Visual Odometry with Omni-directional images

Gabriel Leivas Oliveira

***Abstract*— We describe a method for visual odometry, using optical flow, with a single omni-directional (catadioptric) camera. We show how omni-directional images can be used to perform optical flow, discussing the basis of optical flow and some restrictions needed to it and how unwrap these images, in special how to unwrap omni-directional to a specific kind of view called *Bird's eye view*, that correspond to scaled orthographic views of the ground plane.**

Omni-directional images facilitate landmark based odometry, since landmarks remain visible in all images, as opposed to a small field-of-view standard camera. Also, omni-directional images provide the means of having adequate representations to support accurrate odometry.

Tests were performed to measure robustness and performance of our approach with analysis of the data acquired.

I. Introduction

Visual sensors are usually low cost sensors suitable to navigation in structured environments. Particularly, Omni-directional cameras are efficient sensors because of panoramic view provided by a single image, see figure 1. Hence, it is intended using a optical flow method in rectified images, in order to obtain a odometry estimation of the robot navigation.

Fig. 1. Example of Omni-directional image.

All authors are from Minas Gerais Federal University - UFMG, Belo Horizonte, MG, Brazil gabrielleivas@gmail.com

Our proposed system can be seen on figure 2. First the *pre-procesing* step containing acquisition of the images by the Omni-directional and camera calibration process. Acquired images pass though a unwrapping process and following to extraction of features, these features feed the optical flow method that estimate motion between consecutive pictures and passing for a odometry process where the motion will be transformed to odometry of the robot and return to unwrapping. These last four steps are defined like *Main loop*.

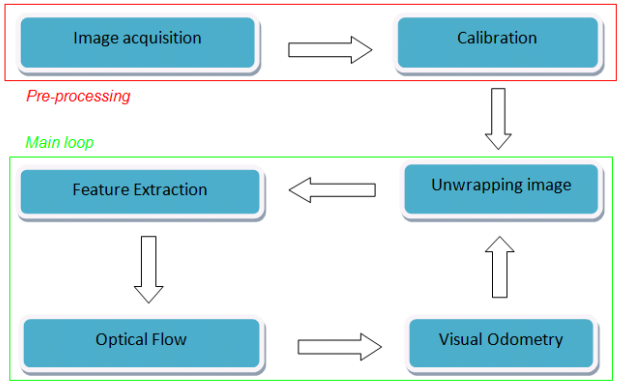

Fig. 2. Overview of the system proposed.

This paper is organised as follows. Section II summarises relevant papers literature in visual odometry and optical flow with Omni-directional and discusses how it is incorporated into some of the proposed techniques. Section III shows the camera calibration used. Section IV describes the fundamental principles behind Bird's eye view unwrapping process. Section V detailed optical flow methods and mathematical basis and proposes an approach that incorporate optical flow to implement visual odometry. Section VI presents experimental results for real data sets which demonstrates the effectiveness of the proposed approach. Section VII discusses the principle conclusions and future works.

II. Related works

A major problem of robotic navigation based on vision systems is to get the match in images taken at different points of view. In the literature, feature matching are extensively researched in standard cameras [7] [15]. These methods have been successful in perspective camera, but such methods can not be applied directly to images obtained from omni-directional systems, because of the nonlinear distortions introduced by the wide field of view (eg, decreased radial image resolution).

To apply traditional methods in Omni-directional images is necessary to unwrap this pictures, removing distortions of Omni-directional images [12].

[13], proposes a method for visual odometry using a new method for removing outliers in the matching process in non-rectified images.

The present work incorporate Bird's eye view unwrap images method, that correspond to scaled orthographic views of the ground plane [5].

In the case of visual odometry methods using Omni-directional cameras, there is one work of [3], which has developed two methods for visual odometry for a planetary Rover, the first using optical flow and then using iterative Kalman filter. The work of [4] proposes an adaptation of optical flow for Omni-directional images without rectification, with neighborhood adaptation to solve the problem of precision loss in furthest regions from the center. The work [16] comes with a visual odometry system of multiple cameras and 3D mapping.

For calibrate the camera used in our test, the method proposed by [11] and toolbox [10] was used.

The present work propose use o optical flow with unwrapped images to perform robotic visual odometry.

III. Calibration

First of all we need know our intrinsic and extrinsic camera parameters to unwrap images. For that was found in literature the method proposed by [11]. Figure 3 illustrate one image used on calibration of a set of seventeen images that were employed.

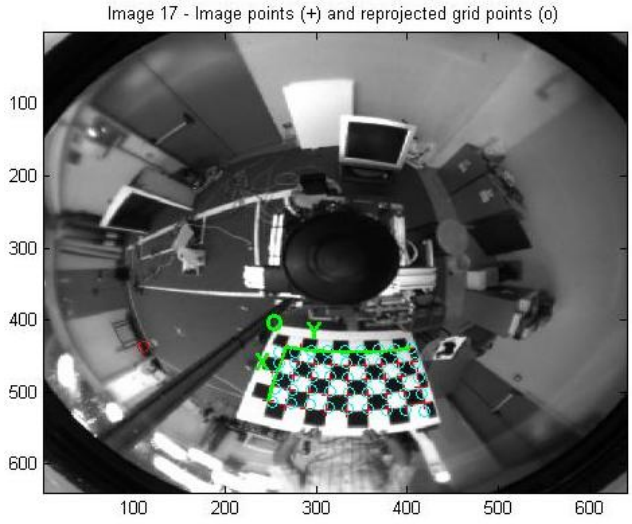

Fig. 3. Image used on calibration.

Knowing all parameters we pass to the Bird's Eye View unwrapping process.

IV. Bird's Eye View unwrapping

According to [5], points in the 3D space, **P**, are projected as image points, p, by means of a projection operator,P:

$$p = \mathrm{P}(\mathbf{P}, \Theta) \tag{1}$$

where Θ contains all the intrinsic and extrinsic parameters of the catadioptric panoramic camera:

$$\Theta = [L \ f \ \mu_0 \ \nu_0]^T \tag{2}$$

The mirror radius can be measured easily, but the camera-mirror distance, L, focal length, f and principal point, (μ_0, ν_0), can only be determined up to some error:

$$\delta\Theta = [\delta L \ \delta f \ \delta\mu_0 \ \delta\nu_0]^T \tag{3}$$

To estimate $\delta\Theta$ we use a set of known 3D points, P^i, and the corresponding image projections p^i, then minimize the following cost function:

$$\delta\Theta = \arg\min_{\delta\Theta} \sum_i \left\| p^i - \mathbf{P}\left(P^i, \theta_0 + \delta\Theta\right) \right\|^2 \tag{4}$$

This procedure defines a mapping between radial distances measured in the ground plane and the respective image coordinates, and can be efficiently implemented by means of a look-up table. It allows us to unwarp omnidirectional images to bird's eye view - see Figures 4 and 5.

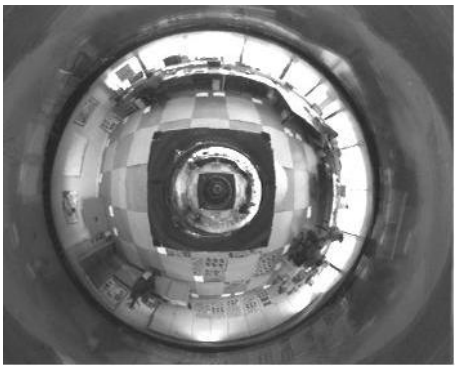

Fig. 4. Original image.

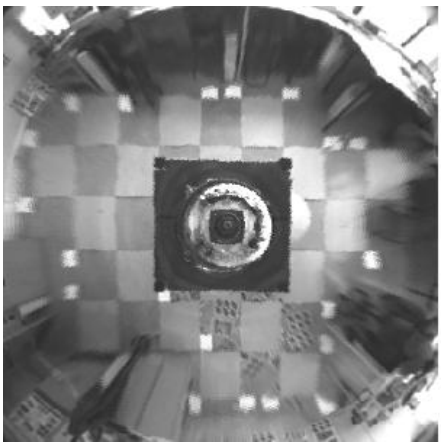

Fig. 5. Unwrapped image.

V. Optical Flow

The methods for Optical Flow computing can be categorize into three major groups major: differential techniques, matching Techniques and frequency energy. The initial hypothesis of optical flow differential techniques is that the intensity between different frames in a sequence of images is approximately constant in a small time period, in other words, in a short time the offset is minimum.

Be $I(x, y)$ the image intensity at time t, it is first assumed that the time interval dt between two images is very short and the image intensity does not change this time interval.

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{5}$$

The above equation can be expanded by the Taylor series and rewritten as:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + o^2 \tag{6}$$

Clumping the two equations by eliminating o^2, which are the higher order terms, we reach in:

$$0 = \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + o^2 \tag{7}$$

where $\bar{V} = (\frac{dx}{dt}, \frac{dy}{dt})$ are the components of velocity vector V looked for. The gradient of the image function in x and y, $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ is called ∇I.

The equation of constraint optical flow becomes:

$$\nabla I.\bar{v} + I_t \tag{8}$$

Only the constraint equation above eq.8 is not sufficient to estimate the components of V, because there are more unknowns than equations to solve.

The process of determining the optical flow is complex, because it involves many variables, not always manageable: occlusions points of interest due to change of perspective and due to appearance of new objects in the scene, one aspect very important when dealing with Optical Flow in real situations is the vibration of the camera and change lighting, shadows and clouds which modify the intensity of the images.

A. Lucas-Kanade Method

Using Optical Flow equation 8, we see that a direct resolution lack of equations for the number of unknowns. This is known as the *problem of openness* in optical flow algorithms. To find the optical flow another set of equations is necessary [8], given by another constraint. The solution given by Lucas and Kanade is an not iterative method that takes a constant local optical flow. Assuming that the flow (Vx, Vy, Vz) is constant in small windows mxm which m sizes with $m > 1$, which is centered in these windows and numbering the pixels $1, ..., n$, a set equations can be found:

$$I_{x1}V_x + I_{y1}V_y + I_{z1}V_z = -It_1 \tag{9}$$
$$I_{x2}V_x + I_{y2}V_y + I_{z2}V_z = -It_2 \tag{10}$$
$$I_{x3}V_x + I_{y3}V_y + I_{z3}V_z = -It_3 \tag{11}$$
$$I_{xn}V_x + I_{yn}V_y + I_{zn}V_z = -It_n \tag{12}$$

With this restriction, there are more equations than variables and then the system becomes over determined, accordingly.

$$\begin{bmatrix} I_{x_1} & I_{y_1} & I_{z_1} \\ I_{x_2} & I_{y_2} & I_{z_2} \\ \vdots & \vdots & \vdots \\ I_{x_n} & I_{y_n} & I_{z_n} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} -I_{t_1} \\ -I_{t_2} \\ \vdots \\ -I_{t_n} \end{bmatrix}$$

that can be summarize in eq. 13

$$A\bar{v} = -b \tag{13}$$

To solve a over determined system of equations was chosen the least squares method:

$$A^T A\bar{v} = A^T(-b) \tag{14}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} \sum I_{x_i}^2 & \sum I_{x_i}I_{y_i} & \sum I_{x_i}I_{z_i} \\ \sum I_{x_i}I_{y_i} & \sum I_{y_i}^2 & \sum I_{y_i}I_{z_i} \\ \sum I_{x_i}I_{z_i} & \sum I_{y_i}I_{z_i} & \sum I_{z_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{x_i}I_{t_i} \\ -\sum I_{y_i}I_{t_i} \\ -\sum I_{z_i}I_{t_i} \end{bmatrix}$$

With the sum ranging from $i = 1ton$. This means that the optical flow can be found by the derived from the image in all four dimensions.

B. Shi-Tomasi Feature Tracker

No feature-based vision system can work unless good features can be identified and tracked from frame to frame. Although tracking itself is by and large a solved problem, selecting features that can be tracked well and correspond to physical points in the world is still hard [14].

Shi-Tomasi calculates the following matrix:

$$\begin{bmatrix} \sum (\frac{\partial I}{\partial x})^2 & \sum (\frac{\partial^2 I}{\partial x}) \\ \sum (\frac{\partial^2 I}{\partial x}) & \sum (\frac{\partial I}{\partial y})^2 \end{bmatrix} \tag{15}$$

where I is the intensity of the pixel, ∂x and ∂y and are the horizontal and vertical displacements of the center of the window containing the neighborhood.

Then [14] define that a *good feature* should have two distinctive qualities, texturedness and corner. When we face

lack of texture, it brings ambiguities in tracking, because of pixels became much more equivalent resulting in outliers and false matchings, on fig. 6 we see low texture region and fig. 7 high texture region . Another problem is images with a reduced number of corners that is described like *aperture problem*.

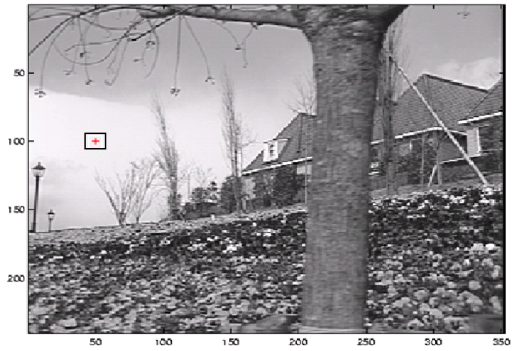

Fig. 6. Low texture region, gradients with small magnitude, small eigenvalue.

Fig. 7. High texture region, gradients with large magnitude, high eigenvalue.

A good feature has a high eigenvalue, which suggest reliable results.

VI. Tests and Results

Due to the need of validation of the proposed system, a set of databases with robotic navigation used as test pattern was searched [2] [17] [9]. Among the research, was choosen an dataset provided by rawseeds project [9]. The available data count with omni-directional camera, whell odometer and a GPS (used as reference), figure 8 show vehicle used to obtain the dataset.

Tests were also conducted with the omni-directional system that we have available, the test was assembled in the Pioneer P3-AT robot and tests were done navigating through corridors of the computer science department, although the data colleted could not be used, considering that the unwrapping process has a constraint on the image resolution of at least 640 lines. Therefore, the data could not be used, since

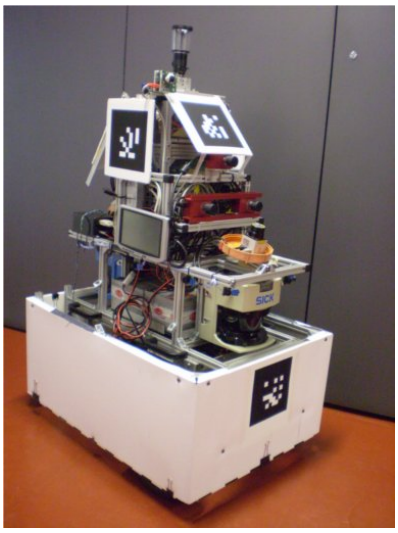

Fig. 8. Vehicle employed for grabbing the dataset.

the camera used, a *point-gray dragonfly* has a maximum resolution of 640x480 and the other accessible cameras, for example, Sony DFW-X700 with 1024x768 resolution, have an C type assembly(17.52 mm) and is incompatible with the omni-directional lens(Remote Reality A1168 Netvision 360) available, which has a CS mount type, that are cameras with focal flange distance of 12.52mm.

The lens used was an omni-VS-C15MR-Vstone with hyperbolic mirror, see figure 9 with a Prosilica GC1020C camera, whole acquisition system(lens+Camera) can be seen on figure 10.

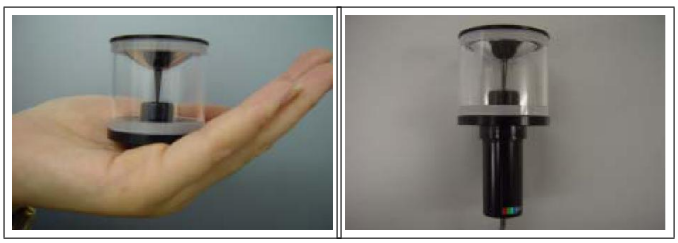

Fig. 9. Lens.

Fig. 10. Acquisition system.

The computer used was a Core i7 920(2.67 Ghz) with 12 Gb of RAM, 320 Gb Hd and a GTX 260 graphic card.

A. Performance Results

First we test the time spend by the both main modules of the proposed system, see figure 11. This figure shows clearly that the unwrapping process spend more time than perform only the optical flow, the overhead add by this routine can be seen on table I.

B. Optical Flow Results

In relation to optical flow figure 12, illustrates the optical flow performed in our tests. Figure 13 shown the number matchings, we have no more than 400 features extracted

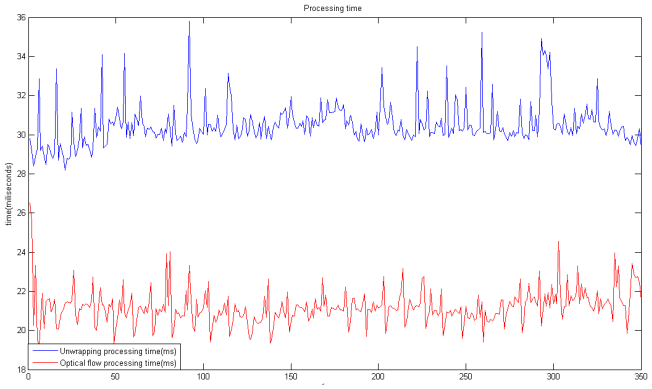

Fig. 11. Processing time of main modules.

TABLE I
STATISTICAL VALUES OF MODULES AND COMPLETE SOLUTION.

Module	Mean(miliseconds)	Standard deviation	Fps
Optical Flow	21	0.9	47
Unwrapping	30	1.1	33
Solution	50	1.39	20

per frame this number set by user. The average number of correspondences in Figure x is 44, which generates a value of 11% in the number of extracted features.

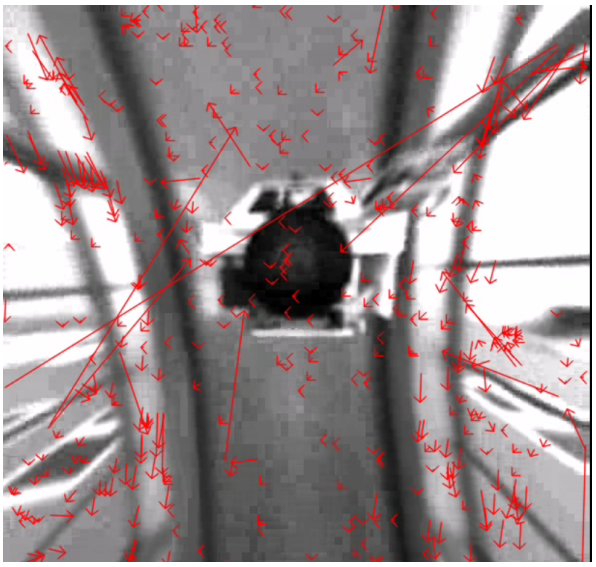

Fig. 12. Optical Flow example.

C. Visual Odometry Results

The first test was performed in a 3.35 meters route moving in straight line. Tests were executed with people passing by(which disturb optical flow) and artificial and natural lighting. Figure 14 on vertical axis are represented odometry values of the robot taking into account his move in x and y axis and ilustrating at the red line(y axis) his final position(3.10 meters), which compare to groundtruth of 3.35 meters get an error in position of about 7.4%.

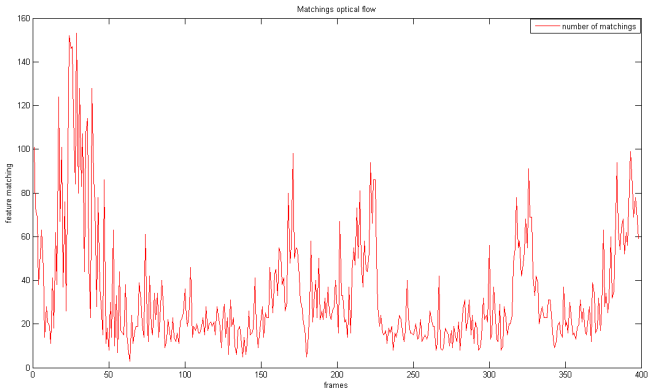

Fig. 13. Number of matchings.

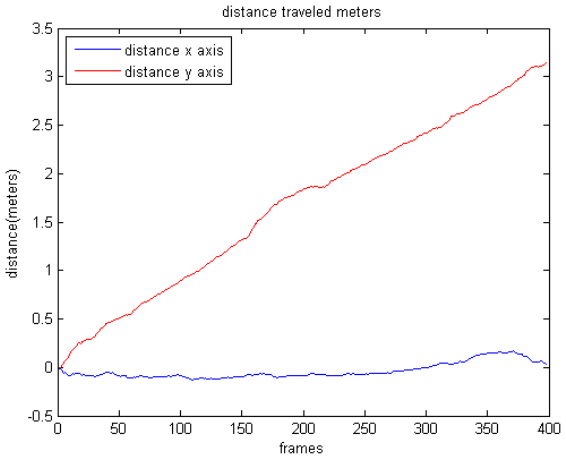

Fig. 14. Results of first test.

The second test was performed in a 7.55 meters route moving in straight line. The second test were executed in a corridor with a uniform structured(low texture in some parts only a white wall), people passing by(which disturb optical flow) and artificial lighting. Figure 15 on vertical axis are represented odometry values of the robot taking into account his move in x and y axis and ilustrating at the red line(y axis) his final position(6.64 meters), which compare to groundtruth of 7.55 meters get an error in position of about 12%.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we presented a method for visual odometry using an omni-directional camera. One of core observations is that wide range cameras have positive aspects that a standard one do not, a landmark tracked on catadioptric images remain observable for a longer time and make visual odometry approaches based on then more robust. In particular, we described a method for obtaining a bird's eye view of the ground floor, that greatly simplified omni-directional optical flow strategies, by removing perspective effects.

Experiments of the proposed approach were presented.

Finally we saw that visual odometry is an important and powerful tool to estimate the motion in image sequences,

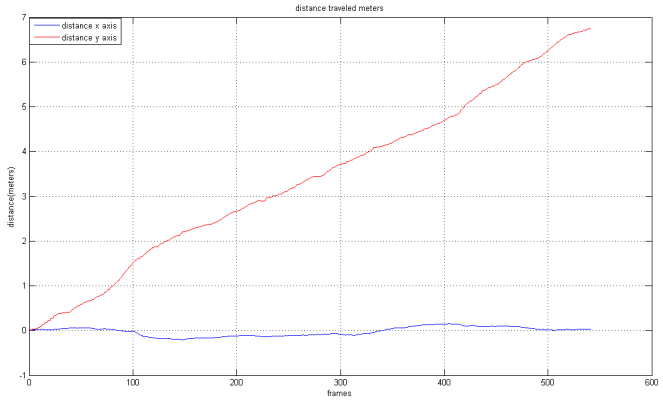

Fig. 15. Results of second test.

many studies are conducted each day to its improvement and optimization computational seeking answers in real time. This research area is vast and still rising.

In the future, we will apply this methodology in more complex environments and test other approaches to solve visual odometry, like Appearance-based odometry [6], more specific methods to feature extraction on omni-directional images [1] and fix the unwrapping resolution limitation.

REFERENCES

[1] J. Barreto and M. Lourenço. Feature detection and matching in images with radial distortion. *IEEE International Conference on Robotics and Automation (ICRA 2010)*, 2010.

[2] F. Blanco, J. Moreno and J. Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 2009.

[3] P. Corke and D. Strelow. Omnidirectional visual odometry for a planetary rover. *Ian 2004*, 2004.

[4] C. Demonceaux and M. Rziza. An adapted lucas-kanade's method for optical flow estimation in catadioptric images. *8th Workshop on Omnidirectional Vision,*, 2008.

[5] J. Gaspar and J. Santos-Victor. Omni-directional vision for robot navigation. *IEEE Workshop on Omnidirectional Vision (in conjuntion with CVPR 2000)*, pages 21–28, 2000.

[6] G. Leivas, S. Botelho and P. Drews. Appearance-based odometry and mapping with feature descriptors for underwater robots. *Journal of the Brazilian Computer Society*, pages v.15 p.47–54, 2009.

[7] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[8] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.

[9] Rawseeds Project. Rawseeds datasets. *http://www.rawseeds.org*, Rawseeds Project, 2010.

[10] D. Scaramuzza and R. Siegwart. Ocamcalib toolbox: Omnidirectional camera calibration toolbox for matlab . *Google for -ocamcalib*, 2006.

[11] D. Scaramuzza and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2006)*, 2006.

[12] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE TRANSACTIONS ON ROBOTICS*, 2008.

[13] D. Scaramuzza and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. *IEEE International Conference on Robotics and Automation (ICRA 2009)*, 2009.

[14] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, pages 593–600, 1994.

[15] Tuytelaars T. and Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[16] J. P. Tardif. Monocular visual odometry in urban environments using an omnidirectional camera. *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2008)*, 2008.

[17] C. Wang and D. Duggins. Navlab slammot datasets. *www.csie.ntu.edu.tw/˜bobwang/datasets.html*, 2004.