ACTIVE DISTANCE MEASUREMENT BASED ON ROBUST ARTIFICIAL MARKERS AS A BUILDING BLOCK FOR A SERVICE ROBOT ARCHITECTURE

Axel Walthelm, Ralf Kluthe

Institut für Technische Informatik, Medizinische Universität zu Lübeck
Ratzeburger Allee 160, D23538 Lübeck, Germany
{walthelm,kluthe}@iti.mu-luebeck.de

Abstract: Software architectures for service robots have to support flexible combinations of basic algorithms to form complex goal directed behaviors. This paper presents a building block for an active distance measurement: the **G**lare **La**ser Range Scanner **R**obotics **E**xperiment.The setup and the intended functionality of GLaRE are described and the more complex algorithms that form modules within GLaRE will be presented in detail. A artificial visual marker is chosen as a point of attention, for which the distances has to be determined. The detection of the marker is distance and orientation invariant and thus suitable for both experimental and practical applications. The overall architectural design of GLaRE will be described.

Keywords: Robot vision, Sensor fusion, Robot control, Architectures.

1. INTRODUCTION

Software architectures for mobile service robots have to support different kinds of algorithms like control, perception, and reasoning. Normally, the algorithms are combined to form more complex goal directed behaviors. These behaviors in turn can be used in even more complex behaviors. From the software designer's view, behaviors should be considered as building blocks. A typical example for such a building block is a behavior which extracts areas or points of interest out of the environment and out of the given sensor readings respectively, tracks them in the case of motion, and determines their distance profile. This kind of behavior may be used in more complex behaviors like grasping, navigation, and obstacle avoidance. Therefore this paper focuses on the exemplary development of an active distance measurement system, called GLaRE, as a building block for service robot architectures.

A passive artificial marker was chosen for this to supply a point of attention. The detection of artificial markers is easier and more robust than the detection of particular features in natural scenes. Therefore, artificial markers are often used for practical and experimental realizations in robot navigation [2], tracking and following [1], visual servoing [5], grasping [9], etc. For many applications the distance of the detected markers relatively to the robot has to be determined.

Artificial markers can be divided into active and passive markers. Passive markers are easy to mount, but normally they are very simple and lack for robustness in natural scenes due to their non-complex structures, e.g. blobs of black or luminous color [5][2]. Although a simplified barcode system is used by [9], barcodes [6] are specialized to carry much information and therefore, they are less robust with respect to distance changes. Active markers like light-bulbs, frequency modulated LEDs, or satellites [10] are complex in that they need power supply or even radio communication. Additionally their installation is expensive.

In order to use GLaRE as a building block, its architecture is based on state representations. State representations are local data memories, which allow loose coupling of modules. Modules are interfacing only to

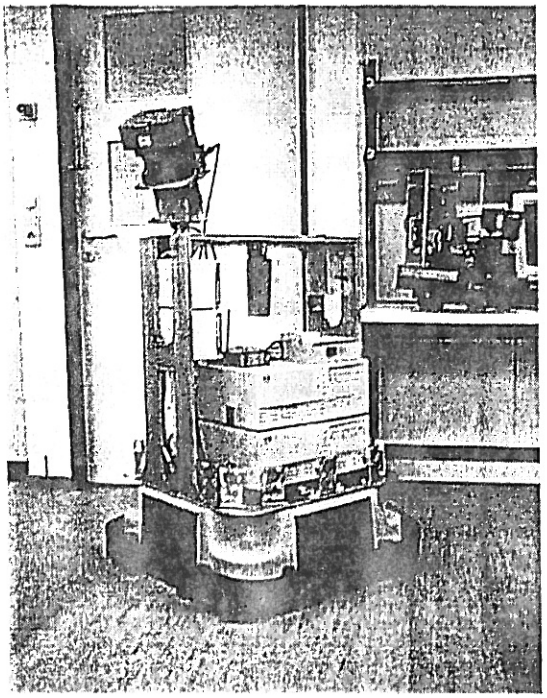

Figure 1. Setup of GLaRE mounted on top of the experimental mobile service robot MAVERIC.

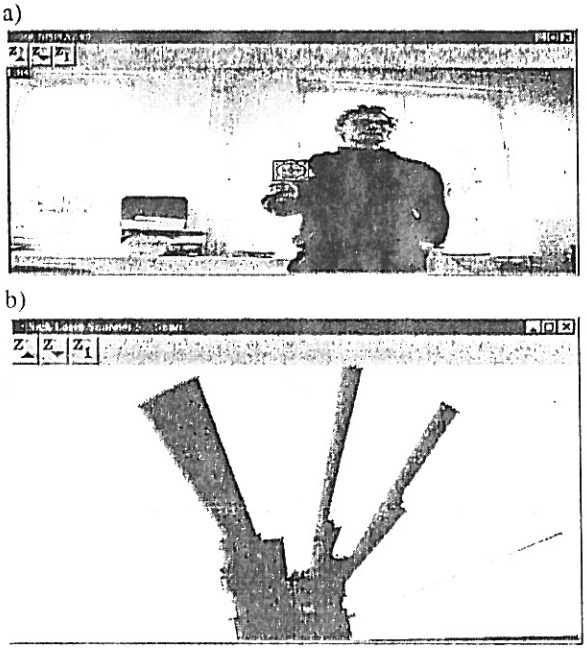

Figure 2. Screenshots of the most important GLaRE windows: a) current camera half-image with results from marker detection and distance computation, b) visualization of the laser scan and the point chosen by distance computation.

data and not to specific modules, which have to be determined at design time. Thus, GLaRE can be adapted easily to varying demands without affecting the overall functionality. For example, the currently used algorithm for detecting artificial markers could be replaced by one for natural markers.

In the following sections the algorithms and architectural concept of GLaRE are described. Section 2 describes the intended behavior of GLaRE and its setup. Section 3 explains the two most complex modules in detail: marker detection and distance computation. Section 4 describes the overall architecture as a continuous data flow based on state representations.

2. EXPERIMENTAL SETUP AND INTENDED BEHAVIOR OF GLARE

In the current implementation of GLaRE the environment is visually observed for the occurrence of interesting objects. The distance of such an object has to be explored actively by focusing on the object and "glaring" at it with both a camera and a laser range scanner. An alert will be triggered if the distance is too short.

Scene interpretation and general object identification is still a complex problem and an active research topic. Focusing on architectural concepts, we developed an artificial marker to represent these objects. This marker will be described below in section 3.1.

The basic strategy to do this is to

1. localize the focus of attention within the camera image,
2. move the camera in a way that the marker moves to the center of the image,
3. use the laser range scanner – which is fixed mounted with the camera – to compute the distance corresponding to the central point of the image, and
4. trigger the alarm if a marker is focused and its distance is critical.

Figure 1 shows the hardware setup of GLaRE mounted on top of the experimental mobile service robot MAVERIC (Mobile Autonomous Vehicle to Experiment upon Robotic Indoor Chores). The hardware used in the current GLaRE implementation consists mainly of an Amtec pan–tilt unit, a JAI CV-M1250 color camera and a SICK laser range scanner connected to a standard PC running Windows NT.

Figure 2 shows a screenshot produced by GLaRE. The upper window displays a camera half-image with a detected and focused marker. At the upper left corner of the image the distance between marker and camera is displayed (here 1.84 meter). The lower window contains the corresponding laser scan. A laser scan gives a set of 361 distance readings distributed over 180°. These readings are converted to 2D points (tiny black points). Based on these points, the remaining area is

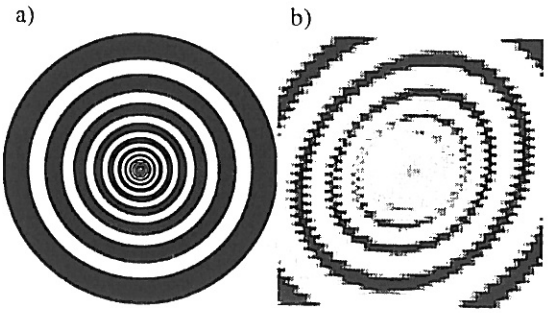

Figure 3. a) Ideal marker, b) Observed marker with effects like spatial quantization, blooming, sensor noise, shifts between half-images, geometric distortion, etc.

classified as free (gray) or as unknown (white). The black cross indicates the point which corresponds to the center of the camera image.

3. COMPONENT ALGORITHMS

3.1 Marker Localization

For the fast and robust detection and localization of potential foci of attention within natural scenes, a combination of a simple artificial marker and a corresponding localization algorithm were developed.

A fast but robust detectable marker should be designed to have properties that reduce the degrees of freedom in the search space, i.e. to have characteristic features that are distance and orientation invariant. Figure 3 shows the chosen marker. Table 1 gives a list of the degrees of freedom and relates them to the methods chosen to deal with them. These methods apply to the design of the marker, the design of the algorithm and to the usage of features of the camera system.

The basic idea of the marker is that for any line intersecting the center of the marker, the marker will cause characteristic local frequency amplitudes, which are symmetric to the center (see Figure 4). The distribution of these frequency amplitudes is independent from the orientation and position of the marker. The amplitudes along the line can be computed by computing the absolute value of the Discrete Fourier Transform on local windows of sufficient length along the line.

$$\text{quality}(w) = \left\| \sum_{k=0}^{l-1} e^{-i2\pi \frac{nk}{l}} \cdot \mathrm{v}(w,k) \right\| \qquad (1)$$

The function $\mathrm{v}(w,k)$ represents the gray values in the window w at position k. The indices n and l describe the number of oscillations within the window and the length of the window respectively.

For the sake of speed, only one Fourier coefficient is computed, which is expected to have a maximum

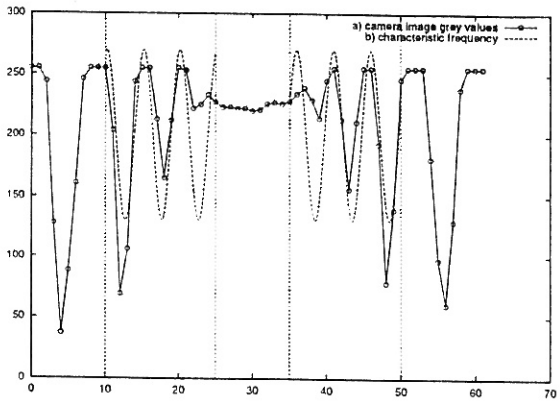

Figure 4. Measured grayscale intensity at a horizontal line intersecting the center of a marker (a). The location of the characteristic complex frequency (b).

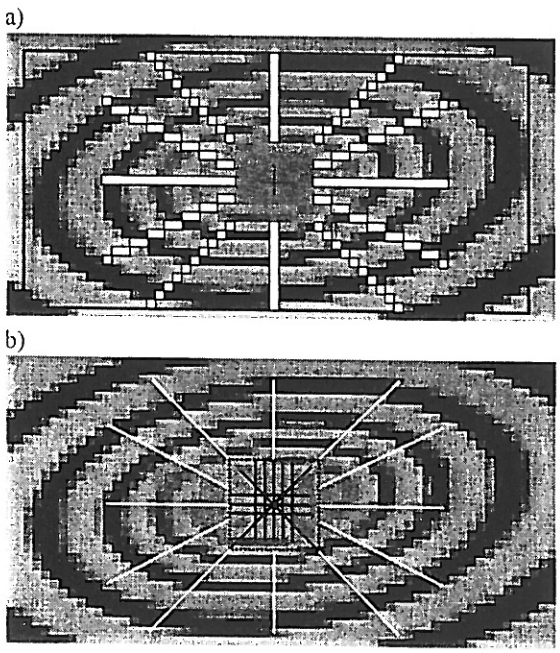

Figure 5. a) Placement of the line-regions that have to exhibit the characteristic oscillation (white lines) centered to the blob boundaries (black rectangle), b) additional line-regions where no such oscillation must occur (black lines) and the center region which has to have low dynamics (rectangle).

value at a characteristic distance from the center of the marker.

To speed up computation, blob analysis is done first to obtain a small number of blobs, of which the centers might be the center of a marker. The marker center is the area around the center point of the marker, which is big enough to detect the characteristic frequencies. Marker hypotheses are then tested for the

degrees of freedom of marker	marker design	camera hardware	algorithm
distance	center is periodic repetitive when scaled		Search for a specific Fourier coefficient at a given distance from the center of the marker.
translation - horizontal - vertical	extensive areas		Full search, speeded up by fast blob analysis for center of marker hypotheses.
orientation - rotation - pan - tilt	rotational symmetry; amplitudes of frequencies along a line intersecting the center of the marker are orientation independent. (note: phase is dependent)		Search for a high amplitude of a specific Fourier coefficient at a given distance from the center of the marker.
light conditions - brightness - contrast - blooming - shadows - reflections - color errors	maximum contrast (B/W) more black than white non-reflective surface no color	automatic gain control	Local brightness normalization with local window of the size of the marker center.
movement - smearing - flickering	maximum contrast (B/W)	short exposure time flickerfixer	Use only half-frames.
occlusion	area around center of marker needed for detection chosen as small as possible		Area around center of marker needed for detection chosen as small as possible.
focus/sharpness	maximum contrast (B/W)	small focal length, focused to infinity	

Table 1. Degrees of freedom when searching a marker within an image and the ways to cope with them.

existence of the characteristic frequencies in all the windows shown by Figure 5a and the non–existence of this frequency at a number of windows at the marker center shown by Figure 5b. A tiny percentage of detection correctness is added by checking whether the dynamics of the center gray values is low.

A valid marker has to fulfill the following equations, whereby *quality*, *antiqual* and *dyn* describe thresholds. The function $\mathrm{v}(x, y)$ in the last equation represents the gray values in the center window of the marker (see the dotted rectangle in figure 5b).

$$\min_{w}(\mathrm{quality}(w)) > quality \quad (2)$$

$$\max_{w}(\mathrm{quality}(w)) < antiqual \quad (3)$$

$$\max_{x,y}(\mathrm{v}(x,y)) - \min_{x,y}(\mathrm{v}(x,y)) < dyn \quad (4)$$

Blob analysis is made to be robust against varying brightness conditions by doing a fast local brightness normalization with local windows of approximately the size of the marker center. Still, blob analysis is sensitive to occlusion, so the size of the marker center was chosen as small as possible. This is a tradeoff between the necessary marker center size and the reliability of classification results.

For the current implementation, the center size is 40x30 pixel on a half-image (see figure 4 and 5). The detection is stable to rotation of the marker up to more than 60° (pan and tilt). The maximum detection range is generally unlimited and depends on the size of the marker and the focal length of the camera lens. At close range, detection may fail because of decreasing image sharpness, depending on the camera system used.

Current studies show that an alternative way to detect the markers in an image is to change the order of evaluating the image as follows: doing a one-coefficient Fourier analysis on the whole image, combining the results analogous to Figure 5 and finally determine the centers of the markers by blob analysis. Local brightness normalization would no longer be necessary and due to the missing preselection step, this should be more robust to occlusion.

3.2 Distance Computation

Computation of the distance d of the center point of the camera image from a laser scan is based on the fixed and given geometry of the camera in relation to the laser scanner (see Figure 6a). The optical axis y_{camera} of the camera is within the scanning plane of the laser scanner and parallel to the central scan direction y_{laser} of the laser scanner. The distance k of y_{camera} and y_{laser} is given.

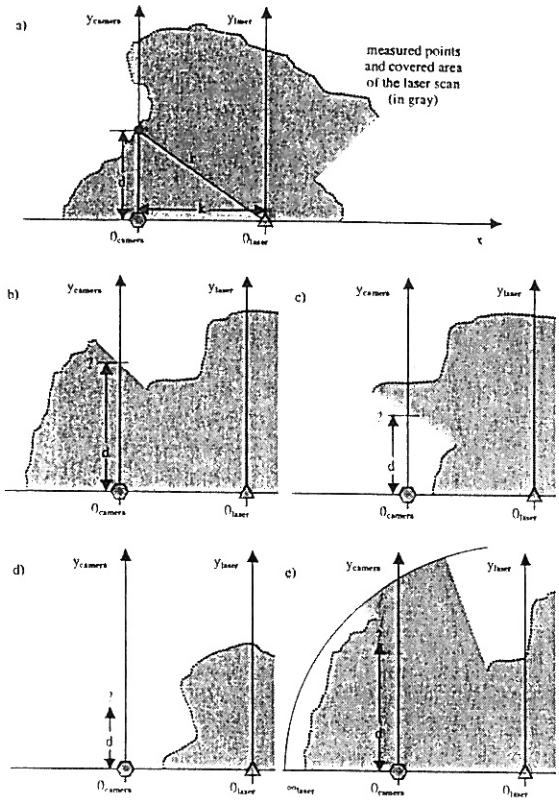

Figure 6. Different situations for computing the distance to the point of interest. (see text for further explanations)

The basic principle of the algorithm is to look for all measured points close to y_{camera} and consider the closest distance of these points to the camera to be d. In some cases, this assumption is not valid. But normally, it is still possible to compute an upper or lower bound for d, which could be useful for collision avoidance or deciding reachability. Figure 6 shows the different forms of occlusion, which can occur:

a) The relevant point is not occluded in the laser scan; this is the usual case, because k is small compared to d.

b) The relevant point is occluded in the laser scan, but it is still possible to assume a lower bound for d.

c) The relevant point may be occluded in the laser scan, but it is still possible to assume an upper bound for d.

d) The relevant point is occluded in the laser scan; no information about d is available.

e) The relevant point is out of the maximum range of the laser scan, which gives a lower bound for d.

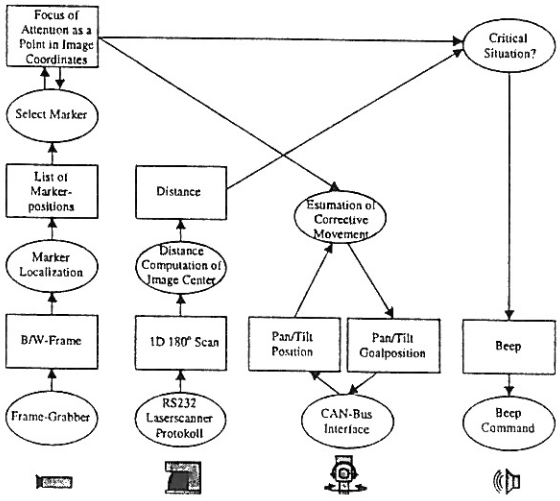

Figure 7. Architecture of GLaRE seen as a continuous, partially cyclic data flow between modules (ellipses) connected via state representations (boxes).

This method needs calibration of camera and laser scanner directions, but given the size of a marker of about 40 pixel in diameter, this is not critical.

4. ARCHITECTURE OF GLARE

Figure 7 presents the complete architecture of GLaRE in form of a continuous data flow. The main modules and the sensors and actuators are shown. Modules are depicted by ellipses. State representations are represented by boxes.

State representations are data structures containing application specific data. They form flexible interfaces. Modules contain basic algorithms, which continuously map data between state representations. Any number of modules may read a state representation, but only one module has write access. So, results of algorithms can be shared freely. Continuously means that every module will not necessarily wait for new data to arrive, and data will not queue up if it can not be processes fast enough. This allows loosely coupled modules and multi-rated systems. E.g. our frame grabber and laser scanner can not be triggered or synchronized and have to generate their data asynchronously. Timestamps can be used to avoid multiple processing of elements and to detect critically out of date data.

Modules within the graph may again consist of a continuous data flow. Figure 8 is an example for this, presenting the subsystem for marker localization of GLaRE. The interfacing state representations can be found in both the graph and the subgraph.

Such a continuous data flow model, in which modules are connected via state representations seems to be appropriate to describe the functional models and their interactions [3]. Similar mechanisms are already used

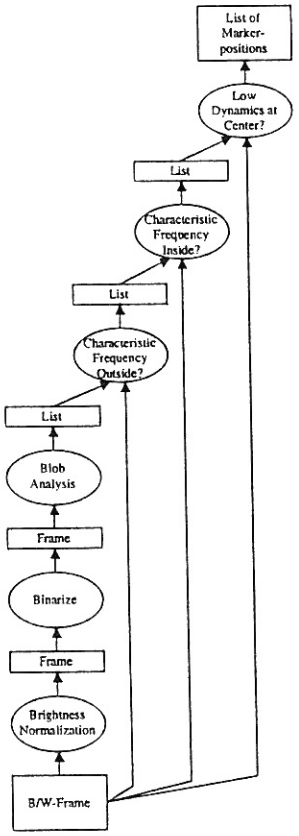

Figure 8. Subsystem "Marker Localization" of GLaRE seen as a data flow.

in some robot architectures. [4] and [11] use a very centralized representation. The Flip-Tick-Architecture FTA [13] suggests asynchronous cyclic execution in random order for robotics application [8].

With the current implementation, a marker with 20cm diameter moving with walking pace can be tracked in a distance range of ca. 0.5 to 5 meters. The marker detection including local brightness normalization (100 ms), blob analysis (50ms), and marker hypothesis verification (0-50 ms) takes from 150 to 200 milliseconds on an Intel Pentium II processor with 300 megahertz. Timing measurements depend critically on the used system. I.e. changes in cache size causes changes in computation time up to a factor of four.

5. CONCLUSION

The GLaRE behavior for active distance measurements was presented as a (re-)usable building block for service robot architectures. GLaRE uses attention-based algorithms to detect interesting areas in the environment, for which a distance profile has to be determined.

For the implementation an artificial marker as point of attention and a suitable detection algorithm were introduced. The architecture of the building block GLaRE was presented.

REFERENCES

[1] C. Balkenius, L. Kopp. Visual Tracking and Target Selection for Mobile Robots. In *IEEE Proceedings of EUROBOT'96*, pages 166–171, 1996.

[2] J. Borenstein, H. R. Everett, L. Feng. *Navigating Mobile Robots – System and Techniques*. A K Peters, Ltd., Wellesley, Massachusetts, 1996.

[3] R. A. Grupen, M. Huber, J. A. Coelho, K. Souccar. A Basis for Distributed Control of Manipulation Tasks. In *IEEE Expert. Special Track on Intelligent Robotic Systems*,10(2):9–14, 1995.

[4] K. Konolige, K. Meyers, E. Ruspini, A. Saffiotti. The Saphira Architecture: A Design for Autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(1):215–235, 1997.

[5] O. Lang, A. Gräser. Regelung eines teilautonomen Roboters mittels Zoomkamera. In H. Wörn, R. Dillmann, D. Henrich, editors, *Autonome Mobile Systeme 1998*, pages 45–56. Springer, 1998.

[6] A. Longacre. Emerging Barcode Symbologies. In *AIM International*, Technical Review, pages 61–67, 1996

[7] B. Mertsching, M. Bollmann. Visual Attention and Gaze Control for an Active Vision System. In N. Kasabov, R. Kozma, et al., editors, *Progress in Connectionist-Based Information Systems*, pages 76–79. Springer, 1997.

[8] G. Richter, F. Smieja, U. Beyer. Integrative architecture of the autonomous hand-eye robot JANUS. In *International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97)*, pages 382–389, Los Alamitos, 1997.

[9] T. Trittin, A. Gräser. Methode der virtuellen Punkte zur autonomen, bildbasierten Roboterregelung. In H. Wörn, R. Dillmann, D. Henrich, editors, *Autonome Mobile Systeme 1998*, pages 65–75. Springer, 1998.

[10] T. Uhlin, K. Johansson. Autonoumous Mobile Systems: A Study of Current Research. *Technical Report ISRN KTH/NA/P–96/03*, The Royal Institute of Technology, Stockholm, Sweden, 1996.

[11] G. Wasson, R. P. Bonasso, D. Kortenkamp. Integrating Active Perception with an Autonomous Robot Architecture. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 325–331,1998.

[12] B. H. Yoshimi, P. Allen. Closed-Loop Visual Grasping And Manipulation. *http://www.cs.columbia.edu/robotics/ publications/*,1996.

[13] H. Veit, G. Richter. A generic architecture for distributed systems of interacting medium-grained agents. In P. Kacsuk, G. Kotsis, editors, *Proceedings of DAPSYS 98*, pages 61–67. University of Vienna, 1998.