ROBOT NAVIGATION METHODOLOGY FROM AERIAL IMAGERY

GERALDO F. SILVEIRA*,†, JOSÉ R. H. CARVALHO†, RUBENS MACHADO†, SAMUEL S. BUENO†

†*Robotics and Computer Vision Laboratory/ITI*
PO Box 6162, 13081-970, Campinas-SP, Brazil
`Fist_name.Last_name@iti.gov.br`

**UNICAMP – State University of Campinas*
PO Box 6101, 13081-970, Campinas-SP, Brazil
`silveira@dsce.fee.unicamp.br`

Abstract— This work presents a mathematical morphology-based three-level framework for ground robot navigation based on images acquired by the aerial vehicle of an air-ground robotic ensemble. The three levels correspond to image processing, path planning, and path following subsystems. Experimental results with different robots and an overhead camera illustrate the validity of the proposed method.

Key Words— Mobile robot, vision-based navigation, vision-based control, mathematical morphology.

1 Introduction

Robotic systems can be conceptually combined in many ways. To address complex tasks efficiently, the highlighted capabilities of each individual robot should be readily emphasized; we suggest that air-ground robotic ensembles specifically are of substantial interest for a large class of field applications such as surveillance tasks (Elfes et al., 1999). In particular, hazardous material inspection and handling can benefit from the partition of responsibilities, where one of the robots provides broad visual coverage while the other executes close-up inspection and manipulation.

Practical applications of mobile robots require the analysis of dynamic and kinematic modeling, sensing, spatial representation of the robot and its environment, control, navigation and path planning (Silveira et al., 2001). One of problems that have to be addressed in air-ground robotic ensembles is navigation of the ground robot based on imagery acquired onboard the aerial vehicle. This problem is of significance when the sensors and cameras of the ground vehicle cannot provide enough information to locate a target which can be spotted from the air, such as when one is searching for missing people or buried land mines.

This article considers the mathematical morphology (MM) framework developed to address this problem (Carvalho et al., 1999). It consists of a three-level structure formed by an image processing level (Level 1), a path planning (Level 2), and a path following level (Level 3). This work focuses on the first and second levels. A very interesting solution for the closed-loop control of mobile robots is based on the well known Model-Based Predictive Controller (MBPC) (Oliveira and Carvalho, 1999). Experimental results using ground mobile robots (*Khepera* and XR4000) and an overhead camera illustrate the validity of the proposed method.

Table 1. Description of the used MM operators

Operator	Description
mmclose(X,SE)	morphological closing of the image X by the structuring element SE
mmareaclose(X,A,SE)	remove any pore with area less than A of the image X according to SE
mmareaopen(X,A,SE)	remove any grain with area less than A of the image X according to SE
mmthreshad(X,min,max)	adaptive threshold of image X according to min and max boundaries
mmero(X,SE)	erosion of image X by the SE
mmunion(X,Y)	pixel-wise union of the images X and Y
mmaddm(X,Y)	pixel-wise sum of the images X and Y
mmsubm(X,Y)	pixel-wise subtraction of the images X and Y
mmgdist(X,Y,C,M)	geodesic distance transf. from image X relative to Y according to connectivity C and metric M
mmskelm(X,SE)	skeleton of image X by the SE

This paper is organized as follows: Section 2 briefly presents the concepts and MM operators. In Section 3 the three-level framework is presented while its implementation details are discussed in Section 4. In Section 5 experimental results are shown. The article is concluded in Section 6.

2 Mathematical morphology

In this section we summarize the MM concepts and operators utilized throughout the article. General concepts regarding digital images, such as connectivity, and image coordinates are beyond the scope of this article. See, for example, (Barrera et al., 1994) for a review on these subjects.

Mathematical morphology is a powerful tool for digital image processing. Opposite to the classical definition of an image as an amplitude function of its coordinates, MM treats the image as a set of pixels. Therefore, MM operators are defined in the context of set theory. The fundamental operations associated with a set are the standard operations: *union* ($\cup$), *intersection* ($\cap$), and *complement* (^{c}). The MM operators utilized here are available commercially from the SDC Mathematical Morphology Matlab Toolbox (*SDC Morphology Toolbox for MATLAB 5*, 2000), and listed in the table 1.

3 MM-Based robot navigation

We are interested in navigating a mobile robot utilizing aerial images of its environment. In the case of indoor navigation, cameras positioned under the roof map the robot's working space. In the case of outdoor navigation, an UAV equipped with a camera sends images of the ground to the robot. In both cases, a 640×480 maximum resolution 2D mapping with 16 bits grey-scaled is used in the representation of pixel intensity.

The three-level MM-based robot navigation method is presented in this section. Figure 1 summarizes the basic framework. In the upper level, image processing locates the robot and its target, while building a map of the available free space. The intermediate level uses this information to plan a 2D path in the robot's environment. Finally, the lower level navigates the robot along the path taking into account kinematic and dynamic constraints.

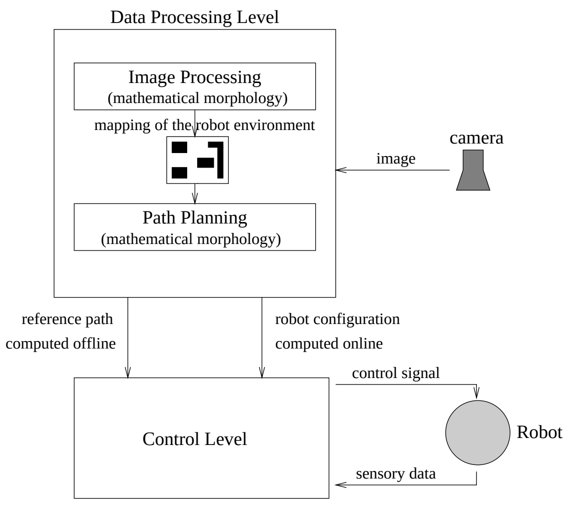

Figure 1. Three-level structure of the visual-based navigation framework.

3.1 *Upper level: Image Processing*

The image processing level is responsible for:

- locating the ground robot in the aerial images, returning the set of pixels that best represents it. For cylindrical robots, with center of rotation located in the center of their circular section, it is sufficient to return the pixel that contains the center of the robot. The same is valid for squared-section robots;
- building a map of the robot's environment, providing information about location of obstacles and free space. This may be either a binary image (pixel occupied/pixel empty) or a gray-scale image with pixel intensities defining the probability of pixel occupation;
- locating the target, if necessary, returning the set of pixels that best represents it. In some problems, the target is a predefined position (or configuration) in the environment. In this case, this level will only have to return the corresponding pixel. On the other hand, the target may be an object to be found. In this case the upper lever considers the target as one more feature to be detected in the image.

For most real cases, it is not trivial to detect obstacles using pure visual information. The lack of knowledge of objects' characteristics like shape, color, and texture, makes the distinction from shadows and reflections on the ground a hard task to be performed. In some specific problems, it is easier and more reliable to work over binary information. In the robotic soccer game (FIRA, 2001), for example, shapes and colors of the robots and of the ball, and the shape, color, and dimension of the field are previously known and the luminosity is controlled. This is a problem where the image features can be easily detected.

The framework presented here, however, permits the utilization of information from local sensors in order to improve the mapping. Moreover, the lattice characteristics of an image and its intensity information make the gray-scale image information very suitable to be used with other mapping techniques, such as occupancy grids (Elfes, 1987).

3.2 *Intermediate level: Path Planning*

Given the images returned by the image processing level, in this intermediate level MM tools are used to:

- find a set of pixels in the free space that connects robot and target positions using a minimum distance criterion. This set of pixels (if it exists) will be called a *channel*. The level has to return an image with the channel or a blank image if no connection is found;
- extract either a particular pixel-wise sequence in the channel linking robot and target positions (called *path*) or just a collection of control pixels (called *way points*).

3.3 Lower level: Path Following

Given the information returned by the intermediate level, MM tools are used to provide an on-line updating of the current robot position to be included in a closed-loop control algorithm. For many applications, an updating of the robot position is sufficient.

The most popular methods for giving robot on-line position estimates are based on dead reckoning, which lead to accumulative errors (Borestein et al., 1996). In this case, MM tools are used to correct the robot position, augmenting the life-time of robot navigation. An interesting issue, not addressed in this work, is the fusion of visual information with others sensorial data to provide the best possible estimation of the robot position and orientation in real-time.

This level is also responsible for tasks which are not computed by MM operators:

- suitably map the Cartesian coordinate system metrics and the pixel coordinate in the image coordinate system;
- deal with robot dynamic and kinematic constraints during the motion along the path, in the case of non-holonomic robots (e.g., cart-type robots);
- deal with local obstacle avoidance. In this case, when the robot local sensors detect a collision route to an *unexpected* obstacle, it enters in an alternative navigation mode, until the upper level recomputes the path considering this new information;
- compute the real-time control signals to be sent to the robot platform and process robot local sensor data.

4 Implementation

This section presents the implementation details of the framework discussed above for the first and second levels.

4.1 Image processing

To implement the image processing level, the following MM-based algorithm is utilized.

Step 1. Locate the robot on the scene S. This is done by applying the following MM operators:

$$\begin{aligned} &S\_closed = mmclose(S, SE_{box}); \\ &ac\_S\_closed = mmareaclose(S\_closed, A_1); \quad (1) \\ &sub = mmsubm(ac\_S\_closed, S\_closed); \end{aligned}$$

and finally obtain the robot through:

$$Robot = mmareaopen(mmthreshad(sub, th), A_2); \quad (2)$$

where th is the threshold level, A_1 is the area of the largest blob in the image, A_2 is the maximum area of present noise and SE_{box} is the 3×3 square centered at the origin.

The sequence of filters above is better understood by the illustration in the Fig. 2.

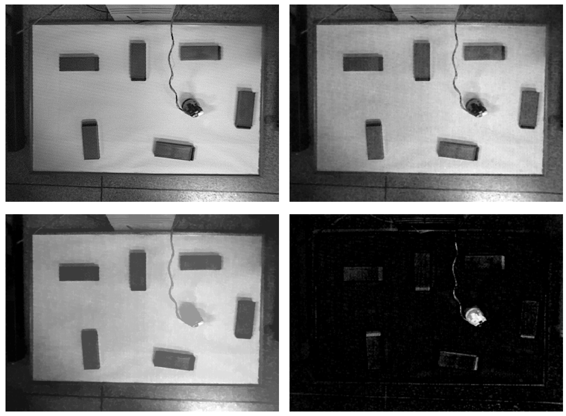

Figure 2. Sequence of transformations utilized to implement (1) and (2). From up to down and left to right one has the initial image, the image after the closing operation, after removing the robot and the robot itself.

To locate the best approximation of the center of the robot, we compute:

$$Robot_c = mmlastero(Robot, SE_{box}); \quad (3)$$

Step 2. Extract the obstacles.

$$I = mmthreshad(ac\_S\_closed, th); \quad (4)$$

The figure of the extracted obstacles is depicted in Fig. 3.

Figure 3. Detection of the obstacles by the image processing level, extracted directly from Fig. 2-c.

Step 3. Send the binary image I and $Robot_c$ to the path planning level.

4.2 Intermediate level implementation

Given a binary representation of free space, obstacles and the robot's initial and desired configurations, the following MM-based algorithm is used to extract a feasible 2D path.

Let I be the binary mapping of the robot environment.

Step 1. Compute the erosion of image I with a disk structuring element SE with radius corresponding to the minimum allowed distance in pixels from the center of the robot and an obstacle.

$$I_E = \text{mmero}(I, SE); \tag{5}$$

Step 2. Compute the gray-scale distance image from the starting pixel P_i and all valid (non-obstacle) pixels:

$$D_{if} = \text{mmgdist}(I_E, P_i, connectivity, M) \tag{6}$$

where $connectivity = 4$ or 8 and M may be the
EUCLIDEAN metric.

Step 3. Repeat Step 2 for the final pixel P_f:

$$D_{fi} = \text{mmgdist}(I_E, P_f, connectivity, M) \tag{7}$$

Step 4. Compute the pixel-wise sum of D_{if} and D_{fi}

$$D = \text{mmaddm}(D_{if}, D_{fi}) \tag{8}$$

Step 5. Compute the binary image

$$Min_D = \text{mmtreshad}(D, D(P_i), D(P_f)) \tag{9}$$

where $D(P_i)$ is the intensity of image D at pixel P_i, to obtain an image channel connecting initial and final pixels.

Step 6. Compute

$$MIN_D = \text{mmskeleton}(Min_D) \tag{10}$$

to extract a path linking initial and final pixels.

Some remarks about this implementation are necessary.

- Step 1 transfers to the environment the robot dimensions, reducing it to a pixel, at the same time that extracts the free space;
- Step 2 and 3 generate two gray-scale images where pixel intensities represent the distance in pixels from initial and final pixels. When computed the sum (Step 4), the resulting gray-scale image has the characteristic that, if a path does not exist, all pixels have the maximum value (an white image).
- If a path exists, the intensity of initial and final pixels in D are equal to the minimum distance between them and are the darker pixels of the image.

The next result will be presented without proof. They can be derived from the connectivity and metric definitions in digital images, see (Barrera et al., 1994).

Lemma 1 *Let p_i and p_f be any two connected pixels in a binary image I according to some connectivity. Let also D_{if} (resp. D_{fi}) be a grayscale image obtained by the distance transform from p_i (resp. p_f) to every connected pixel in I. Then, the distance from p_i to p_f is equal to $D(r_i, c_i) = D(r_f, c_f)$, where $D = D_{if} + D_{fi}$ and (r_i, c_i) (resp. (r_f, c_f)) are the coordinates of p_i (resp. p_f).*

Corollary 1 *Let P_D be any minimum path in Min_D. Then, for metrics based on 4-connectivity and 8-connectivity, all inner elements of P_D have the same value, equal to $d_{fi} = d_{if}$.*

Corollary 2 *For the Euclidean metric, all inner elements of Min_D with value equal to $d_{fi} = d_{if}$ are sufficient to navigate the robot. These are potential candidates for control points.*

For any cylindrical robot (omnidirectional or not) with center of rotation located in the center of its circular section, this planner, together with the controller described in the sequel, provides the complete solution. □

4.3 Lower level implementation

Most research robots are non-holonomic platforms. Therefore, the 2D planner provided by the intermediate level is not sufficient to move the robot among obstacles safely, what makes necessary to consider robot maneuvers.

To overcome this problem, the responsibility in controlling the robot is divided. While the planner provides a reference 2D path, which is sufficient for omnidirectional platforms, the path following (control) level considers the robot's particular dynamic and kinematic constraints.

The current implementation is based on the well known Model-Based Predictive Controller

(MBPC) approach. MBPC methods are based on the prediction of the future behavior of the process by using a model. Then, based on the well known kinematic model of the unicycle-type robot, a discrete-time model is obtained in order to compute the j-step ahead prediction equation. The reader may refer to (Oliveira and Carvalho, 1999) for further information since the control aspects are not the main issue of this paper.

The framework, however, allows for the inclusion of any other control method, such as (Sørdalen, 1993) and (Samson and Ait-Abderrahim, 1990). Indeed, in the future the authors will create a library of control algorithms so as to deal with any type of mobile robot (omnidirectional, cart or car) and control scheme.

5 Experimental results

This section presents the experimental results obtained with the implementation of the proposed framework. The experimental setup consists of a mobile robot Khepera (K-Team S/A, 1997) connected to a Sun UltraSparc 1 workstation. The image is captured by an off-the-shelf camcorder. The communication with the robot is implemented in C and encapsulated by mex-functions called inside MATLAB.

The robot's environment is a flat table populated by obstacles. Figure 2-a shows the actual scene image. The robot's initial and final orientations were arbitrarily assigned. the target location is set according to the task to be performed by the ensemble.

Figure 4 shows the output of the image processing level, with the initial and goal positions (the two small circles). Observe that the Khepera could reach the final position through several different manners. However, it has found the path, see Fig. 5, that corresponds to the minimum distance way to goal, achieved by using our MM-based methodology. The sequence of pixels of the minimum path is then sent to the MBPC controller, shown in the Fig. 6.

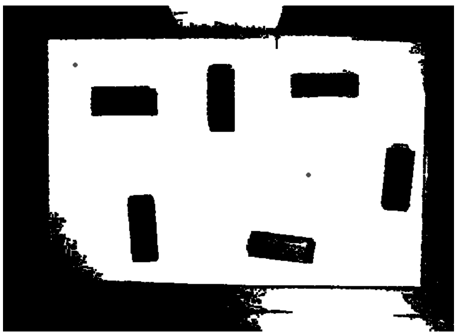

Figure 4. Output of the image processing level overlapped with the initial and final position markers.

An example with the XR4000 navigating in

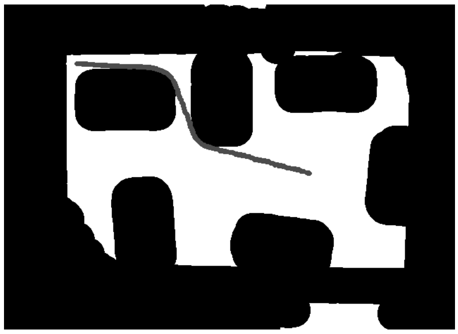

Figure 5. Path planning results.

an outdoor environment is shown in Fig. 7 (Elfes et al., 1999), achieved through the use of the framework described here. Note that the thin line is not the robot's actual path, but simply the planned path. Experiments with this scenario are currently under way.

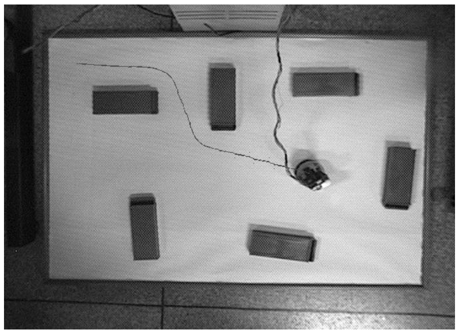

Figure 6. Original scene and the planned trajectory sent to the third level.

Figure 7. Path planning for a mobile robot in an outdoor environment.

6 Discussion

In this paper, a practical implementation of a framework for vision-guided mobile robot navigation is discussed. It is conceived as a very general and flexible three-level hierarchical structure. The two upper levels are entirely computed by MM tools. The framework integrates image processing, path planning, and closed-loop path following for mobile robots.

The responsibility for moving the robot is divided between the planning and the controller. This permits the use of the same methodology for omnidirectional and non-holonomic robots.

The framework may be implemented with many setups and conditions. It also permits the use of MM tools along with others robotic techniques, including probabilistic maps and local obstacle avoidance methods, which constitute the next topic to be investigated.

In the example presented, we mimic the air-ground robotic ensemble concept with a laboratory setup consisting of a Khepera robot and an overhead camera. Experimental results with the actual ground robot and air vehicle will be shown in future works.

Acknowledgments

This work is partially funded by FAPESP under grant nos. 97/13384-7, 98/13562-5, and 98/13563-1 and by CNPq/CTPETRO under grant no. 466713/00-2.

References

Barrera, J., Banon, J. F. and Lotufo, R. A. (1994). Image algebra and morphological image processing V, *International Symposium on Optics, Imaging and Instrumentation, SPIE's Annual Meeting*, San Diego, USA.

Borestein, J., Everett, H. R. and Feng, L. (1996). *Where am I? Sensors and methods for mobile robots positioning*, J. Borestein, Ed., The University of Michigan, Michigan, USA.

Carvalho, J. R. H., Elfes, A., Bergerman, M., Bueno, S. S. and de Paiva, E. C. (1999). A methodology for aerial imagery-based ground robot navigation, *15th International Conference on CAD/CAM Robotics & Factories of the Future*, Águas de Lindóia - Brazil.

Elfes, A. (1987). Sonar-based real-world mapping and navigation, *IEEE Journal of Robotics and Automation* **RA-3**(3): 249–265.

Elfes, A., Bergerman, M., Carvalho, J. R. H., de Paiva, E. C., Ramos, J. J. G. and Bueno, S. S. (1999). Air-ground robotic ensembles for cooperative applications: Concepts and preliminary results, *1999 International Conference on Field and Service Robotics*, Pittsburgh-USA, pp. 75–80.

FIRA (2001). International robot-soccer association federation, *http://www.fira.net*.

K-Team S/A (1997). Web page of the company, *http://www.k-team.com*.

Oliveira, G. H. C. and Carvalho, J. R. H. (1999). Trajectory tracking for nonholonomic mobile robots using predictive methods, *1999 European Control Conference*, Germany.

Samson, C. and Ait-Abderrahim, K. (1990). Mobile robot control, part 1: Feedback control of a nonholonomic wheeled cart in cartesian space, *Technical report*, INRIA, Centre de Sophia-Antipolis.

SDC Morphology Toolbox for MATLAB 5 (2000). *http://www.mmorph.com*.

Silveira, G. F., Carvalho, J. R. H., Madrid, M. K., Bueno, S. S. and Rives, P. (2001). Towards vision guided navigation of autonomous aerial robots, *Proceedings of the IV Brazilian Symposium on Intelligent Automation*, Canela/RS, Brazil.

Sørdalen, O. J. (1993). *Feedback Control of Nonholonomic Mobile Robots*, PhD thesis, The Norwegian Institute of Technology, Trondheim, Norway.