Visual Measurement Estimation for Autonomous Vehicle Navigation

Mario F. M. Campos and Luiz Chaimowicz

Laboratório de Robótica, Visão Computacional e Percepção Ativa
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, MG 31270-010, Brazil

ABSTRACT

The autonomous navigation of a mobile vehicle can be described as the task it undertakes to move itself in the environment through a series of positions based on information gathered by its sensors. In order to accomplish this task, the vehicle has to cope with two main subtasks namely obstacle avoidance and self localization. The later implies in the ability to determine its position and orientation with respect to the environment. This work describes a simple but efficient method that performs pose estimation for a mobile vehicle based on visual information from artificial landmarks using a sequence of frames from an uncalibrated camera. The landmark is segmented from image sequences and the vehicle's localization is computed using landmark geometric properties and vehicle's motion vector. This methodology can be easily extended to be used by different types of mobile agents. One of the key advantages is that it is computationally efficient making it suitable for real time navigation. Experiments conducted with a Nomad 200 mobile robot equipped with a color camera system have shown the method to be repeatable and very robust to noise. Visual measurements were compared with readings from other onboard sensors such as ultrasound with excellent consistency.

Keywords: Visual based distance estimation, landmarks, mobile robots

1. INTRODUCTION

The navigation of an autonomous vehicle, more specifically a mobile robot, can be defined in a very simple form as the task of moving it from a certain place to another (the goal) in its workspace. This has to be done in an autonomous form, i.e., without the intervention of a human operator. To accomplish this task, the robot has to deal with two main subtasks that are obstacle avoidance and the identification of its position in the environment or, more specifically, the ability to know if the goal was reached. The complexity of the navigation task will depend on a series of factors[1]: the size of the robot, its kinematics, its computing power and the kind of its sensors. The environment can be known or unknown, with static or moving objects, with a flat or rough terrain, indoors or outdoors. The navigation task itself may have time constrains, best path criteria, and so on. Therefore, several different approaches for this problem exist.

There has been a long discussion if only a global motion planning is sufficient to navigate a robot. Planning can be defined as determining how to act in order to satisfy certain goals based upon predictions of possible future states.[2] The problem with motion planning is that it is hard to predict accurately the future states. Even if the environment is static and completely known, the actions (moves) executed by the robot are not "perfect". A command to move the robot ten inches in the forward direction will move it approximately ten inches close to the forward direction. The sum of these little errors leads to uncertainty that can prevent the robot to reach the desired goal. So, it is clear that in order to navigate in an autonomous way, the robot has to sense his environment, plan and control its actions. The navigation is an instance of the general robotic task paradigm: "perceive-decide-act".[1]

Basically, two kinds of sensors are more frequently used to acquire information in the "perceive part" of the navigation loop: ultrasound sensors (sonar) and visual based sensors (cameras). There are a lot of studies in both of these fields, and this work is focused on visual based navigation. It is known that the visual perception plays a fundamental role in humans, animals and insects.[3] It allows us to identify things, navigate, estimate distances and

Correspondence: E-mail: mario@dcc.ufmg.br; WWW:http://www.dcc.ufmg.br/~mario; Telephone:+55 31 499-5860; Fax:+55 31 499-5858

many other activities that would not be possible with the sole use of other senses. So, the use of visual based sensors in a navigational task seems to be a good choice.

One way of use visual sensors is to build a 3D representation of the environment.[4] The idea is to construct a mathematical model of the environment from a series of images taken from different locations. This model allows the determination of the robot position and the definition of a trajectory to be followed. The problem with this kind of method is that the 3D model construction is not an easy task because it requires the capture and mathematical processing of several images. Normally, this requires a great computational effort preventing real time navigation. Instead of use several images to construct the 3D representation, a more efficient approach is try to extract from the images only the information that is necessary to execute specific tasks. Several types of information can be used separately, depending on the task to be realized. This approach is more direct and simple, allowing the extracted information to be used directly on a visual closed-loop control of the robot.[5]

One type of information that can be extracted from an image sequence without the need of 3d reconstruction is the optical flow, the apparent velocity of moving objects in the image. Optical flow can be defined as the apparent motion of brightness patterns observed when the camera is moving relative to the objects being imaged.[6] It is an approximation of the motion field, that is the actual velocity vector (magnitude and direction) of each point of the image. The optical flow is not always equal to the motion field. If the illumination changes but the objects do not move, the optical flow will show the change (because the image intensity is changing), but the motion field will be zero (because there is no movement). The inverse situation also happens: when the object is moving but the brightness pattern does not change, the optical flow can be zero but not the motion field. There are two main types of techniques for optical flow computation[7]: the differential based techniques and the feature correspondence techniques. The former is based on the spatio-temporal derivatives of the image intensity function and is suited for small motions in the image. When the motion is large, the second type is better because it tries to match features (points, edges, etc.) that move in a pair of images. In the work of Barron *et al.*[8] there is a comparative study of various techniques of optical flow computation. It describes and implements nine techniques and show comparative results of the execution with several image sequences. The optical flow analysis is very important because several other information can be obtained from it, such as the motion direction, the focus of expansion and time to collision.

Unfortunately, the computation of the optical flow can be very time consumable, depending on the image size and other characteristics. A simpler information that can be extracted from visual sensors is the presence of some landmark in the image. Normally, landmarks are objects or image patterns that can be identified in the visual field of the camera allowing the robot to localize and orient itself, detect obstacles, and navigate in the direction of the goal. There are basically two kinds of landmarks: the first are natural landmarks, things that already exist on the environment like geographical accidents, trees (outdoors) or chairs, tables, walls (indoors). The second type are artificial landmarks, built on the environment with a specific objective. Natural landmarks require less effort in preparing the environment for robot navigation, but they are harder to segment, identify, and gather information from. On the other hand, artificial landmarks make these tasks easier to implement because they can be built in specific forms, allowing the algorithms to extract lots of information from one single landmark. But the construction of an environment with artificial landmarks may not be simple, and the robot will be restricted to navigate in this environment. One example of use of artificial landmarks can be found in Becker *et al.*[9] It shows how artificial landmarks can be used to reduce the uncertainty of the navigation with a low environment engineering cost. Other example is the work of Taylor and Kriegman.[10] They put bar code landmarks in several places of an unknown environment so that the robot can perform an exploration of the environment going from one visible landmark to another. The robot can do this without accurately determining its absolute position in a global frame of reference. Another interesting approach using landmarks is the work of Huttenlocher *et al.*[11] They use real objects as landmarks (chairs, tables, sofas) and navigate a robot using the changes in size and location of the objects observed when the robot is moving in their direction. This changes are detected matching intensity edges in successive frames and are used to compute the position and orientation of the robot with respect to the landmarks.

The present work shows a very simple method that uses a monocular uncalibrated camera to estimate the distance from a mobile robot to an artificial landmark. The method uses a similar approach from Huttenlocher *et al.*,[11] but in a more simple fashion. Instead of matching intensity edges from objects in two consecutive frames to compute their sizes, this method simply segments an artificial landmark and measure its width, counting the pixels directly from the image. Using the width of the landmark in two consecutive frames and the camera motion when the robot is approaching the landmark, it is possible to estimate the distance between them. The simplicity and robustness of this method allows it to be used on simple tasks, or as a initial step for a more elaborated method for real time navigation. The remainder of this paper is structured as follows. In next section the method to estimate the distances is explained in details. Section 3 shows the experiments realized and discuss the results. Finally, section 4 presents the conclusions and the possibilities for future work.

2. THE METHODOLOGY

When a camera is approaching from some object, the size of the object image in the focal plane of the camera increases proportionally to the inverse of the distance to the object. This is the central idea of this method: to use the variation of the size of a landmark and the camera motion to estimate the distance between them.

The method works as follows: the camera is mounted on a mobile robot. When it is approaching directly the landmark, the distance is estimated using the landmark width (in pixels) in two consecutive frames and the motion between the frames. In the first frame the size of the landmark is smaller than in the second image. The increment in the size is inversely proportional to its distance to the camera. Figure 1 shows a geometry schema of a pinhole camera approaching a landmark: f is the focus of the camera, W is the width of the landmark, w and w' are the widths projected on the focal plane in the two consecutive frames, m is the motion between the frames and d is the distance to be computed.

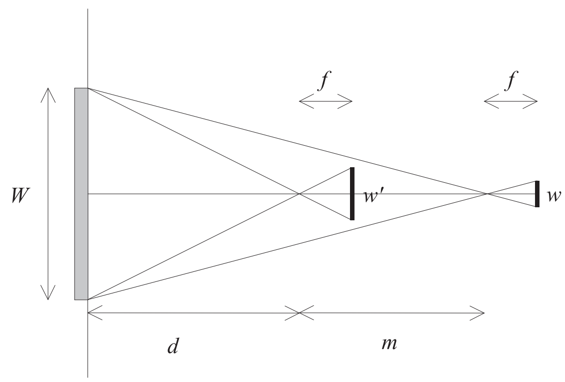


Figure 1. Geometry schema of distance estimation.

From the geometry there are two project equations:

$$\frac{f}{m+d} = \frac{w}{W} \tag{1}$$

$$\frac{f}{d} = \frac{w'}{W} \tag{2}$$

Joining equations 1 and 2 and isolating d, leads to the equation 3, that measures the distance:

$$d = \frac{m}{\frac{w'}{w} - 1} \tag{3}$$

So the distance is directly proportional to the motion and inversely proportional to the ratios of the landmark widths in the image frames. The focus (f) is not present in the equation 3, showing that the camera does not need to be calibrated in order to estimate the distance.

The main restriction of this method is that the robot movement must be approximately in the straight direction of the landmark, otherwise, the formulation showed in figure 1 may not be valid, and other terms may have to be included in the equations. To avoid this problem, it is possible to estimate the bearing and correct the trajectory if it is necessary,[11] but in order to make this method as simple as possible, the correction was not included. Actually, the experiments showed that the estimated distance is very reasonable, even if the robot is not moving exactly in the straight direction of the landmark.

The estimated distance is used to control the movement of a robot in the direction of the landmark. Initially, the robot captures an image, moves five inches in forward direction and captures a new image. The distance is computed using the widths, and this process is repeated until the calculated distance be less than ten inches. In the basic algorithm, shown below, the robot stops its movement to capture the images, but this can be easily extended to be performed in parallel, allowing a continuous movement. It is important to say that, if the landmark is not identified in the image, the robot stops its movement. This can occur, for example, if the movement is not in the direction of the landmark, or some obstacle appear between them.

```
While Distance > 10 do begin
   Capture Image
   Apply Threshold
   Segment Landmark
   Calculate Landmark Width
   Compute Distance
   Move 5 Inches
end
```

The image processing part of this algorithm is very simple. The color image captured by the camera is converted to gray level, a threshold operation is applied and the landmark is segmented. The threshold value is fixed, since there is a good contrast between the landmark and the background. The segmentation is used to isolate the landmark, allowing the computation of its width. After that, the width is used in the method explained above for the distance estimation. Next section shows and discusses the results of the execution of this algorithm, comparing with measurements obtained from an ultrasound sensor.

3. EXPERIMENTAL RESULTS

The test bed to the algorithm was a Nomad 200 mobile robot, from Nomadic Technologies, shown in Figure 2. The Nomad 200 is driven by a synchro-drive mechanism and uses one motor to drive all of the wells (three) and a second motor to steer all of the wells. Its turret can rotate independently from the base, which is an important feature when performing directional tasks such as direct sensing. Nomad 200 is equipped with various sensors such as a visual system, sixteen ultrasound sensors, sixteen infrared sensors, a compass, position encoders, and bumpers. The visual system is composed by a fixed color camera (Hitachi, KP-D50) and a PCI frame grabber (Matrox Meteor). The ultrasound sensors (Polaroid 6500) can measure distances from 6 inches to 35 feet, with a typical absolute accuracy of 1 percent over the entire range. The robot has a Pentium-Pro 200 MHz processor, with 64 Mb of RAM running Linux, and all the image processing is performed in it.

Figure 2. Nomad 200 mobile robot.

The landmark used was a black square with two fixed sizes: 5x5 and 10x10 inches. The setup of the experiment, with the robot moving in the direction of the 5 inches landmark is shown in figure 3. Figure 4 shows two images

(320x240 pixels) of the landmark after the application of the threshold, captured from 85 and 15 inches (widths 17 and 90 pixels respectively). The experiments using each size were repeated several times and the measurements from the visual method was compared with readings from a Polaroid ultrasound sensor.

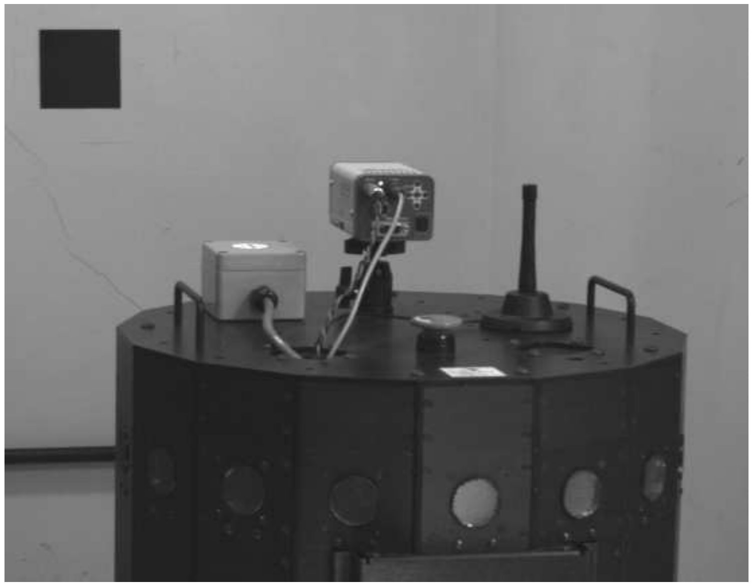
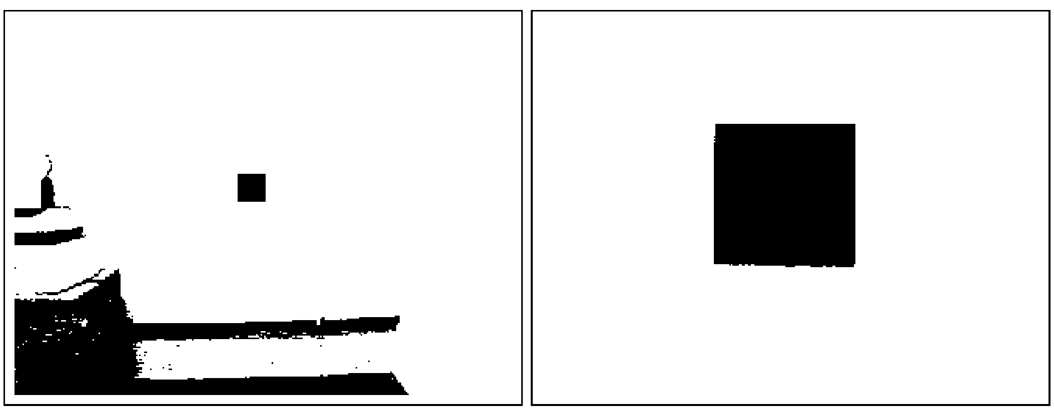


Figure 3. Setup of the experiment: the robot is moving in the landmark direction.



(a) (b)

Figure 4. Images of the landmark after the threshold: (a) distance: 85 inches, width: 17 pixels e (b) distance: 15 inches, width: 90 pixels.

The first experiment was performed using the 5 inches landmark. The graphic of Figure 5 shows the results: the x-axis is the frame number (the first is frame 0) and the y-axis is the distance computed in inches by the sonar and the camera for each frame. Table 3 shows the results in a numerical form: the second row shows the width (in pixels) computed for each frame and rows 3 and 4 shows the distances measured by the camera and sonar respectively. An important detail is that the focus of the camera is positioned approximately 2 inches behind the ultrasound sensor. So, in the measurements shown on table 3 a little adjust has to be done in order to compare the sonar readings with the camera measurement. The distance moved by the robot between each frame is about five inches and the frame 0, where the movement begins, is captured from approximately 80 inches of the landmark. The position where the

Frame	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Width	17	18	20	21	24	26	29	31	35	40	46	54	66	86	123	199
Camera	-	85	45	100	35	60	43	73	39	35	33	29	23	17	12	8
Sonar	82	77	71	66	61	56	51	46	41	36	31	26	21	16	10	5

Table 1. Data of the execution of the algorithm with 5 inches landmark.

robot starts its movement do not influence the measurements. This experiment was also performed with the robot beginning at 120 inches from the landmark and the results are almost the same.

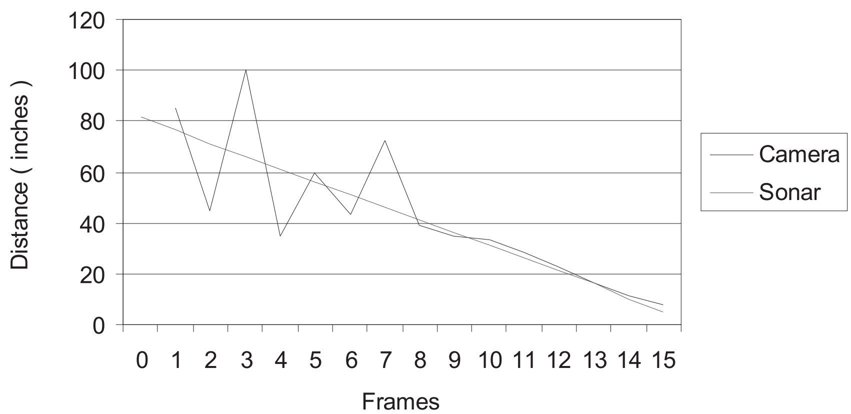


Figure 5. Vision based distance estimation and sonar readings when the robot is approaching a 5 inches landmark.

In this experiment, it can be observed that the method does not work fine when the distance is greater than 40 inches (before frame 8). This happens because, when the camera is far from the landmark, the widths computed in two consecutive frames are almost the same. A difference of 5 inches in the distance causes only differences of 1 or 2 pixels between the frames and this leads to a wrong distance estimation. When the camera gets closer to the landmark (less than 40 inches), the visual estimation works fine, and the measurements from the sonar and from the camera coincide. Figure 6 exposes this fact more clearly: it shows the difference of the distances estimated by the camera in relation to the sonar measurements (x-axis). It can be seen that when the landmark is close (less than forty inches), the difference in the measurements are small.

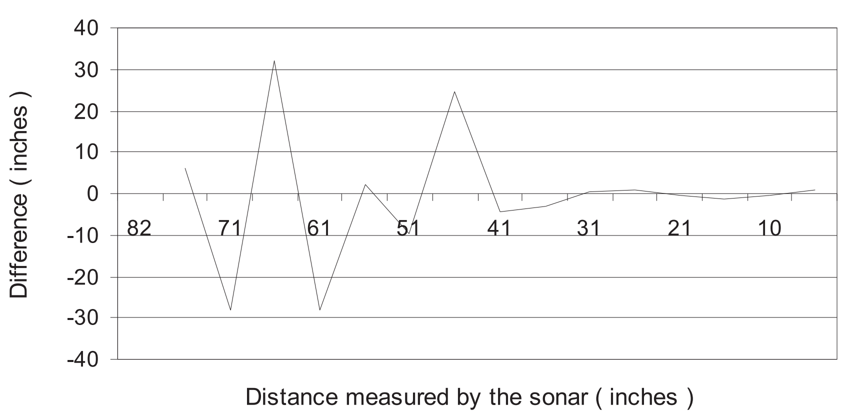


Figure 6. Differences of the distances measured by the cmera in relation to the sonar readings

The incorrect results obtained when the robot is far from the landmark can be better understood looking at Figure 7. The graph shows the landmark size (in pixels) computed in each frame as the robot is approaches. The width is inversely proportional to the distance and it can be observed in the first frames that the curve slope is close to zero, causing the mistakes in distance computation. In some experiments, when the distance is greater than 100 inches, the extreme case happened: the difference between the widths in two consecutive frames reaches zero causing a zero division in Equation 3.

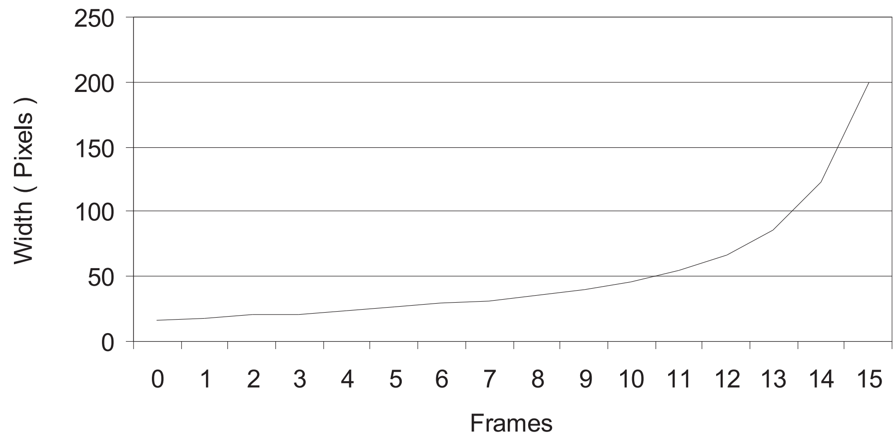


Figure 7. Landmark dimensions computed by the visual method.

This kind of mistake can be avoided in part using a larger landmark. When the size is increased, the width differences in two consecutive frames also increase, and the distance computation becomes more precise. Figure 8 shows this fact: using the 10 inches landmark, the method begins to work correctly earlier, when the distance is about 60 inches. As mentioned before, the initial position of the robot (120 inches in this experiment) does not influence the measurements. The problem using larger landmarks is that it occupies all the image area (or hit one of the image borders) when the robot is very close to it. When this happens the method could not compute the width correctly, and the robot stops its movement. In the graphic of figure 8 the robot stops its motion before passing the 10 inches mark.

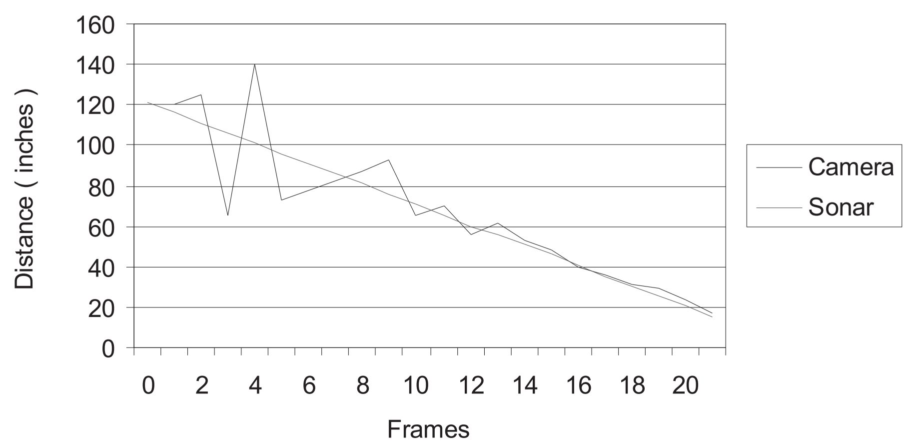


Figure 8. Vision based distance estimation and sonar readings when the robot is approaching a 10 inches landmark.

As mentioned before, one of the key advantages of this method is its simplicity. The segmentation of the landmark, the width computation and the distance estimation are very simple tasks, and can be executed very rapidly. To confirm this fact, the processing times was measured: the average time of processing one image and computing the distance varies from 11 to 17 miliseconds, proportionally to the size of the landmark. Most of this time is spent in the segmentation algorithm. The image capture spends a little bit more time: 50 to 60 miliseconds. The total time spent with each frame (capture + processing) is about 70 miliseconds, which give a rate of approximately 15 frames per second. This rate can be improved even more: in this implementation, the image is captured by the camera in colors, and only one of the three bands RGB is passed to the threshold. Hence, the use of grayscale images at the capture, and some modifications in the segmentation method can improve the performance. But even without these improvements, the processing times measured indicate that this method can be used to control a moving vehicle with good precision.

4. CONCLUSIONS

This work described a simple method to estimate the distance of a mobile vehicle to an artificial landmark using an uncalibrated camera. It uses the landmark width in consecutive frames and the distance traveled by the robot between them to compute the distance. This method was used in a algorithm to control the approach of a mobile robot to the landmark.

Several tests were performed where the distance estimated by the visual method was compared to readings of ultrasound sensors. The results showed that the method works well when the distance from the robot to the landmark is relatively small. This distance depends on the landmark size, and varied in the experiments from 40 inches, using the 5 inches landmark, to 60 inches using the 10 inches landmark. One of the main advantage of this method is that it is very fast, spending approximately 70 miliseconds on average to capture and process an image. This gives a rate of 15 frames per second, which is sufficient to control a moving vehicle with good precision.

Future works will include improvements in the method and studies to apply it in a more complex navigational task, such as obstacle avoidance or tracking for example. Besides that, it is possible to use this method with other kinds of vehicles on other activities. For example, a docking system for a piler or an landing system for a blimp.

ACKNOWLEDGMENTS

This work is partially funded by FAPEMIG grant TEC-609/96, and CNPq grants 522618/96-ONV, 170645/97-5NV-BSP and 141558/98-9.

REFERENCES

1. R. Chatila, "Mobile robot navigation algorithms," in *Algorithmic Foundations of Robotics*, A. K. Peters, 1995.
2. T. L. Dean and M. P. Wellman, *Planning and Control*, Morgan Kaufmann Publishers, San Mateo, California, 1991.
3. V. Bruce, P. R. Green, and M. A. Geogeson, *Visual Perception - Physiology, Psychology and Ecology*, Psychology Press, East Sussex, UK, 1996.
4. Z. Zhang and O. D. Faugeras, "Building a 3d world model with a mobile robot," in *Proc. of the 10th International Conference on Pattern Recognition*, 1990.
5. S. Hutchinson, G. D. Hager, and P. I. Cork, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation* **12**, pp. 651–670, October 1996.
6. B. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1987.
7. C. Fremüller and Y. Aloimonos, "Tracking facilitates 3-d motion estimation," *Biological Cybernetics* **67**, pp. 259–268, 1992.
8. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision* **12**(1), pp. 43–77, 1994.
9. C. Becker, J. Salas, K. Tokusei, and J.-C. Latombe, "Reliable navigation using landmarks," in *Proc. of teh IEEE International Conference on Robotics and Automation*, pp. 401–406, 1995.
10. C. J. Taylor and D. J. Kriegman, "Vision-based motion planning and exploration algorithms for mobile robots," *IEEE Transactions on Robotics and Automation* **14**, pp. 417–426, June 1998.
11. D. P. Huttenlocher, M. E. Leventon, and W. J. Rucklidge, "Visually-guided navigation by comparing edge images," in *Algorithmic Foundations of Robotics*, A. K. Peters, 1995.