The Spatial Semantic Hierarchy implemented with an omnidirectional vision system

Emanuele Menegatti [a] Giovanni Aneloni [a] Mark Wright [b]
Enrico Pagello [a,c]

[a] *Department of Informatics and Electronics, University of Padua, Italy*

[b] *Edinburgh Virtual Environment Center, University of Edinburgh, UK*

[c] *Institute ISIB of CNR, Padua, Italy*

Abstract

In this paper, we propose a new approach to the map building task: the implementation of the Spatial Semantic Hierarchy (SSH), proposed by B. Kuipers, on a real robot fitted with an omnidirectional camera. The original Kuiper's formulation of the SSH was slightly modified, in order to manage in a more efficient way the knowledge the real robot collects while moving in the environment. The sensory data experienced by the robot are transformed by the different levels of the SSH in order to obtain a compact representation of the environment. This knowledge is stored in the form of a topological map and, eventually, of a metrical map. The aim of this paper is to show that a catadioptric omnidirectional camera is a good sensor for the SSH and nicely couples with several elements of the SSH. The panoramic view and rotational invariance of our omnidirectional camera makes the identification and labelling of place a simple matter. A deeper insight is that the tracking and identification of events on an omnidirectional image such as occlusions and alignments can be used for the segmentation of continuous sensory image data into discrete topological and metric elements of robot maps. Such a combination of the SSH and omnidirectional vision provides a powerful general framework for robot map making and indeed new insights into the concept of "place" in such activities. Some preliminary experiments performed with a real robot in a unmodified office environment are presented.

Key words: Spatial Semantic Hierarchy, omnidirectional vision, map building, localization
PACS:

---

* Emanuele Menegatti
Email address: `emg@dei.unipd.it` (Emanuele Menegatti).
URL: `http://www.dei.unipd.it/ricerca/airg` (Emanuele Menegatti).

Preprint submitted to Elsevier *22 May 2007*

1 Introduction

In several application a mobile robot does not need a map of the environment to perform its tasks. This is especially true if the environment is very simple or highly engineered and the robot can move using some form of reactive strategy. However, if the robot's task requires an *understanding of the world*, the robot has to answer the three questions posed by Levitt and Lawton [21]: "Where am I?", "How do I get to other places from here?", "Where are other places relative to me?". In other words, the robot needs some kind of map of its *world*. Since the beginning of mobile robotics, the map building problem has been a fundamental problem [12]. A wide spectrum of solutions have been proposed using a wide range of sensors. There is a wide range of different maps a robot can use. Different kinds of maps answer the three basic questions using different properties of the environment. We have to keep in mind that the distance which separates two objects is only one of the properties of the space in which the two objects are immersed. The choice of which property to exploit, and therefore, which kind of map to use, depends on the task of the robot.

Metric maps and qualitative maps are two extreme examples of this idea. They use very different properties of the space. In metric maps the space is represented in a single global coordinate system. The relations between different places are metrical relations described in terms of measures of distances and angles. Conversely, in qualitative maps, the environment is represented as a set of *places* connected by *paths*. There is no metric or geometric information, such as distances, angles, etc., but only the notion of proximity and order [12]. Depending on the robot's task qualitative maps can use very different representations [3] [7].

One of the most effective qualitative representations of an environment is the so called *topological map*. This is a qualitative map which extracts from the environment the topological relationships between the different places and paths. One of the key issues in the generation of topological maps is the abstraction of a discrete set of distinct places from the continuous sensorial experience. Topological maps can be transformed into metric maps by adding metric information to the places and to the relationships between paths and places. Therefore, a map can be seen as a hierarchal structure built layer by layer. Benjamin Kuipers created a formalisation of this intuition: the Spatial Semantic Hierarchy (**SSH**). To the best of our knowledge, so far the SSH was only implemented either on simulated robots or on real robots with very simple sensors, like sonars. No attempt to use a vision sensor has been made. In the last years, omnidirectional vision systems have been exploited successfully in robot navigation and map building [33]. The success of this kind of sensors is explained by the wide field of view achievable [10]. Omnidirectional cameras offer in one shot a global view of the surroundings. The purpose of this paper is to present an implementation of the SSH on an autonomous robot fitted with an omnidirectional camera in order to build a map of a building. Our final aim is to use this approach to create a SLAM (Simultaneous Localization and Mapping) algorithm based on the framework of the SSH combined with an omnidirectional vision system. The preliminary experiments presented in this work show the approach is sound and promising. In fact, event though the robot infers the environmental structure from the vision data only, without exploiting the information on the robot motion coming from the encoders, the environment structure is correctly retrieved.

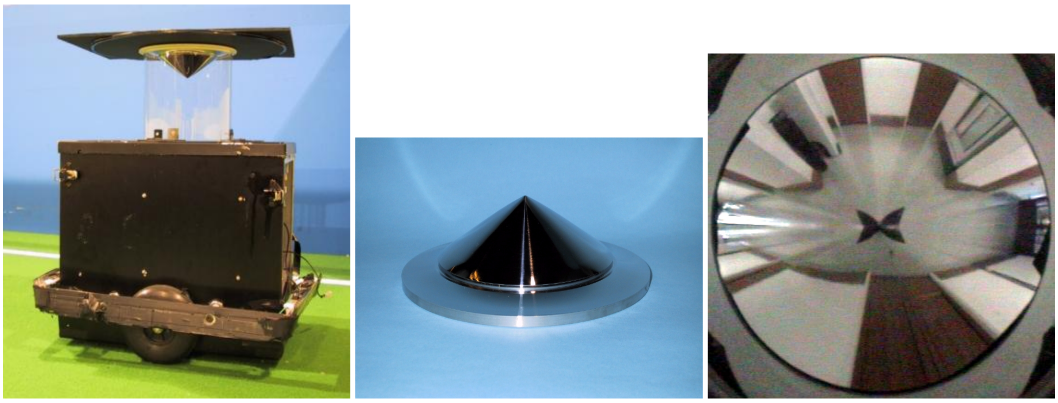
However, at this stage of the work our results are not comparable with the results obtained by other SLAM approaches. Just to cite a few of them. Lemair et Lacroix presented a successful approach to 3D bearings-only SLAM using an omnidirectional camera [20]. Salient points are detected and matched between consecuitive images. Se et al [30] used SIFT visual landmarks from the stareo camera to build a 3-D map of the environment and to localize the robot. In [5] localization is obtained via mapping of a sparse set of features observed from a single camera. Kim and Chung used an omnidirectional stereo vision system which combined structure from motion and stereo algorithms [11]. The work presented in this paper was developed as SLAM was emerging. In hindsight, SLAM approaches in combination with powerful feature detectors, such as SIFT, have become the dominant paradigm. However, the framework we propose is independent of feature detection methods, and these could be incorporated into our SSH based approach. For example, planar patches detected by SIFT could be tracked for emergence, occlusion and other topological events.

1.1 The assumptions

The discussion in the next sections and the experiments are based on some assumptions that are worth making explicit here.

- The robot is moving in a indoor environment;
- The objects present in the scene are static: they do not change their positions;
- The floor is almost flat and horizontal;
- The walls and the objects present in the scene have vertical edges and surfaces;
- The robot can only turn on the spot or move along a straight line. It cannot make more complex movements;

The last assumption is strong, but greatly simplifies the image sequence interpretation and the robot control.



(a) (b) (c)

Fig. 1. (a) The robot on which the SSH has been implemented. (b) A closer view of the omnidirectional mirror mounted on the robot.(c) An omnidirectional image grabbed by the robot.

2 The Spatial Semantic Hierarchy

The SSH is a model of the way humans organise their knowledge of a large environment. A large environment is defined by Kuipers as an environment that extends beyond the sensorial horizon of the perceiving agent, i.e. an environment with sections that are not directly perceptible. This model was proposed by Benjamin Kuipers [15] [16] [17] [18] [27] and it is intended to serve as a "method for robot exploration and map building" [14]. The SSH is composed of several layers (see Fig.2): the sensory level, the control level, the causal level, the topological and the metrical level. Each layer can be implemented independently, even if they strongly interact. In the following we will describe each layer in detail:

2.1 The Sensory Level

The sensory level is the interface with the agent's sensory system. It extracts the useful environmental features from the continuous flow of information it receives from the robots' sensors.

2.2 The Control Level

The control level describes the world in terms of continuous actions called control laws. A control law is a function which relates the sensory input with the motor output. Each control law has conditions for its appropriateness and termination. A selected control law is retained until a transition of state is detected. These transitions can be detected with a function called a *distinctiveness measure*. The distinc-

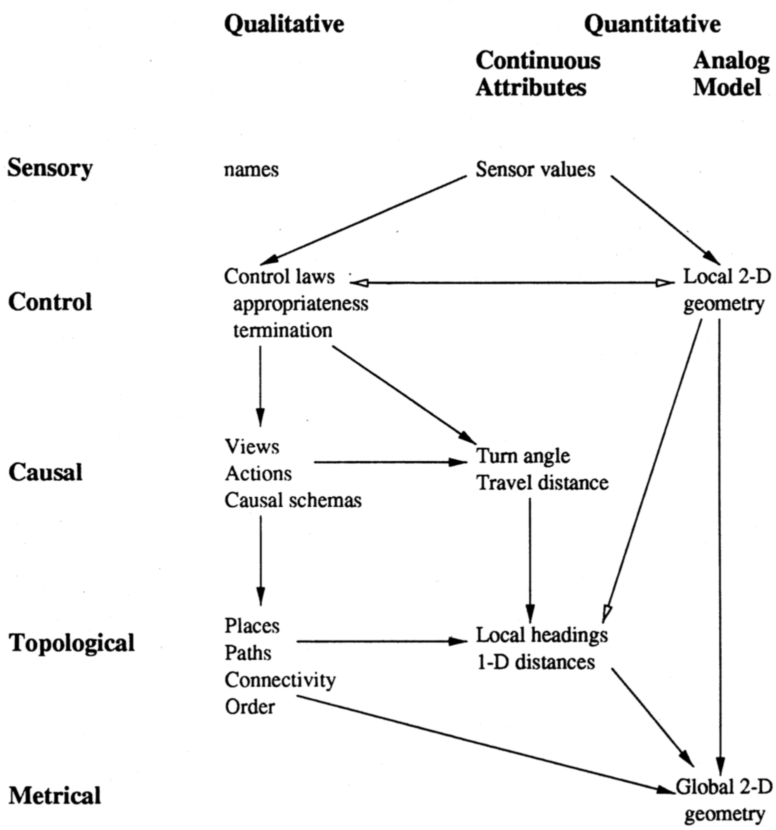


Fig. 2. A graphical representation of the different components of the SSH (from B. Kuipers).

tiveness function must be identified depending on the sensor used and the features which are to be extracted from the environment.

2.3 The Causal Level

The causal level abstracts a discrete model of the environment from the continuous world. This discrete model is composed of *views*, *actions* and the causal relations between them. A view is defined as the sensor's reading at a place where a transition of state is detected. An action is defined as the application of a sequence of control laws. At this stage causal maps and planning are possible using these three basics elements. For this purpose, it is convenient to classify actions into two categories: *travels* and *turns*. "A **turn** is an action that leaves the agent at the same place. A **travel** takes the agent from one place to another" [27].

2.4 The Topological Level

The topological level represents the environments as places, paths and regions, with details of how they are connected or contained one in the other. To use Kuipers's words:

> *The topological model of the environment is constructed by the non-monotonic process of* ***abduction****, infering the minimal set of places and paths needed to explain the regularities observed among views and actions at the causal level.*

2.5 The Metrical Level

The metrical level augments the topological representation of the environment by including metric properties such as distance, direction, shape, etc. At this stage, it is possible to build a global geometric map of the environment in a single frame of reference.

3 Omnidirectional vision and map building

Omnidirectional cameras produce images with a wide angle of view, but with low resolution. As many authors have already observed, this is not a problem in the case of a map building robot [7] [28]. For map building, a camera with high resolution is not so useful. It is not necessary to capture the details of objects and surfaces, but only to estimate their positions and dimensions. By using an omnidirectional camera, the robot does not need to take several shots to understand the surroundings. It does not need to turn and take a look around. It does not need to be fitted with moving parts (camera or mirrors) to increment its field of view. However, beside these that can be seen just as implementation considerations, there are more fundamental aspects supporting the use of an omnidirectional sensor in the process of building a map with the SSH.

For instance, an omnidirectional image captures at once all the objects visible from the robot location, see Fig. 1c. This image has a strict connection with the *views* introduced in the causal level of the SSH, i.e. with the sensor reading at a distinct place.

In addition, for the particulary geometry of our omnidirectional camera, the vertical edges in the scene are mapped on the image plane as radial lines originating from the point corresponding to the tip of the mirror. (The assumption which underlies this fact is that the axes of the mirror and camera are vertical and aligned.) Thus, it is very simple to extract vertical edges from the images by searching the image for

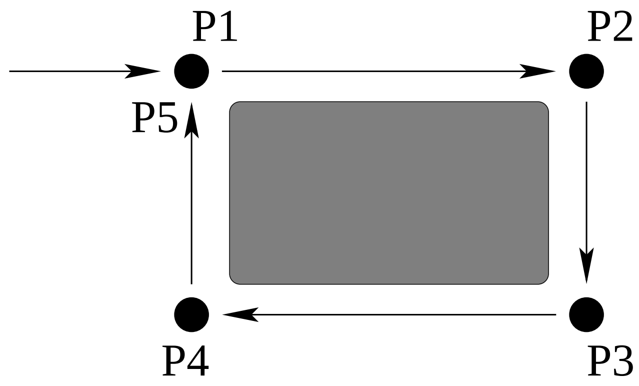


Fig. 3. The “exploring around the block” problem. The problem of recognising the same place under different state labels [14].

radial lines. The azimuth of a radial line in the image corresponds to the azimuth of the vertical edge in the scene, as viewed from the optical axis of the camera. In a man-made environment, the vertical edges in the environment provide an optimal feature to divide the environment into topologically different places. Some examples of vertical edges in a building are: door-posts, corners between two walls, the lateral edges of furniture, etc. In the SSH framework, vertical edges can be used to generate a *distinctiveness measure* to identify transition of state in the robot ontology.

Another advantage of this omnidirectional vision system is its rotational invariance. If the robot rotates a certain angle about the optical axis of the camera, the relative position of the objects in the image does not change. The image is only rotated and the objects appear to have experienced an azimuthal shift equal to the angle of rotation. This permits a straightforward solution to the problem of *exploring around the block* [14], i.e. of recognising the same place under different state labels, see Fig. 3. Here the robot is moving around the block following the arrows. When the robot reaches *Place 5* from *Place 4*, it is very difficult to recognize *Place 5* as the previously visited *Place 1* when using a perspective camera or a frontal sonar array. This is because the robot experiences very different sensory input in the same place coming from different directions. On the other hand, if the robot is equipped with an omnidirectional camera and it makes use of the rotational invariance the sensory experience is the same. In other words, using the SSH terminology, it is easy to spot whether the current **view** is the same it has experienced before and therefore to consider this view, not as a different **place**, but as the same place reached from a different direction.

Another problem which is easily solved by omnidirectional vision is to discriminate the type of movement the robot is performing at a given time. Using optical flow techniques, Svoboda showed that with an omnidirectional vision system it is very easy to discriminate between a small rotational movement and a small translational movement [31]. This task is very difficult for a vision system fitted with a perspective camera. Moreover, using active vision on an omnidirectional vision system it is possible to estimate precisely the motion of a robot. See again [31] for a literature review.

4 SSH Implementation

In this section we will present our implementation of the SSH proposed by B. Kuipers. We will refer to the terminology introduced in Section 2. As we will see, our implementation is a little different from the original one proposed by Kuipers. This is because of the use of very different sensors in the two implementations. One example is the fact that we joined the *Causal Level* and the *Control Level* in a single level that better suited our omnidirectional vision system. In fact, in our implementation, the Control Level and the Causal Level both obtain data from the same source. To convey a clear understanding of the details of the implementation we will start by describing the robot used in the experiments.

4.1 Robot description

The robot used in this implementation is depicted in Fig. 1. This is a robot originally built to serve as goalkeeper in the RoboCup competitions. The robot has two driven wheels and two spherical wheels (for balance). The robot can rotate on the spot around the optical axis of the camera. The omnidirectional camera is composed of a standard perspective camera (a SONY XC-999) and a convex mirror with a specially designed profile. The design of this mirror was inspired by the work of Marchese and Sorrenti [22]. The shape of the mirror is designed in order to maximize the image resolution in the regions of interest [23]. This new shape exploits all information it is possible to gather from the environment with respect to the mirror we used in previous work [26], in addition of being smaller and lighter than the old one.

The area close to the center of the image is strongly deformed, because of the derivative's discontinuity at the vertex of the mirror, see Fig. 4. The body of the robot appears distorted in the shape of a black cross (the cross's arms correspond to the corners of the robot body). For our application, this is not a disadvantage, because, even if the central part and the periphery of the image (the regions marked with small diamonds in Fig. 4) are not be used for measurement purposes due to the strong distortion, they can be used to discriminate between vertical edges and accidentally apparent radial edges.

The omnidirectional camera is calibrated. Thus, a mapping function is known, which associates the coordinates of every single pixel in the image to a 3D ray in space and, thus, to the coordinate of a corresponding point in the world (assum-

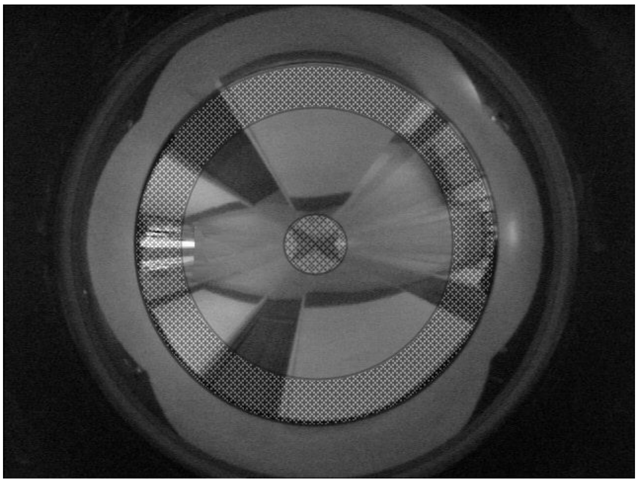


Fig. 4. The region of the image used to measure the position of the objects. The regions marked with small diamonds are not used in measuring distances due to high distortion.

ing a flat ground plane or other known world geometry). The calibration procedure also associates every pixel of the image with the average error encountered when estimating the position of an object in the world. When the position of an object is measured, the corresponding measurement error is associated with this measurement. In addition, a likelihood index is associated to every measure. This index is incremented if the same object is also detected in the following frames. Conversely it will be decremented.

An overview of our implementation of the SSH is given in Fig. 5. The figure shows the classical SSH of Fig.. 2 overlayed with the software modules which encode them. In the following, our implementation of each level is presented in detail.

4.2 The implemented Sensory Level

As stated in Sec. 2, the sensory level extracts the useful environmental features from the continuous flow of information of the robot's sensors. Our only sensor is an omnidirectional camera and the information flow is a sequence of omnidirectional images. As environmental features we used the vertical edges present in the environment. Several authors selected features that strictly speaking are not present in the environment but only in the pictures of the environment, like brightness pattern or other features only loosely related to the objects in the world. Usually, these features are extracted from the images with the use of heavy mathematical tools [7] [13] [32]. We decided not to follow this approach, but to select features that are strictly bound to the objects in the real world, these features are the vertical edges existing in the environment. When the robot moves, the edges appear to move in the image. Analysing this motion, it is possible to extract information both on the topology of the environment and on the robot's movements.

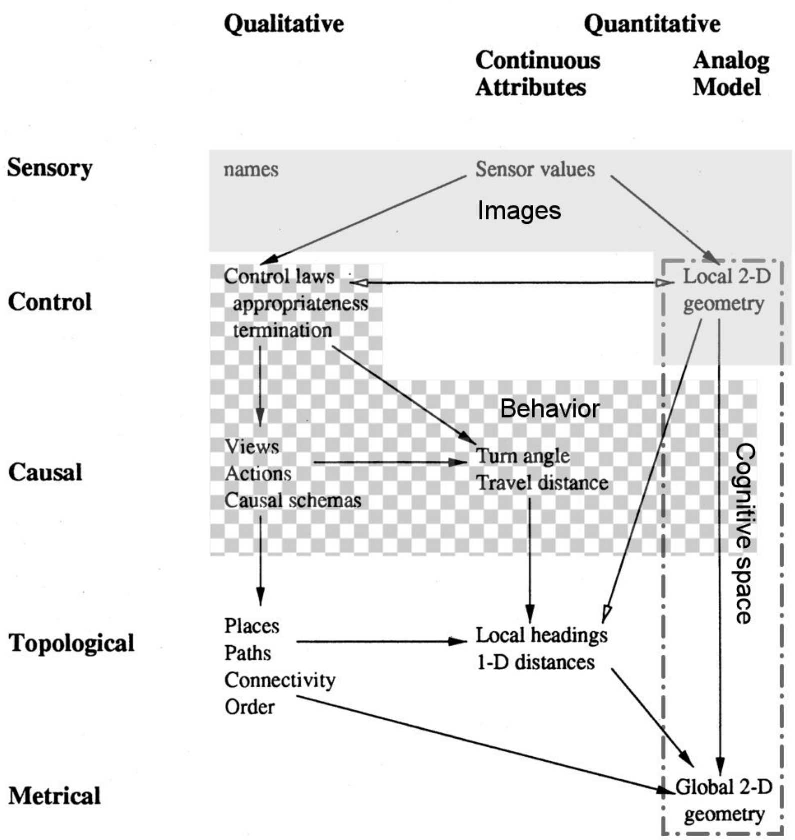
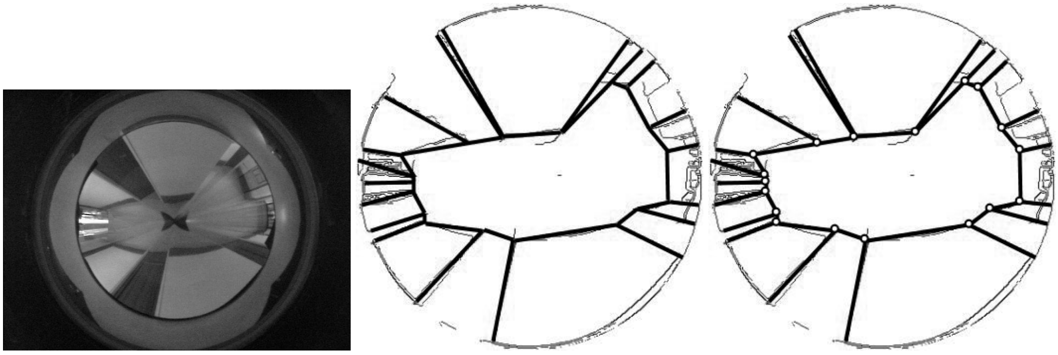


Fig. 5. Our implemenation of the SSH: The boxes with different fill-in represent the implemented software modules and are overlaid on top of the graphical representation of the SSH presented in Fig. 2 (adapted from B. Kuipers).

The processing performed by the Sensory Level can basically be summarized in few steps. The robot takes a snapshot at a certain location, Fig. 6a. First, it performs an edge detection to extract the edges from the picture, generating a binary image, Fig. 6b. Second, the black and white image, containing only the detected edges, is processed with a Hough transform to identify the radial lines, Fig. 6c, where the the end-point of the detected radial lines is marked with a dot. The end-point of the edge is assumed to lay on the floor and it is used to calculated the distance of the edge from the robot.

In order to reliably detect vertical edges in complex images we created a novel implementation of the Canny edge detector [4]. The main differences with respect to the original work of Canny are: (i) a different way of calculating the gradient in color images and (ii) a specification of the filter for radial edges. The gradient of color images is calculated by modifying the technique developed by Wesolkowski [29], which exploits the information of the three color channels of color images and returns a scalar gradient value. The original formulation was modified in order to improve its performance with regards to both computational performance and pixel accuracy. In particular, the `abs()` operator was preferred to euclidean distance since it's faster and the weight matrix can have an arbitrary dimension that depends



(a) (b) (c)

Fig. 6. The image processing sequence on an example image. (a) The original omnidirectional image. (b) The result of the edge detector. (c) The vertical edges and their support point.

1	1	2	1	1
1	2	3	2	1
2	3	0	3	2
1	2	3	2	1
1	1	2	1	1

Table 1
The kernel used in the edge-detector filter.

on the number of pixels around the one currently processed. The best results were obtained with the kernel values reported in Table 1. In order to specialize the filter for radial edges in the image, the classical *non-maxima suppression* phase of the Canny's filter has been modified putting a bias on following pixels along a radial direction. This was implemented with dynamic programming, as suggested in the paragraph "Edge following as dynamic programming" in [8]. The basic idea is to start a hysteresis cycle when a pixel on an edge is found and pixels lying on radial lines are preferred for edge continuation. The threshold and the radial tolerance are dynamically redefined during the process. Even if this approach is computationally intensive, it offers high quality and flexibility. We compensated the time required by the more complex elaboration, by eliminating the promote/demote cycles of the pixels. The resultant processing is only a few milliseconds slower than the original Canny algorithm, while providing better results.

To select the radial edges among the edges identified by the edge detector, a Hough transform was used. The geometry of the image enables a major simplification of the Hough algorithm [9]. If the edge pixels are projected from the Cartesian coordinate system into a polar coordinate system with the origin in the centre of the image, a radial line can be described as a set of pixels with the same angular coordinate and with varying radial coordinate. By looking at the histogram of the pixels' angular values, we can spot the radial lines as those where the histogram count is over a certain threshold. The threshold corresponds to the minimal length (in pixels) of the radial lines that we consider as vertical edges. The choice of the threshold for the minimal length of a vertical edge is a critical parameter. A low threshold can also detect pixels that accidentally have the same angular coordinate, but do not belong to the same line. On the other hand, a high threshold misses some vertical edges, especially when they are far away and they appear as small segments. The threshold has been set empirically to 60 pixels.

4.3 The implemented Control and Causal Level

When the robot moves in the environment, the vertical edges appear to move in the image sequence. It is possible to identify some "events" in the edge motion that are topologically meaningful. These events happen at single points or lines in the space, therefore they can be used to identify distinct points or boundaries in that space. This is the key idea that permits us to extract from the continuous world a set of distinct places as required by the Causal level of the SSH. Based on these events, we also created some distinctiveness measures used to trigger appropriate control laws, as required by the Control level. We have two control laws: translations and rotations. The events we identify in the edge motion during a translation of the robot are:

- A new edge exits from occlusion;
- An edge disappears, because occluded by another object;
- The two vertical edges are 180 deg. apart in the image;
- Two pairs of vertical edges are 180 deg. apart in the image;

The third event is particularly related to the natural topology of the environment, because it happens when the robot is passing through a door or is exiting/entering a corridor, see Fig. 7. Every time one of these events is detected, the robot create a new distinct place and stores the local view relative to this position, i.e. the omnidirectional image see Fig. 6a.

When the robot performs a rotation (i.e. it turns on the spot), the distance of the robot from the objects does not change, so the objects do not change their shape in the image, see Fig. 8. The events we identify in the edge motion are:

- The vertical edges of the scene appear as radial lines that move only by changing their azimuth;
- All the edges experience the same azimuthal shift;
- The number of visible edges is constant: no edges appear or disappear;

The last consideration comes from the fact that there is no relative displacement between the robot and the objects. Therefore, the occlusions do not change. In other words, the image does not change, it appears only rotated around its centre. The

Fig. 7. The sequence of the robot passing through a door. In the first frame, the posts of the door are not yet at 180 deg. In the second frame, they are at 180 deg. In the third, they are not longer at 180 deg.

Fig. 8. The sequence of images acquired by the omnidirectional camera of the robot while it turns on the spot

topological consideration we can draw from the rotation sequence is that nothing changes. Therefore, in the jargon of the SSH, all the *views* that differ only for a rotation around the centre of the image have to be associated to the same *place*.

In the original SSH formulation, the causal level abstracts a discrete model of the environment from the continuous world. Our discrete model of the environment is composed by storing a local view, i.e. an omnidirectional image, for each distinct point. The omnidirectional images are linked by the actions (i.e. sequences of control laws (translations and rotations) to be applied to reach the next location from the current one. Using the stored omnidirectional images, one could also implement *hill-climbing* control laws proposed by Kuipers at the Control Level [14]. The hill-climbing strategy can be based on visual homing algorithms for omnidirectional images such as the ones proposed by Franz et. al [6] or by Argyros et al. [2] (or on adapted homing algorithms for standard cameras such as the one proposed by Rizzi et al. [1]).

In the environment exploration phase, the robot selects a sequence of control laws to move in the environment, using the map generated at the Metrical Level (see Sec. 4.5), in order to determine the free space. When a new action is initiated, the robot aims at the mid-point of the line connecting the two closest vertical edges not

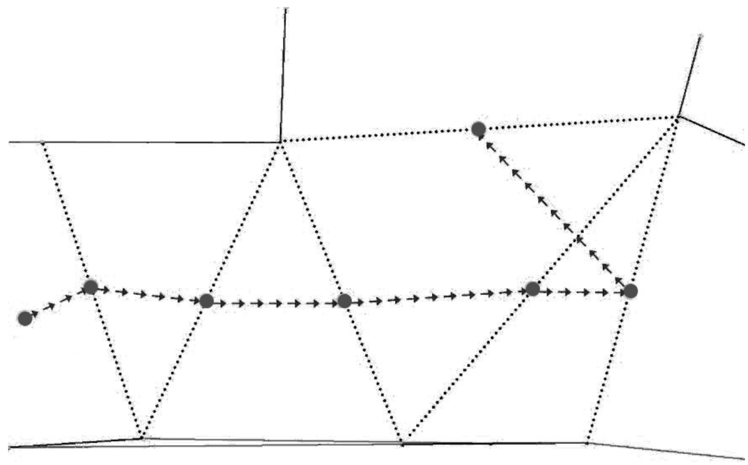


Fig. 9. An example of path generated by the robot in the exploration of the environment.

already explored. If the line between two vertical edges is an obstacle, the robot marks it as a wall. If not, it marks it as free-space and it flags it as already visited. A sequence of exploration actions is depicted in Fig. 9. The lines connecting two vertical edges are marked with the dashed line if they are to be explored, while if they represent a wall they are marked with a solid line . The mid-points of the lines connecting two vertical edges are marked with the circle. The robot path is marked with the arrows.

4.4 The implemented Topological Level

In our implementation the topological map built at the Topological Level does not differ a lot from the map built at the Control and Causal Level. The main difference is in the representation of the nodes of the map, i.e. the distinct points. At the Control and Causal Level they were represented by the omnidirectional images grabbed at those locations, while at the Topological level they are represented by local maps extracted from these images such as the one depicted on the right of Fig. 10. As written by Lee [19]:

> *"The places and paths in the topological map are more complex structures than the nodes and arcs of a mathematical graph since they can be annotated with geometrical properties."*

The local map is created in real-time by exploiting a look-up table created when the omnidirectional camera is calibrated. The look-up table provides the world coordinates (at the floor level) of every pixel in the image.

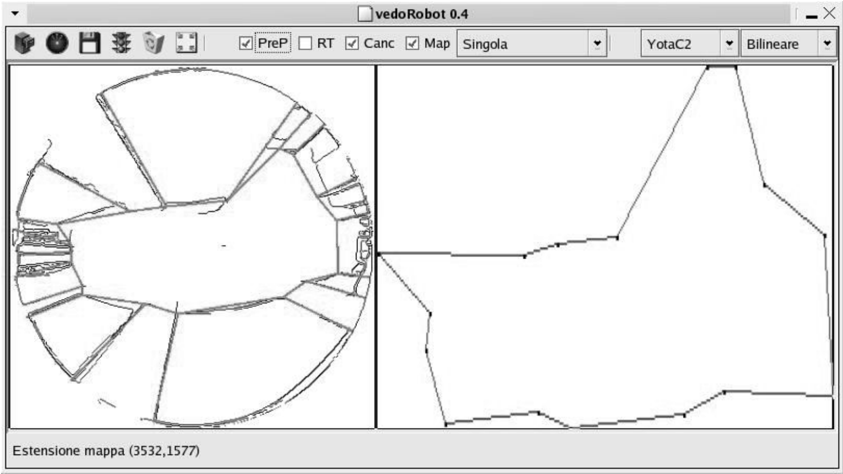


Fig. 10. The construction of a local metrical map (right) from an omnidirectional image (left).

4.5 The implemented Metrical Level

At the Metrical Level the topological map is augmented with metrical information on the relations and distances among the nodes. In our implementation this is done by estimating the robot motion and inferring the robot location at each motion step. This is done by matching the visible edges in the image with the edges of the local map. The matching is performed purely on the geometric information exploiting the fact that only two kinds of motion are possible: translation or rotation. First, a rotation is assumed and a matching is tried. This is done by finding the rotation angle that provides the overlap with the minimum error of all visible vertical edges in the image with all edges in the map. As we will see in the experimental section, this process is very robust because of the small error associated with the measurement of the azimuth of the vertical edges and the one-to-one matching. If the matching is not successful, a translation is assumed. A new match is tried by finding the translation vector that minimizes the mean squared error in the overlap between the support point of the vertical edges in the image and the edges position in the map. In this case the matching is a little more complex, because there is not not a one-to-one relationship between the vertical edges in the image and the edges in the map. In the case of translation, new edges can appear in the image and edges can disappear because of occlusion. The matching is complicated also by the fact that the measurement of the distance of the support point of the vertical edge from the robot is affected by an error that depends on the distance between the edge and the robot.

As already stated in Sec 4.1, the measurement of the location of a vertical edge in the environment is associated with a corresponding measurement and a likelihood index. Once an edge is detected several times in the frame sequence, its likelihood index increases and its true position is calculated as the average of the measured

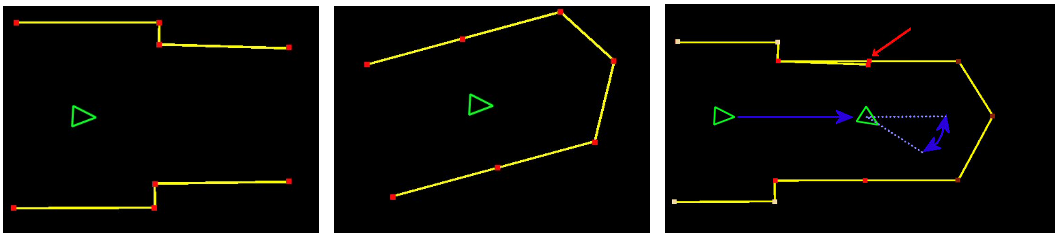


Fig. 11. A sketch representing the fusion of two local maps (left and middle) into a single global map (right).

positions weighted by the errors associated with these measurements.

At the Metrical Level, the local map generated at the Topological Level is enhanced by distinguishing the space lying between two vertical edges as either being a wall or free-space. This is done mainly by evaluation of the reciprocal relationships between vertical edges and by exploiting the information given by the motion of the robot. The rules regarding the vertical edges are:

(1) a new wall cannot be created between two vertical edges if there is already a third vertical edge in the middle;
(2) a new wall cannot cross another wall;
(3) a new wall cannot occlude a visible vertical edge;
(4) walls not connected to any vertical edge, and edges not connected to a wall, are eliminated from the map

The rules regarding the robot motion are:

(1) a wall cannot be created if it intersects the previous path of the robot
(2) a wall cannot create if it intersects the robot's body

The local maps generated at the Topological Level are merged into an incremental global metrical map exploiting the information on the position of the vertical edges, on the position of the walls and on the estimated motion of the robot. An example is shown in Fig. 11, in which two simplified local maps are fused into a single global map, by solving small misalignment as the one indicated by the arrow on the top of the image. Rotation and translation movements, estimated using the techniques previously explained, are annotated in the global map.

5 Experimental Results

To test our implementation of the SSH, we performed some preliminary experiments in the corridors of our department. The final experiment was performed in the corridor depicted in Fig. 12. This corridor has white walls and wooden doors and no artificial landmarks. The robot is the black box in the middle of the corridor

Fig. 12. A picture of the environment in which the experiments have been performed.

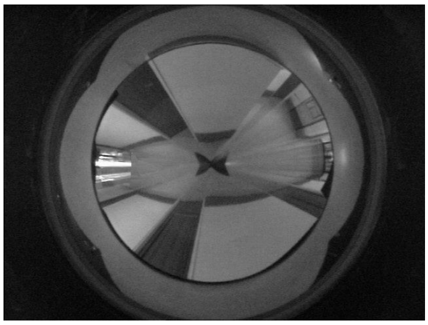


Fig. 13. The omnidirectional image taken by the robot in that position.

and an omnidirectional image grabbed by the robot in that position is depicted in Fig. 13. Fig. 14 shows a sub-set of the image sequence grabbed and processed by the robot, while it moves in the environment. In the first column are the original omnidirectional images, in the second column are the results of the edge detection and Hough transform to find the vertical edges and in the third column is the map created by the Metrical level of our implementation of the SSH. The final map created by our Metrical level is compared with the ground-truth plan of the test environment in Fig. 15. As one can see, our implementation is able to retrieve the gross structure of the explored environment to an accuracy acceptable for topological mapping and navigation.

Most of the inaccuracy in the resulting map comes from the edge matching process during translations. The estimation of the positions of vertex of the vertical edges are affected by much higher noise than the estimated azimuth of vertical edges. We measured an experimental average error of 15 centimeters in the position of vertex of the vertical edges resulting in a matching rate ranging from 60% to 80%. While in the case of rotation we have obtained an average error of less than 1 degree and the mean match rate is of 90% on roto-translations and 95% on pure rotations. While these results leave room for improvement, they indicate the system is working correctly. In addition, it should be considered they were obtained by estimating the rotations and especially the translations from vision only, without using the encoder data. It should be noted that the approach combines topological and metric levels of mapping and navigation, so such levels of error are acceptable for these purposes.

6 Conclusions and future work

This work implemented the Spatial Semantic Hierarchy on a real robot. The main contributions of this work are: (i) the realization of the sensory level with an om-

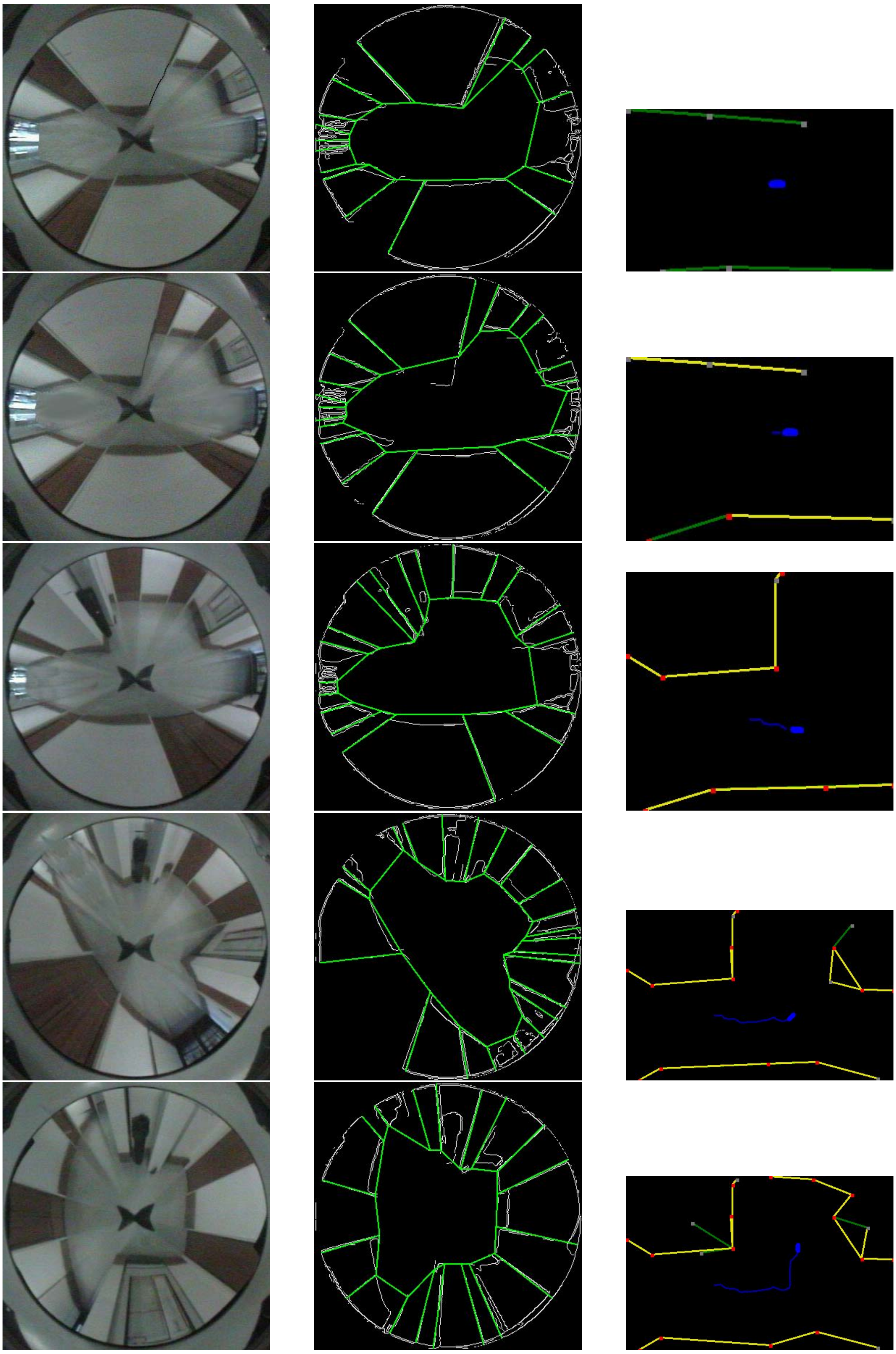


Fig. 14. A sub-set of the image sequence grabbed and processed by the robot. (From left to right) the original omnidirectional image, the edge detection and the Hough transform, the map created at the Metrical Level.

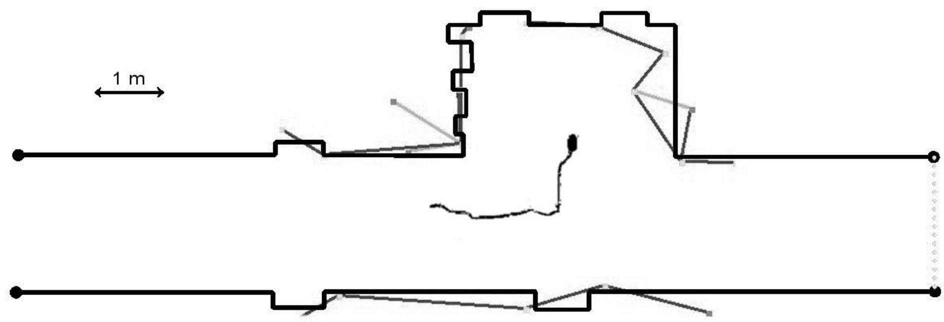


Fig. 15. The resulting map compared with the ground-truth of the explored environment.

nidirectional vision system, (ii) we showed that an omnidirectional vision system is a good sensor for the SSH, (iii) we pointed out which of the features present in a omnidirectional image can be used to detect the *transitions of state* needed by the control level of the SSH and we showed the existence of a strict link between the *views* of the SSH and the image taken by an omnidirectional sensor. In Sec 5, we presented some preliminary experiments that showed the feasibility of the approach using only the omnidirectional camera as the unique sensor and revealed some limitations of the implementation.

This work was formulated over a period of time while other approaches, most notably SLAM (Simultaneous Landmark and Mapping), were developing. SLAM identifies landmarks and builds a map in order to reduce and put bounds on errors in the mapping process. SLAM and SSH can be seen to share certain philisophical similarities. Inthe next works, we will consider what crossover there may be between the two approaches. In particular we will look at the topological aspects of our approach. Some works on SLAM already includes topological elements.

In the future, in addition to improving the robustness of the implementation by improving the matching algorithm and by exploiting the data collected by the wheel encoders, we will work to extend our approach to multi-robot systems. The basic idea is that every robot builds the local map of the portion of environment it visited. When two robots meet, they share their portions of the map, fusing them into a global map. Some preliminary ideas and experiments were presented in [24] [25].

We will seek to relax the contraints of motion to rotation or translation only. It is well known in the computer vision literature that rotational and translational components of planar pin hole cameras can be decomposed into rotational and translational components. We will seek to extend this to our sensor geometry.

We will seek to extend our approach to include new and very powerful and robust landmarking techniques such as SIFT (Scale-Invariant Feature Transform). Any such feature may be incorporated by tracking occlusions and other topological transitions of such features using new visual odometry techniques.

References

[1] R. C. A. Rizzi, G. Bianco. A bee-inspired visual homing using color images. *Robotics and Autonomous Systems*, 25(3):pp. 159–164, 1998.

[2] A. Argyros, C. Bekris, S. Orphanoudakis, and L. Kavraki. Robot homing by exploiting panoramic vision. *Journal of Autonomous Robots*, vol. 19(no. 1):pp. 7–25, July 2005.

[3] G. Bianco and A. Zelinsky. Biologically-inspired visual landmark learning and navigation for mobile robots. *Proc. of the 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1999.

[4] J. Canny. A computational approach to edge detection. In *Transactions on Pattern Analysis and Machine Intelligence*, volume 8, pages 679–697. IEEE, 1986.

[5] A. Davison. Real-time simultaneous localisation and mapping with a single camera. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410, 2003.

[6] B. S. Franz, M. and H. H. Blthoff. Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79:191–202, 1998.

[7] M. O. Franz, B. Schlkopf, H. A. Mallot, and H. H. Blthoff. Learning view graph for robot navigation. *Autonomous Robots*, 5:pp. 111–125, 1998.

[8] G.X.Ritter and J.N.Wilson:. *Handbook of computer vision image algebra*. CRC Press, 1996.

[9] P. Hough. Method for recognizing complex patterns. Technical report, US Patent 3069654, 1962.

[10] H. Ishiguro. Development of low-cost compact omnidirectional vision sensors. In R. Benosman and S. Kang, editors, *Panoramic Vision*, chapter 3, pages pp. 23–38. Springer, 2001.

[11] J. Kim and M. Chung. Slam with omni-directional stereo vision sensor. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 1, 2003.

[12] D. Kortenkamp, P. Bonasso, and R. Murphy, editors. *Artificial Intelligence and Mobile Robotics*. AAAI Press/ MIT Press, 1998.

[13] B. Krse, N. Vlassis, R. Bunschoten, and Y. Motomura. Feature selection for appareance-based robot localization. *Proc. 2000 RWC Symposium*, 2000.

[14] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:pp. 191–233, February 2000.

[15] B. Kuipers. An intellectual history of the spatial semantic hierarchy. In M. Jefferies and A. W.-K. Yeap, editors, *Robot and Cognitive Approaches to Spatial Mapping*, page (To appear). Springer Verlag, 2006.

[16] B. J. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:pp. 47–63, 1991.

[17] B. J. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:pp. 47–63, 1991.

[18] B. J. Kuipers and T. Levitt. Navigation and mapping in large scale space. *AI Magazine*, 9(2):pp. 25–43, 1988. Reprinted in Advances in Spatial Reasoning, Volume 2, Sushing Chen (Ed.), Norwood NJ: Ablex Publishing, 1990, pages 207–251.

[19] W. Y. Lee, Ph.D. *Spatial Semantic Hierarchy for a Physical Mobile Robot*. PhD thesis, The University of Texas at Austin, 1, 1996.

[20] T. Lemaire and S. Lacroix. SLAM with Panoramic Vision. Technical report, Technical report, LAAS-CNRS, 2006. submitted to the Journal for Fields Robotic in the special issue SLAM in the Fields. Available from: http://www. laas. fr/tlemaire/publications/lemaireJFR2006_SLAM. pdf.

[21] T. S. Levitt and D. T. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence Journal*, 44(3):pp. 305–361, 1990.

[22] F. Marchese and D. G. Sorrenti. Omni-directional vision with a multi-part mirror. In P. Stone, T. Balch, and G. Kraetzschmar, editors, *RoboCup 2000: Robot Soccer World Cup IV*, LNCS. Springer, 2001.

[23] E. Menegatti, F. Nori, E. Pagello, C. Pellizzari, and D. Spagnoli. Designing an omnidirectional vision system for a goalkeeper robot. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: Robot Soccer World Cup V.*, L. N. on A. I, pages pp. 78–87. Springer, 2002.

[24] E. Menegatti and E. Pagello. Omnidirectional distributed vision for multi-robot mapping. In *Proc. of the 6th International Symposium on Distributed Autonomous Robotic Systems (DARS02)*, pages pp. 279–288, Fukuoka, Japan, June 2002.

[25] E. Menegatti and E. Pagello. Toward a topological mapping with a multi-robot team. In *Proc. of the Workshop on Cooperative Robotics, A. Saffiotti Organizer IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS02-WS7)*, pages pp.V/1–V/7, Lausanne, October 2002.

[26] E. Menegatti, M. Wright, and E. Pagello. A new omnidirectional vision sensor for the spatial semantic hierarchy. In *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM '01)*, pages pp. 93–98, July 2001.

[27] K. Pierce and B. Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:pp. 169–227, 1997.

[28] A. Rizzi and R. Cassinis. A robot self-localization system based on omnidirectional color images. *Robotics and Autonomous Systems, Elsevier*, 34(1):pp. 23–38, 2001.

[29] S.B.Wesolkowski. Color image edge detection and segmentation, a comparison of the vector angle and the euclidean distance color similarity. Master's thesis, Systems Design Engineering, Faculty of Engineering, University of Waterloo, Canada, 1999.

[30] S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21(3):364–375, 2005.

[31] T. Svoboda, T. Pajdla, and V. Hlavac. Motion estimation using central panoramic cameras. *IEEE Conf. on Intelligent Vehicles, Stuttgart, Germany*, October 1998.

[32] N. Winters and J. Santos-Victor. Mobile robot navigation using omni-directional vision. *3rd Irish Machine Vision and Image Processing Conf. IMVIP99, Dublin, Ireland*, September 1999.

[33] Y. Yagi, Y. Nishizawa, and M. Yachida. Map-based navigation for a mobile robot with omnidirectional image sensor copis. *IEEE Transaction on Robotics and automation*, VOL. 11(NO. 5):pp. 634–648, October 1995.