Development of a Visual Object Localization Module for Mobile Robots

Erol Şahin and Paolo Gaudiano*
Boston University Neurobotics Lab
Dept. of Cognitive and Neural Systems
677 Beacon St., Boston, MA, 02215, USA
{erol, gaudiano}@cns.bu.edu

Abstract

This article reports preliminary results from the design and implementation of a visual object localization module for mobile robots. The module takes an object-based approach to visual processing and relies on a preprocessing step that segments objects from the image. By tracking the size and the eccentricity of the objects in the image while the robot is moving, the visual object localization module can determine the position of objects relative to the robot using the displacement obtained from its odometry. In localizing the objects, the module is designed to combine the results of two different techniques. The visual looming *technique measures the distance to an object using the change in the size of the object on the image plane. This technique is to be complemented by a variant of the* triangulation *technique that can locate an object using the eccentricity of the object when viewed from two different points. The module – with the preprocessing algorithm – is being implemented to run in real-time on a mobile robot.*

Evaluation of the visual localization module is being done in an integrated system introduced in this article. The integrated system creates an environment for real-time evaluation of the module as well as other mapping and navigation algorithms for mobile robots.

1. Introduction

A robot's ability to navigate successfully in unstructured environments is often determined by the quality of its perception, which in turn depends on the richness and reliability of its sensors. Visual sensors can provide rich and high-dimensional information about the environment. However, high computational requirements and a lack of fast and robust visual algorithms have prevented extensive use of visual sensors in mobile robotics. Other sensors, such as sonar, have often been preferred to vision due to their simplicity. Recently, new visual algorithms that can run in real-time on mobile robot platforms have begun to emerge, making visual processing useful as a tool for mobile robot navigation [1, 7, 10, 13, 15].

Some visual navigation algorithms process each image separately to extract navigational information. One example is Horswill's *"visual sonar"*, [10], which extracts from a camera image the relative distance of obstacles on the floor. Other algorithms [7, 15], use the principle of *motion parallax* [9] which states that a field of velocity vectors, also known as *optical flow*, appears on the retina of a moving agent. These algorithms work in the spatiotemporal domain, processing a sequence of images to extract and use optical flow for navigation.

The major challenge in working with a sequence of images is that visual entities have to be tracked through the sequence. This is done by solving the *temporal correspondence problem*, i.e., matching corresponding entities between images. The entities can range from pixels to lines and regions. The choice of the type of visual entity to be tracked has important implications for the complexity of the processing and the nature of the visual information to be extracted. For example, although pixels provide a dense representation of the scene, solving the temporal correspondence problem becomes costly due to the number of pixels, and tracking pixels over a long sequence of images is almost impossible.

At the other extreme, some algorithms track larger visual entities, such as lines and regions across multiple frames [5]. Here the tracking becomes easier, but the initial segmentation in each frame can be problematic. For instance, the appearance of an object being tracked might change dramatically as a result of cast shadows or changes in viewing perspective.

This project takes an *object-based* approach to visual processing. Object-based processing works in a low-dimensional *object space* created by an algorithm that segments "objects" from the image. Objects are defined as regions that can be segmented easily from the background.

*This work is supported by DARPA, the Office of Naval Research and the Naval Research Laboratory through grants ONR-00014-96-1-0772 and ONR-00014-95-1-0409.

0-7803-5672-1/99/$10 © 1999 IEEE

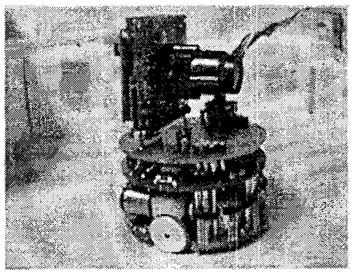
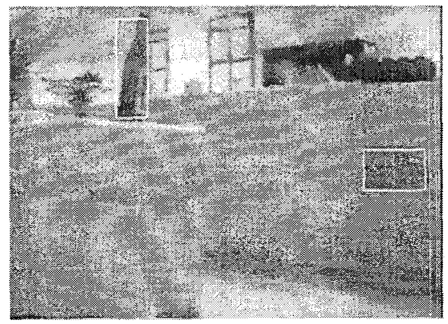
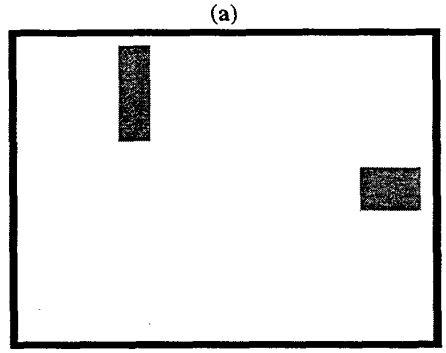


Figure 1. The miniature robot Khepera equipped with a color camera.



(a)



(b)

Figure 2. Preprocessing: (a) Two compact regions (one is a red bottle on the table, the other is a yellow block in the wall) are chosen as objects. Bounding boxes of these objects are drawn as white rectangles. (b) Output of the preprocessing algorithm which can be referred to as an object-space representation of the scene.

They may or may not correspond to actual objects in the environment, and they usually depend on the segmentation algorithm being employed. In the object space, objects are represented by a set of features such as color, size and position in the image.

The temporal correspondence problem, which is a difficult and computationally expensive problem to solve in the pixel space, becomes simpler in the object space. Cheap and robust tracking of objects leads to a longer tracking time and opens up new possibilities for visual navigation of mobile robots.

The visual object localization module and the techniques being implemented in the module are discussed in the following section. Section 3 describes an integrated system being developed for the real-time quantitative evaluation of the module. A miniature Khepera robot, as shown in Fig. 1, will be used as the robot platform. Then, preliminary results of the module in the integrated system are presented. The article concludes with a discussion of the module and the integrated system.

2. Visual Object Localization Module

The main goal of the project is to build a complete visual module to localize objects using the on-board camera as the robot moves in its environment. The change in the size and the eccentricity [1] of objects on the image due to the displacement of the robot are combined with the amount of displacement obtained through robot's odometry for the localization of objects. Localization of objects is important not only for navigation, but also for the robot's self-localization, mapping, and other functions requiring knowledge of the robot's position in its environment.

The visual object localization module works in the object space created by a preprocessing algorithm, as shown in Fig. 2. It uses the position and the size of the bounding boxes of the objects to localize them. Ideally, the preprocessing algorithm should be able to segment an unstructured scene into objects using visual cues such as shape, texture, edges and color in real-time. In the lack of a preprocessing algorithm which can satisfy all these constraints for a completely unstructured environment, one is forced to put some structure into the environment to make the detection and segmentation of objects easier.

In this project, objects are defined by patches of compact colored regions in the image with no assumptions about their color, size or shape. In particular, blocks of different sizes and colors are inserted into the walls. The preprocessing algorithm analyzes the scene for patches of compact colored regions that can be segmented from the background easily. The image is segmented into colored regions to create a low-dimensional object representation of the scene.

The preprocessing algorithm runs in two phases. The

[1] Eccentricity of an object is defined as the angle between the object and the optical axis of the camera

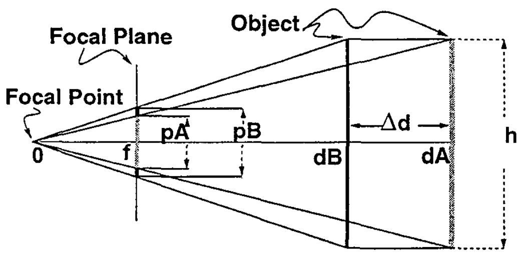


Figure 3. Diagram of the visual looming relationships in a camera-centered coordinate frame.

image is subsampled and normalized in the first phase. In the second phase, the pixels of the subsampled image are clustered in the color (RGB) domain with a clustering algorithm. Each cluster is also tagged with its bounding box on the image and the number of pixels included in it. Clusters that are neither too large nor too small and relatively compact on the image are selected as objects worth tracking. The bounding boxes of these objects are then fine-tuned in the full-resolution image.

The module will combine and extend two different techniques. The *visual looming* technique estimates the distance to an object using the change in the projection size of the object that results from known robot displacements. This technique will be complemented by a variant of the *triangulation* technique which can locate an object using the eccentricity of an object from two different positions. Currently, only the visual looming technique is implemented in the module. Both techniques are described in the next two subsections.

2.1. Visual Looming

Visual looming, the expansion of the projection size of an object on the retina, is usually the indication of an approaching object. It is normally perceived as a threat for a possible collision and is sufficient to elicit avoidance and escape behaviors in animals [4].

Although its behavioral effects have been studied mainly in psychology, looming also has interesting implications for mobile robotics. Several independent studies have reported the use of looming for obstacle avoidance [12] or for extracting the depth of an object [11, 16, 17]. In particular, Raviv [14] has done an excellent quantitative analysis of visual looming. He defined the looming of a point mathematically and showed how this information can be used for effective obstacle avoidance behavior.

The geometric relations shown in Fig. 3 assume that camera is viewing an object of size h from two different positions, at distances d_A and d_B from the object. Note that it is irrelevant whether the displacement between the two positions is the result of camera movement or object movement. In the rest of the discussion, it is assumed that objects are stationary while the camera is moving. When the camera is mounted on a mobile robot, the displacement Δd can be obtained directly through the robot's odometry.

Given a camera focal length f, the size of the projection of the object onto the focal plane depends on the distance between the object and the camera. In the case shown in Fig. 3, p_A and p_B, respectively, represent the size of the projection of the same object at distances d_A and d_B. Using similar triangles, it can be easily shown that

$$d_B = -p_A \frac{\Delta d}{p_B - p_A}, \quad d_A = -p_B \frac{\Delta d}{p_B - p_A} \qquad (1)$$

where $\Delta d = d_B - d_A$ denotes the net displacement. Since p_A, p_B and Δd are all known, the initial and final distance to the object can be computed using Eqn. 1 which is referred to as the *looming equation.*

Some important observations can be made from the looming equation.

1. The looming equation can be applied to vertical or horizontal dimension of the projection independently.
2. The looming equation is independent of the implicit parameters of the optical system, such as the focal length, and the pixel size. However, calibration of the camera is necessary to measure the eccentricity of the object.
3. The looming equation is independent of the actual object size.
4. The range estimates have the units of displacement, i.e., robot's odometry. Therefore range estimates can be considered as independent of the robot platform.

The looming geometry suggests that the displacement of the robot be perpendicular to the object plane. This is required to prevent any foreshortening effects of the projection due to the displacement. Usually the object plane is unknown, and the only acceptable type of displacement to minimize foreshortening is towards or away from the object. In most prior analyses and implementations, the looming technique is used when the robot is moving towards the object. The only exception is Joarder and Raviv's use of looming for obstacle avoidance [12]. They used spherical light bulbs as objects to eliminate the foreshortening effect. The restriction on the type of displacement, or on the type of the objects, is a major bottleneck for the looming as an object localization technique.

A more careful look at the looming geometry, however, reveals that the displacement has to be perpendicular only to the dimension (horizontal or vertical) of the object being

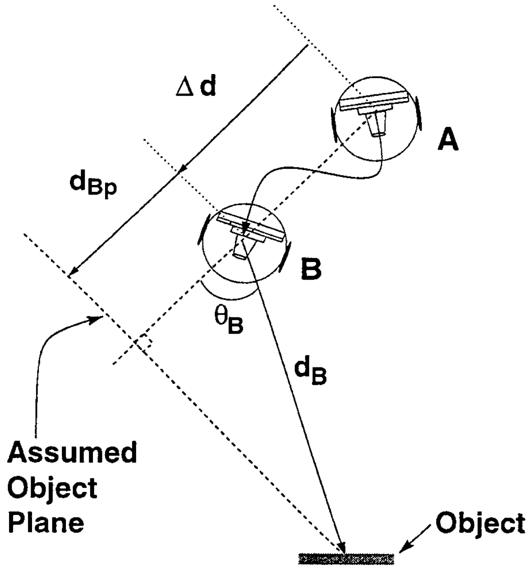


Figure 4. On a flat surface, vertical looming can be used to localize an object under arbitrary displacements.

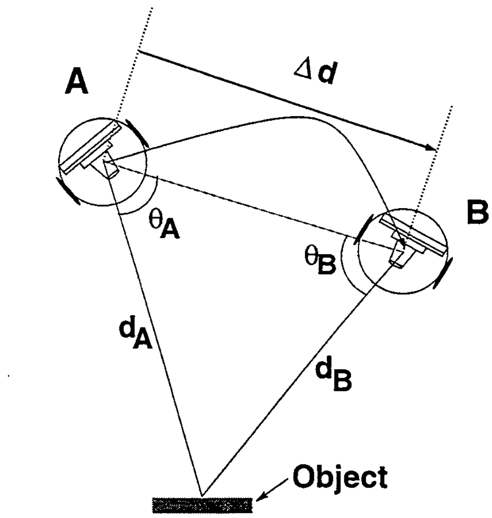


Figure 5. An object can be localized on the basis of its eccentricity at two different points separated by a known displacement vector.

used by the looming technique. For instance, if the vertical looming of the object is being used, then the displacement must be perpendicular to the vertical dimension of the object only. In an indoor environment, usually the robot moves on a flat ground and the objects stand vertical. In such an environment, though horizontal looming of objects is subject to foreshortening, usually vertical looming of objects is not. Vertical size (height) of the projection of an object depends only on the distance between the observer and the object. It is independent of the slant angle of the object with respect to the robot [2]. Therefore, use of vertical looming can relieve the restrictions on the type of robot's displacement.

Figure 4 sketches how vertical looming can be used to localize an object when the robot makes an arbitrary displacement. In the figure, the robot moves from point A to point B making a net displacement of Δd. The object is assumed to lie on a "assumed object plane" that is perpendicular to the displacement. d_{Bp}, the distance to the assumed object plane can then be computed using the looming equation. Finally the range of the object can be derived from d_{Bp} using

$$d_B = d_{Bp} \cos \theta_B$$

where θ_B is the angle between the object and the displacement vector from robot's position.

This extension makes the looming a useful object localization technique that can be employed on a freely moving robot.

2.2. Running Fix Triangulation

Triangulation is a localization technique for determining the location and orientation of a robot using three or more beacons. It allows a robot to determine its location and orientation by measuring the eccentricity of objects whose positions are known, [2, 6].

An interesting variation of triangulation is the *running fix* [3] triangulation technique proposed by Case [3]. The geometry of the running fix triangulation technique is sketched in Fig. 5. The basic idea of the running fix is the utilization of the angle or range obtained from a beacon at a previous time step, by adding the displacement vector to the position of the beacon. Case used the running fix triangulation technique to determine the position and orientation of a mobile robot using a single landmark.

[2]Here, 'projection size of an object' is used in an ideal sense. If the size of the bounding box is used as the projection size of the object, then the vertical size of the projection of an object may not be independent of the slant angle. However, the change in the vertical size due to the slant angle depends on the shape of the objects, and is expected to be small for most of the objects .

[3]This is a known technique in celestial navigation and has been used by sailors. By making successive measurements of a celestial body over a period of few hours, one can estimate his latitude and longitude.

Two views of an object separated by a known displacement vector can be used to localize the object. The eccentricity of the object in both views and the change in the robot's orientation are used to calculate the angles θ_A and θ_B. The displacement vector derived from odometry is combined with these angles to compute the object range as,

$$d_B = \Delta d \frac{\tan\theta_B}{\cos\theta_A(\tan\theta_A + \tan\theta_B)},$$
$$d_A = \Delta d \frac{\tan\theta_A}{\cos\theta_B(\tan\theta_A + \tan\theta_B)}.$$

Initial analysis suggests that visual looming and triangulation can be complementary techniques for object localization. When the robot is moving directly toward an object, triangulation fails since the eccentricity remains unchanged, whereas visual looming can localize the object by using the change in the retinal size of the object. On the other hand when the robot is moving on a curved path, triangulation is expected to produce better localization than looming.

An important part of the project will be the evaluation of the visual object localization module. The techniques to be implemented in the visual localization module require some assumptions and simplifications. For instance, the visual looming technique implicitly assumes navigation on a flat surface and planar objects perpendicular to the ground. The bounding box segmentation of objects neglects the effect of perspective when objects are slanted relative to the robot's line of sight. The triangulation technique implicitly assumes that the centroid of the object images corresponds to the same point on the object at both points. A complete theoretical analysis of these problems and others is very difficult. Moreover there may and will always be unforeseen problems with these techniques. To obtain quantitative performance data under a variety of constrained or unconstrained movements, the object localization module, which combines looming and triangulation, will be evaluated in the integrated system that will be discussed in the next section.

3. Integrated system for the real-time evaluation of the object localization module

Quantative evaluation of *object localization* methods has been a difficult task in mobile robotics research. To evaluate the performance of an object localization method, one has to know not only the position of the robot, but also the position of the object that is being localized. In typical localization systems [2], the position of the robot is usually tracked using a number of active or passive beacons mounted in the environment. The position of objects needs to be derived from a model of the environment created through manual measurements. However, the creation of an environment model through manual measurements is time consuming. It introduces an extra source of error into the system, and it makes changes in the environment cumbersome to encode.

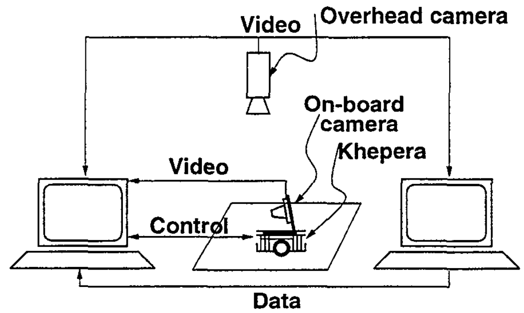


Figure 6. Sketch of the Integrated system.

We have developed an integrated system for real-time evaluation of object localization methods that does not need an environment model. The integrated system is built using the setup sketched in Fig. 6. It consists of a modifiable, indoor-like environment for the miniature robot Khepera and an overhead camera mounted over the environment. The integrated system is implemented on two desktop computers running the Linux operating system. The computers are connected via a local area network. Each computer is equipped with a frame grabber. The overhead camera is connected to both of the computers. One of the computers uses the overhead camera to track the Khepera using a small color marker on the robot. The position and the orientation of the color marker, and hence of the robot, are computed and then passed to the second computer. The second computer is connected to the Khepera and its on-board camera. The visual object localization module runs on this computer to process the images coming from the Khepera's on-board camera.

The system creates an integrated view as shown in Fig. 7. First the overhead view of the environment is grabbed as the background. The position of the robot and the estimated positions of the objects are then superimposed on this background. The position of the robot is tracked by the overhead camera and drawn on the integrated view as a circle. The estimations obtained from the object localization module through the on-board camera are drawn in the integrated view as rays emanating from the robot. Since the position of each object is also visible in the integrated view, the performance of the object localization method can be visually and quantitatively evaluated.

In the next section, preliminary results from the evaluation of the partially implemented visual object localization module are presented and discussed.

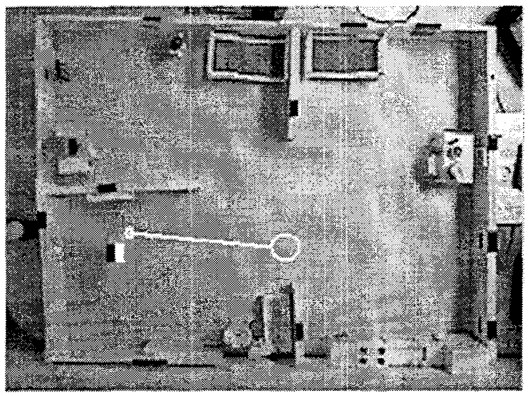


Figure 7. Integrated view for the real-time visual and quantitative evaluation of the object localization module. Here the object localization module estimating the position of an object, marked with a white rectangle, while the robot is approaching.

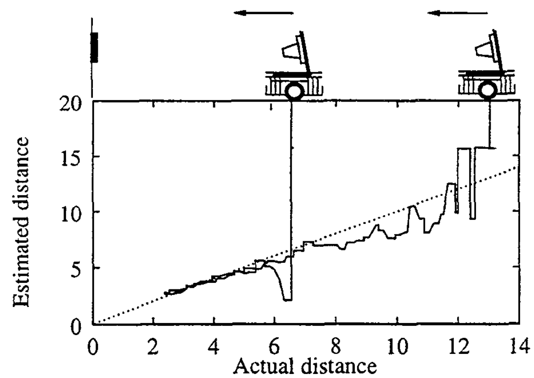


Figure 8. Range estimations of the looming algorithm when the robot starts approaching an object from two different distances.

4. Preliminary Evaluation of Object Localization Module in the Integrated System

The integrated system provides a nice testbed for the evaluation of the visual object localization module being developed. The module is not fully implemented yet. Only the basic looming algorithm is implemented to localize objects while the robot is moving towards an object, as shown in Fig. 7.

An important concern in the practical applicability of the looming algorithm is its noise sensitivity. It can be seen from Eqn. 1 that the looming equation is likely to amplify any noise when the change in the projection size is small. Therefore, it is important to know how different sources of noise contribute to the range error.

The looming equation has two sources of noise: odometric and visual. Odometric noise is the error, due to slippage and other factors, in measuring the displacement of the robot. It is proportional to the distance traveled and usually in the order of a few percent of the distance traveled on most surfaces. Visual noise arises in measuring the projection of the object on the focal plane, and is due mainly to pixel quantization, object segmentation and optical distortion.

Şahin and Gaudiano [8], analyzed the noise sensitivity of the looming equation. The analysis led to several important observations about looming. First, error in the estimated range is proportional to the distance, which means that – other parameters being constant – measurements will get better as the robot gets closer. Second, the range error is initially dominated by the noise in the projection size change, whose effect decreases as the projection size change increases.

The amount by which the projection size changes depends jointly on the distance and on the size of the object: a short movement toward a nearby small object may yield the same projection size change as a longer movement toward a distant large object. Therefore, rather than considering absolute size and distance, it is most sensible to use the ratio of the distance to the size of the object as a dimensionless parameter to measure the error.

Figure 8 shows the results of an experiment demonstrating the performance of the looming algorithm when the robot is approaching an object. The figure consists of two parts. The upper part is a sketch of the experiment in which the robot approaches an object from two different distances. The object is shown as a thick vertical strip on the left of the figure and the starting positions of the robot are drawn to the right of the object.

The lower part of the figure plots the results of the experiment: Estimated ranges to the object are plotted against the actual distances. The axes are marked in units of object size (height for this case) to make the results independent of the size of the object or the actual distance to the object. The left portion of the plot shows the results when the robot starts approaching the object from a distance of 6.5 units. It can be seen that range estimates gets into an acceptable precision after moving one unit. The right portion of the figure plots the range estimates when the initial distance is twice as far (13 units). It takes almost 5 units of displacement for the range estimates to get into the same precision These results agree with the results of the theoretical noise sensitivity analysis which is briefly summarized above.

The sudden jumps intermitted by slow variations are due to the difference between the rate of processing in the module, and the rate of communication with the robot.

5. Discussion

Within the frame of this work, the following claims can be made:

- A preprocessing step can create a low-dimensional object-space representation of the scene from the high-dimensional visual data. The 'complex' visual sensor can be transformed into a simpler sensor –output of which is easier to handle– with such preprocessing.

- The preprocessing can be done in real-time (at rates close to frame rate) using standard hardware and current color tracking algorithms. Also off-the-shelf hardware designed to track arbitrary colors at frame rate are available.

- The idea of automatic selection of colors that 'worth' to be tracked by the color tracking algorithm seems promising. The current implementation of this idea for the indoor-like environment seems to work in unstructured environments as well.

- Storing and matching of scenes becomes easier with the low-dimensional object space representation. This representation made it possible to track objects over a long sequence of frames. It can also be promising for the place recognition task with the on-board camera of a mobile robot.

- Longer tracking of objects allows the robot to make bigger displacements, resulting in larger changes in the size and the eccentricity of objects. This reduces the effect of noise on the visual object localization module and allow it to make accurate localizations.

- The visual object localization module combines the odometry information with the visual information to localize objects.

- The localization information obtained from the visual object localization module is labeled, i.e. it tells not only that there is an object at a certain location, but it also tells the color, etc., of the object. This can be very helpful for navigation algorithms. Other range sensors, like sonar, infrared or laser, lack this feature.

- The integrated environment is a useful tool for quantitative evaluation of visual algorithms on the miniature Khepera robot.

6. Conclusion

This article introduced a visual object localization module which takes an object-based approach to visual processing. The module is designed to localize objects relative to the robot, by tracking the change in the size and eccentricity of their images. The techniques to be implemented in the module were described. An integrated system for the evaluation of the module was also introduced. Finally some preliminary results from the evaluation of the partially implemented module in the integrated system were presented.

The visual object localization module is close to completion. The integrated system will be used for quantitative evaluation of the module for different cases. Though the integrated system is created for the evaluation of the module, it can be used to evaluate other visual or nonvisual localization and map creation methods as well.

References

[1] N. Ancona and T. Poggio. Optical flow from 1D correlation: Application to a simple time-to-crash detector. In *Proc. Image Understanding Workshop, IUW-93*, pages 673–682, 1993.

[2] J. Borenstein, H. Everett, and L. Feng. *Navigating mobile robots.* A.K. Peters, Wellesley, MA, 1996.

[3] M. Case. Single landmark navigation by mobile robot. In *Proc. of the 1986 SPIE Conf. on Mobile Robots*, pages 231–237, Cambridge, MA, Oct. 1986.

[4] J. A. Caviness, W. Schiff, and J. J. Gibson. Persistent fear responses in rhesus monkeys to the optical stimulus of 'looming'. *Science*, 136:982–983, 1962.

[5] R. D. Chaney. Multiple-frame motion estimation using simple region features. In *Proc. of SPIE, Intelligent Robots and Computer Vision XIII: 3D Vision, Product Inspection, and Active Vision*, volume SPIE-2354, pages 10–25, 1994.

[6] C. Cohen and F. Koss. A comprehensive study of three object triangulation. In *Proc. of the 1993 SPIE Conf. on Mobile Robots*, pages 95–106, Boston, MA, Nov. 1993.

[7] D. Coombs and K. Robert. "Bee-bot": using peripheral optical flow to avoid obstacles. In *Proc. of the SPIE Conf. on Intelligent Robots and Computer Vision XI: Algorithms, Techniques, and Active Vision*, Boston, MA, Nov. 1992.

[8] E. Şahin and P. Gaudiano. Mobile robot range sensing through visual looming. In *Proc. of the ISIC/CIRA/ISAS*, pages 370–375, Gaithersburg, MD, USA, 1998. NIST.

[9] H. v. Helmholtz. *Handbuch der Physiologischen Optik.* Voss, Leipzig, 1867.

[10] I. Horswill. Collision avoidance by segmentation. In *Proc. of the 1994 Int. Conf. on Intelligent Robots and Systems (IROS-94).* IEEE Press, 1994.

[11] D. P. Huttenlocher, M. E. Leventon, and W. J. Rucklidge. Visually-guided navigation by comparing edge images. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 85–96. A K Peters, 1995.

[12] K. Joarder and D. Raviv. Autonomous obstacle avoidance using visual fixation and looming. In *Proc. of SPIE*, volume 1825, pages 733–744, 1992.

[13] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson. Visually-guided obstacle avoidance in unstructured environments. In *Proc. of the IEEE Conf. on Intelligent Robots and Systems*, September 1997.

[14] D. Raviv. Visual looming. In *Proc. of SPIE*, volume 1825, pages 703–713, 1992.

[15] M. V. Srinivasan, J. S. Chahl, and S. W. Zhang. Robot navigation by visual dead-reckoning: inspiration from insects. *Journal of Artifical Intelligence and Pattern Recognition*, 11:35–47, 1997.

[16] L. R. Williams and A. R. Hanson. Translating optical flow into token matches and depth from looming. In *Proc. of the 2nd Int. Conf. on Computer Vision*, 1988.

[17] S. B. Xu. Qualitative depth from monoscopic cues. In *Proc. of the Int. Conf. on Image Processing and its Applications*, pages 437–440, 1992.