correlation known from stereo matching. Fig. 6 shows the expected vertical position of the

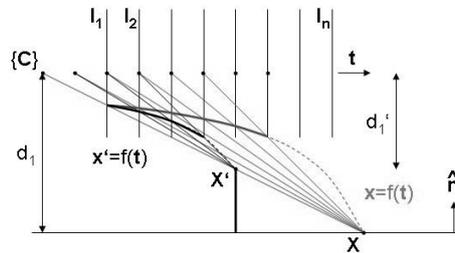


Figure 6: Vertical trajectories for different heights.

tracked features for a translational movement of the camera and two features X, X' with different heights.

4.2 Implementation

The software architecture is shown in Fig. 7. From the first image we calculate an initial guess for the FOE by applying a Edge corner detection, a line segmentation algorithm and finally we discriminate the lines parallel to the corridor. The FOE is the intersection of those lines while the horizon line given by the IMU will put an additional constraint. The further update of the FOE will be done by calculating the optical flow field and the intersection of the flow vectors with the horizon line.

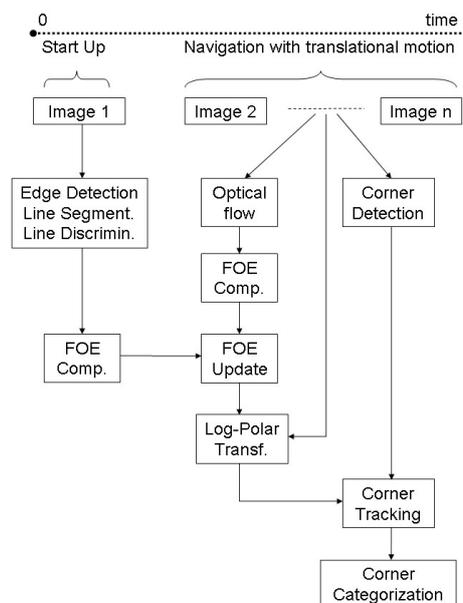


Figure 7: The software architecture.

The detection of our main features, the edges are performed directly on the images and provides us with cartesian coordinates for the edge features. For the tracking of the edges we first transform the image to a log-polar representation using the FOE as the center. We will benefit from the fact that the edges will move along the epipolar lines, thus we only need to search along horizontal scan lines in the log-polar image. The correspondence is performed using the cross-correlation algorithm well known from stereo matching. The output of the tracker gives the trajectory of the features from which we finally discriminate ground and non ground features.

5 Results

Fig. 8 shows the trajectory of some features over a sequence of 15 frames. The corners 1,2,3,4,7 and 8 belong to the ground plane the corners 5 and 6 to the middle and upper part of the socks. In Fig. 8 a) it can be seen that all features follow a trajectory along the epipolar lines intersecting

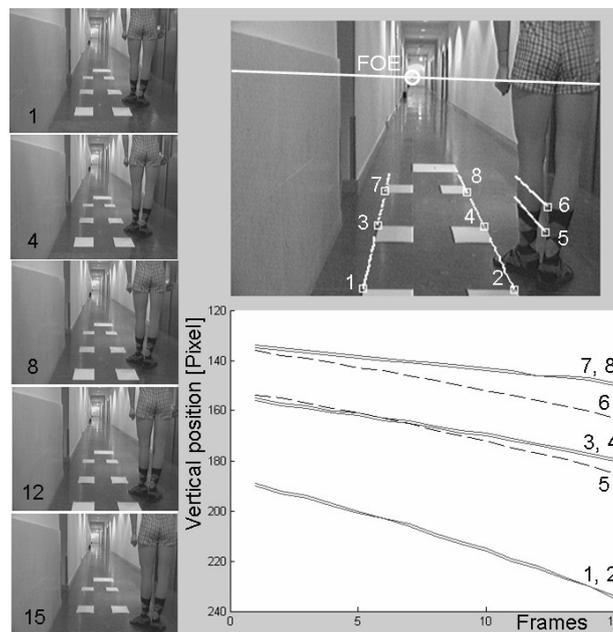


Figure 8: Trajectory of the corner features. a) in the image plane b) along the image sequence

at the FOE. In Fig. 8 b) the movement of the features along the image sequence can be seen. It is obvious that the two features belonging to an obstacle above ground plane (middle and upper part of the socks) follow a trajectory which can easily be distinguished from the ground plane features. As presented in Fig. 6 features above ground plane follow a trajectory with a higher gradient.

6 Discussion and Conclusions

We showed in this article that it is possible to develop an algorithm for obstacle avoidance based on projective geometry. We used the equations for the homography H to distinguish between ground plane features and non ground plane features. We presented an approach without solving the equation explicitly but emphasizing that the features move along the epipolar lines described by equation 7. We used optical flow to get the FOE and assigned the former as the central point of a logpolar transform. We showed that the features follow parallel lines in the logpolar domain which could facilitate feature tracking.

References

- [1] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 1:63–86, 2004.
- [2] Nick Pears Bojian Liang and Zezhi Chen. Affine height landscapes for monocular mobile robot obstacle avoidance. *Proceedings of Intelligent Autonomous Systems*, 8 (IAS-8):863–872, 2004.
- [3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
- [4] B. Liang and N. E. Pears. Ground plane segmentation from multiple visual cues. *2nd International Conference on Image and Graphics, Hefei, China, Proc SPIE*, 4875:822–829, 2002.
- [5] J. Dias, H. Arajo, C. Paredes, and J. Batista. Optical normal flow estimation on log-polar images. a solution for real-time binocular vision. *Real-Time Imaging Journal*, 3:213–228, 1997.
- [6] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *Computer Vision, Graphics and Image Processing*, 14:365–372, 1980.
- [7] E. L. Schwartz. Anatomical and physiological correlates of visual computation from striate to infero-temporal cortex. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-14 (2):257–271, 1984.
- [8] C. F. R. Weiman. Exponential sensor array geometry and simulation. In *Digital and Optical Shape Representation and Pattern Recognition*, SPIE Proceedings Vol. 938. Edited by Richard D. Juday. Bellingham, WA: Society for Photo-Optical Instrumentation Engineers, 1988., p.129, pages 129–+, January 1988.
- [9] L. Massone, G. Sandini, and V. Tagliasco. Form-invariant topological mapping strategy for 2d shape recognition. *CVGIP*, 30:169–188, 1985.
- [10] . Snorrason, J. Norris, and P. Backes. Vision based obstacle detection and path planning for planetary rovers. In *Proceedings of SPIE Vol. 3693, 13th Annual AeroSense conference, Orlando, FL*, 1999.
- [11] L.M. Lorigo, R.A. Brooks, and W.E.L. Grimson. Visually-guided obstacle avoidance in unstructured environments. In *Proceedings of IROS '97, Grenoble, France*, pages 373–379, 1997.
- [12] D J Fleet and A D Jepson. Computation of component image velocity from local phase information. *IJCV*, 5:77–105, 1990.
- [13] T. Gautama and M.M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural Networks*, 13(5):1127–1136, 2002.