Auton Robot (2006) 21:15–28
DOI 10.1007/s10514-005-6066-z

Information based indoor environment robotic exploration and modeling using 2-D images and graphs

Vivek A. Sujan · Marco A. Meggiolaro · Felipe A. W. Belo

Published online: 22 April 2006
© Springer Science + Business Media, LLC 2006

Abstract As the autonomy of personal service robotic systems increases so has their need to interact with their environment. The most basic interaction a robotic agent may have with its environment is to sense and navigate through it. For many applications it is not usually practical to provide robots in advance with valid geometric models of their environment. The robot will need to create these models by moving around and sensing the environment, while minimizing the complexity of the required sensing hardware. Here, an information-based iterative algorithm is proposed to plan the robot's visual exploration strategy, enabling it to most efficiently build a graph model of its environment. The algorithm is based on determining the information present in sub-regions of a 2-D panoramic image of the environment from the robot's current location using a single camera fixed on the mobile robot. Using a metric based on Shannon's information theory, the algorithm determines potential locations of nodes from which to further image the environment. Using a feature tracking process, the algorithm helps navigate the robot to each new node, where the imaging process is repeated. A Mellin transform and tracking process is used to guide the robot back to a previous node. This imaging, evaluation, branching and retracing its steps continues until the robot has mapped the environment to a pre-specified level of detail. The set of nodes and the images taken at each node are combined into a graph to model the environment. By tracing its path from node to node, a service robot can navigate around its environment. This method is particularly well suited for flat-floored environments. Experimental results show the effectiveness of this algorithm.

Keywords Mobile robots · Localization · Map building · SLAM · Information theory

V. A. Sujan
Advanced Controls Division, Cummins Engine Company,
Columbus, IN 47201
e-mail: vivek.a.sujan@cummins.com

M. A. Meggiolaro
Department of Mechanical Engineering, Pontifical Catholic University of Rio de Janeiro,
Rio de Janeiro 22453-900, RJ-Brazil
e-mail: meggi@alum.mit.edu

F. A. W. Belo
Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro,
Rio de Janeiro 22453-900, RJ-Brazil
e-mail: felipe-belo@uol.com.br

1. Introduction

In recent years, mobile service robots have been introduced into various non-industrial application areas such as entertainment, building services, and hospitals. They are relieving humans of tedious work with the prospect of 24-hour availability, fast task execution, and cost-effectiveness. The market for medical robots, underwater robots, surveillance robots, demolition robots, cleaning robots and many other types of robots for carrying out a multitude of services has grown significantly (Thrun, 2003). The sales of mobile robots are projected to exceed the sales of factory floor robots by a factor of four, exceeding US$2 billion within this decade (Lavery, 1996). And unlike the factory floor robot market, the sources for the vast majority of these machines could be U.S. companies.

Service robots for personal and private use are mainly found in the areas of domestic (household) robots, which include vacuum cleaning and lawn-mowing robots, and entertainment robots, including toy and hobby robots. If the technology for personal service robots provides what it has promised, at a competitive price, and if there is a sufficient degree of consumer acceptance, then this can be indeed a very large market.

Due to increased computational performance, algorithm complexity has grown thus providing increased system capability (Borenstein and Koren, 1990; Khatib, 1999; Lawitzky, 2000; Nister, 2003; Wong et al., 2000). This growth in algorithm complexity has been in conjunction with growth in hardware complexity. However, the high costs associated with hardware complexity are a discouraging factor. This economic drive has been seen in the last decade, where the performance of industrial and personal robots has radically increased while prices have fallen. A robot sold in 2000 would have cost less than a fifth of what a robot with the same performance would have cost in 1990 (World Robotics, 2001). Although hardware costs have declined with respect to their sophistication, this economic trend will still require the replacement of complex hardware architectures by more intelligent and cost-effective systems.

In this work, an algorithm is developed to allow a mobile service robot to explore and build its environment model for future navigation requirements, using a limited sensor suite consisting of a single monocular camera system fixed to the mobile base, wheel encoders and contact switches. The main objective of this algorithm is to allow a low-cost robot to localize itself and navigate through a flat-floored static environment such as an office floor or apartment. The algorithm is based on determining the information present in sub-regions of a 2-D panoramic image of the environment from the robot's current location. Using a metric based on Shannon's information theory (Reza, 1994), the algorithm determines potential locations of nodes from which to further image the environment, traversing the graph in a depth-first manner. Using a feature tracking process, the algorithm helps navigate the robot to each new node, where the imaging process is repeated. When a node is sufficiently explored (i.e. no new exploration nodes are identified), then the algorithm uses a Mellin transform (Alata et al., 1998; Casasent and Psaltis, 1978; Ruanaidh and Pun, 1997) and tracking process to guide the robot back to a previous node. This imaging, evaluation, branching and retracing its steps continues until the robot has mapped the environment to a specified level of detail. The level of detail is application-dependent and specified before the exploration/mapping process is initiated. The set of nodes and the images taken at each node are combined into a graph to model the environment. This graph model is essentially the causal map described by Kuipers (2000), where the panoramic images correspond to views, navigation methods correspond to actions, and nodes correspond to distinctive states. Finally, by tracing its path from node to node, a service robot can then continue to navigate as long as there are no substantial changes to the environment (even though the proposed approach is relatively robust to such changes).

Environment mapping by mobile robots falls into the category of Simultaneous Localization and Mapping (SLAM). In SLAM a robot localizes itself as it maps the environment. Researchers have addressed this problem for well-structured (indoor) environments and have obtained important results (Anousaki and Kyriakopoulos, 1999; Castellanos et al., 1998; Kruse et al., 1996; Leonard and Durrant-Whyte, 1991; Thrun et al., 2000; Tomatis et al., 2001). These algorithms have been implemented for several different sensing methods, such as stereo camera vision systems (Castellanos et al., 1998; Se et al., 2002), laser range sensors (Tomatis et al., 2001), and ultrasonic sensors (Anousaki and Kyriakopoulos, 1999; Leonard and Durrant-Whyte, 1991). Sensor movement/placement is usually done sequentially (raster scan type approach), by following topological graphs, or using a variety of *greedy* algorithms that explore regions only on the extreme edges of the known environment. Geometric descriptions of the environment are modeled in several ways, including generalized cones, graph models and Voronoi diagrams, occupancy grid models, segment models, vertex models, and convex polygon models. Sensing uncertainties have been investigated for single or multi-robot systems (Roumeliotis and Rekleitis, 2004). However, the focus of most of these works is only accurate mapping. They do not address mapping efficiency. Researchers have addressed mapping efficiency to a limited amount (Kruse et al., 1996), but in these cases sensing and motion uncertainties are not accounted for. Prior work also assumes that sensor data provides 3-D (depth) information and/or other known environment clues from which this may be derived.

To achieve the localization function, landmarks and their relative motions are monitored with respect to the vision systems. Several localization schemes have been implemented, including topological methods such as generalized Voronoi graphs and global topological maps (Tomatis et al., 2001), extended Kalman filters (Anousaki and Kyriakopoulos, 1999; Leonard and Durrant-Whyte, 1991), and robust averages. Although novel natural landmark selection methods have been proposed (Simhon and Dudek, 1998), most SLAM architectures rely on identifying distinct, recognizable landmarks such as corners or edges in the environment (Taylor and Kriegman, 1998). This often limits the algorithms to well-structured indoor environments, with poor performance in textured environments.

On the other hand, the algorithm proposed in this work is able to localize the robot even in textured environments, without the need of 3-D information. In the following sections, an analytical development of the proposed algorithm is provided. The three primary components of the algorithm are developed and classified as: (i) potential child node identification, (ii) traverse to unexplored child node, and (iii) traverse to parent node. Experimental studies are conducted to demonstrate the validity of each primary component and the effectiveness of the entire algorithm.

2. Algorithm overview

The environment exploration and modeling algorithm proposed here consists of 3 primary components. The overall process is shown in Fig. 1. The mobile robotic agent models the environment as a collection of nodes on a graph. The first component of the algorithm is to identify potential child nodes from a given location, see Fig. 2. At each node the robot conducts a panoramic scan of the environment. This scan is done as a series of 2-D image snapshots using in-place rotations of the base by known angles. Next, an information theoretic metric is used to divide the panoramic image into regions of interest. If any region of interest contains sufficient information then it is identified as a potential child node, which would then need to be explored.

After the list of child nodes is collated, each child node is explored sequentially. To explore a child node, the mobile agent must traverse to that node. The second component of the algorithm is to traverse to a child node from the current node, see Fig. 2. This is achieved by tracking the target node using a simple region growing tracker and a visual servo controller. If the node cannot be reached by a straight line due to an obstruction, then the point of obstruction is defined as the new child node. At each child node the process of developing a panoramic image, identifying the next level of child nodes, and exploring them continues.

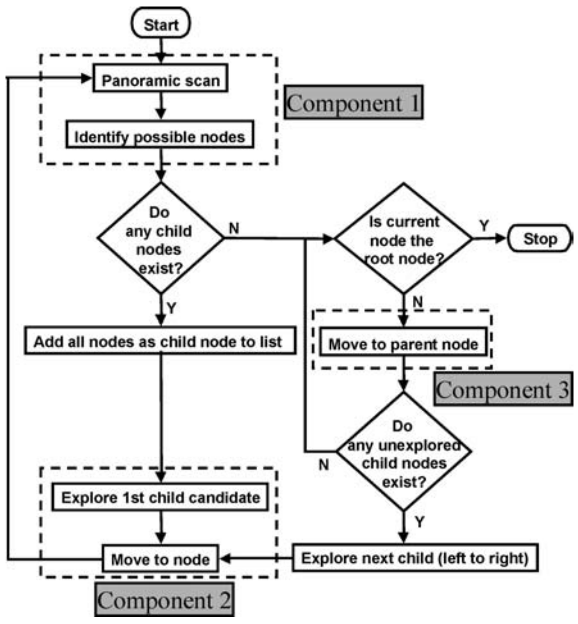

Start
Panoramic scan
Identify possible nodes
Component 1
Do any child nodes exist?
N
Y
Add all nodes as child node to list
Is current node the root node?
Y
Stop
N
Move to parent node
Component 3
Do any unexplored child nodes exist?
N
Y
Explore 1st child candidate
Move to node
Explore next child (left to right)
Component 2

Fig. 1 Algorithm overview

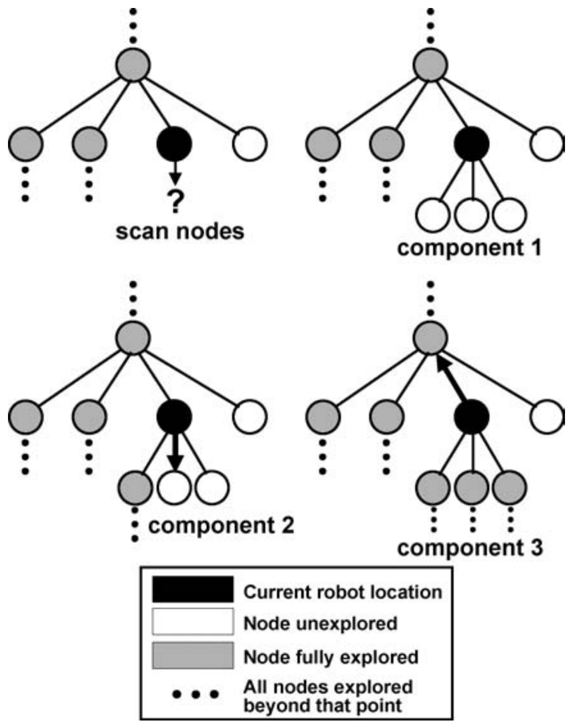


Fig. 2 Key components of the environment exploration and modeling algorithm

Once all the child nodes of the current node have been completely explored, the robotic agent traverses back to the parent node of the current one. The third component of the algorithm is to traverse to a parent node from the current node, see Fig. 2. This allows the other child nodes of the parent node to be explored. To move to a parent node requires a process quite different than the process of moving to a child node. To return to a parent node, the robot must first identify the direction of the parent node. This is done using a Mellin transform (invariant to scale) to determine if the image that the robot currently sees is what it would expect to see if it were pointed in the correct direction toward the parent node. The expected image is derived from the panoramic scan taken at the parent node. Once this direction is established, the robot moves toward the parent node. Visual servo control, based on the correlation between the current image and the image the robot would see if it were at the parent node pointed in the same direction, governs if the robot has reached the parent node.

Once all the nodes have been completely explored, the process of developing an environment model is complete. The relative locations of each node with respect to its parent node and child nodes may then be used to navigate from one node to another to guide the agent through its environment.

In the next sections, the details of the three key components of the algorithm are developed. Experimental results are presented along each section, obtained from a mobile robot agent adapted from an ER-1 commercial

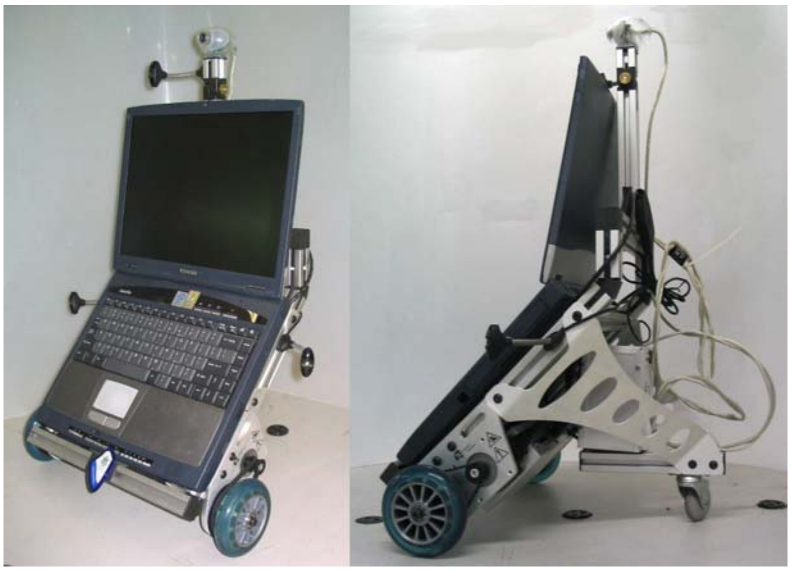


Fig. 3 Mobile robot experimental system

system (Evolution Robotics, 2005), see Fig. 3. The system consists of a 2-wheel differential system driven by step motors, equipped with wheel encoders (for odometry) and a single color camera set at a 176 × 144 pixel resolution mounted to its base. Three infrared sensors are used just as on-off bump switches for measuring contact, without providing any range data. The robot is controlled by a 1.5 GHz Pentium IV notebook mounted on its structure. The algorithm component 1 is described next.

3. Algorithm component 1: Potential child node identification

As described above, the first component of the algorithm is to identify potential child nodes from a given location. This process is initiated by developing a panoramic scan at the current node by the mobile robot agent. This scan is done as a series of 2-D image snapshots using in-place rotations of the base by known angles. The process involves first capturing a 2-D image. The elevation of the image is trimmed to maintain attention on lower regions that are accessible by the robot. A new image is captured by rotating the robot about its center by a fixed amount equal to the projection angle of the camera (or camera field of view). The process of capturing and trimming images continues until a full 360° panoramic scan is achieved.

It is assumed that rotations for this process can be achieved with sufficient accuracy so that the images may be directly fused without any further operations. This constraint may be readily relaxed with the addition of conventional image joining operations. For example, two images may be joined by identifying fiducials common to both to find a transformation between them. Figure 4 shows the result of a 360° panoramic scan generated by the ER-1 system using this approach.

To identify potential child nodes in the acquired panoramic image of the environment, it is necessary to reduce the image using a quadtree decomposition method. In a quadtree decomposition, an image is divided into four equal quadrants. Each quadrant is evaluated to determine if further division of that quadrant is required based on a metric. This process continues until no further decomposition is required. This quadtree decomposition of the image may be used in a manner suitable to the application. Here the decomposition metric is the quantity of information in the quadrant. If the information in the quadrant exceeds a predefined value, then further decomposition is warranted. The goal of determining the amount of information in the image is to identify regions where there are significant changes occurring in environment. These changes would indicate the presence of a corner, an edge, a doorway, an object, and other areas worth exploring.

However, changes may also be identified due to highly textured surfaces (such as furniture upholstery, carpets,

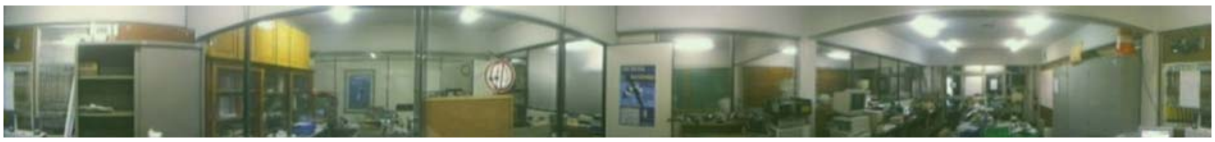


Fig. 4 Panoramic scan (360°) taken and joined by the experimental system

curtains, wallpaper, etc.) and noise. Clearly these are regions that need not be explored and mapped as they represent mostly uninteresting areas from the perspective of a service robot. Thus, before the quadtree decomposition process starts, the image is pre-processed to remove these high frequency effects. Pre-processing includes a two-step process. First, a low pass filter is applied to the image to remove the higher frequency effects. Second, an adaptive decimation process (Huntsberger et al., 2003; Sujan et al., 2003; Sujan and Meggiolaro, 2005) smoothes out patterns that exceed neighboring pattern variation. This is achieved by measuring the rate of change of pattern variance across the image. The algorithm limits the rate of change of pattern variance within its neighborhood, by comparing pixel intensity variance in windows with respect to similar windows located in the immediate neighborhood (nearest neighbors). When the rate of change of pixel intensity variance exceeds a prescribed rate, then the window is gradually passed through a low pass filter until the rate of change decreases to within bounds. Calibration is required and it is task dependent to determine the prescribed rate of change of pixel intensity variance as well as maximum window size. Once the image has been preprocessed, the information content may then be evaluated.

3.1. Measures of 2-D image information content

The information gained by observing a specific event among an ensemble of possible events may be described by the function (Reza, 1994)

$$H(q_1, q_2, \ldots, q_n) = -\sum_{k=1}^{n} q_k \log_2 q_k \tag{1}$$

where q_k represents the probability of occurrence for the kth event. This definition of information may also be interpreted as the minimum number of states (bits) needed to fully describe a piece of data. In 2-D signals, the gray level histogram of an ergodic image can be used to define a probability distribution:

$$q_i = f_i/N \quad \text{for} \quad i = 1 \ldots N_{\text{gray}} \tag{2}$$

where f_i is the number of pixels in the image with gray level i, N is the total number of pixels in the image, and N_{gray} is the number of possible gray levels. With this definition, the information of an image for which all the q_i are the same—corresponding to a uniform gray level distribution or maximum contrast—is a maximum. The less uniform the histogram, the lower the information. Although this is generally true, it is critical to note that images with ordered patterns may result in the same information content as one with no order. For example, a uniform histogram may be mapped to two very different images, such as a random arrangement of intensity values and a (uniform) smooth color gradient. Intuitively, the former would be expected to contain more information but, using Eqs. (1) and (2), they result in the same value.

However, this is readily rectified using conventional lossless image compression algorithms. Thus, before the information content of a data set can be evaluated, it must be processed by a compression algorithm. Only compression is needed here, since just a measure of the information present after compression is required, with no needs for decompression of the data. A few common methods of lossless compression are Simple Run-length compression, Lossless JPEG, Huffman coding, and Lempel-Ziv-Welch (LZW) compression. An ideal compression algorithm would remove all traces of any pattern in the data. Such an algorithm currently does not exist, however the LZW is well recognized to approach this limit. A thorough review is beyond the scope of this paper, but it can be found in (Smith, 1999). Limited studies on several of the above methods have been carried out and results presented in Sujan and Meggiolaro (2005).

3.2. Using information content to identify potential nodes

Potential nodes to map the environment may now be identified using the information theory developed above. The algorithm breaks down the compressed image into a *quadtree of high information regions*. The resulting image is divided into four quadrants. The information content of each quadrant is evaluated using Eqs. (1) and (2). This information content reflects the amount of variation of the image in that quadrant (where higher information content signifies higher variation in the image section). Quadrants with high information content are further divided into sub-quadrants and the evaluation process is continued. This is done by defining that a quadrant is divided if and only if the information content of any of its sub-quadrants is greater than its content.

Once the quadtree of the image is generated, a series of binary image operations are carried out to identify the nodes. First, starting with an image template of zero intensity, the edges of every quad in the quadtree are identified and marked on in the template image. Next, the dimensions of the smallest quad are ascertained. All pixels falling within quads possessing this minimum dimension are marked on (filled) in the template image. Next, an image opening operation is carried out on the template image to remove weak connectivity lines. The resulting image consists of distinct blobs. Region growing is used to define the centroids of each blob, which are identified as the potential nodal points. Finally, the image coordinates of these nodal points are transferred to the original panoramic scan to obtain the final nodal point.

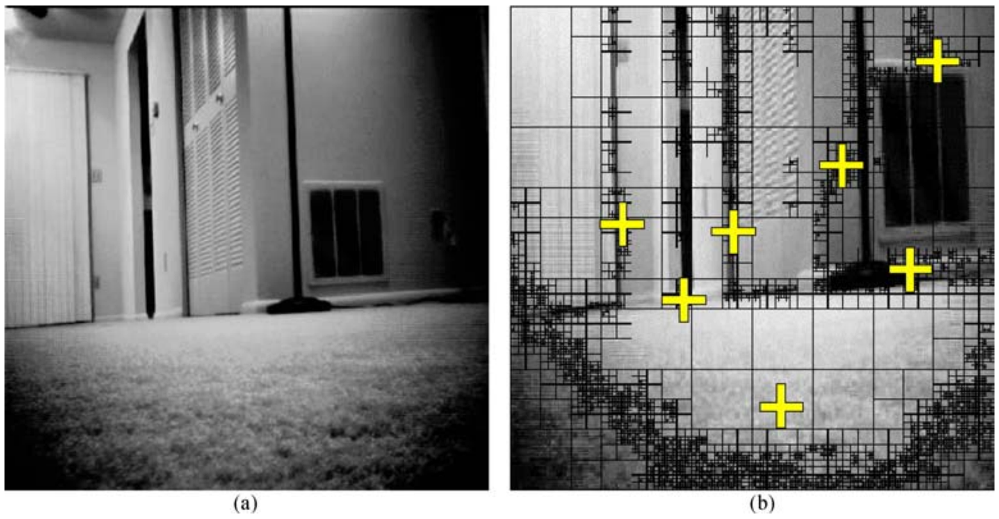


Fig. 5 (a) Raw image taken by mobile robot onboard camera and (b) processed image with trimmed elevation, showing quadtree decomposition and identified child nodes

Figure 5(a) shows an example of a raw image taken by the mobile agent during its exploration process. This image is trimmed and simplified using the information-based quadtree decomposition process described above. The results of the decomposition are shown in Fig. 5(b). It can be seen that several key areas in the image have been determined to have high quantities of information present. These areas are further processed to determine the coordinates of the child nodes. The identification process has selected nodes that are both useful (such as the ones near the doorway), but has also picked up nodes that may not be very useful (such as the one on the carpet). These latter nodes are often eliminated with greater low pass filtering in the image pre-processing steps.

The number of child nodes can be limited for each parent using a few heuristic rules—such as planar proximity—to speed up the mapping process. In addition, to avoid generating child nodes at points that have already been explored, a uniqueness confirmation is carried out for every new child node identified. Given the camera properties, the direction of the identified child node from the current node may be readily determined. Consequently, the direction to get back to the current node from the identified child node may be computed. The expected view from the child node in this direction may then be approximated. If any previously explored nodes have a similar view in the same direction (i.e., if both node images correlate within some user-defined tolerance) then the identified child node is not considered unique and thus may be eliminated. The algorithm component to traverse to the unexplored child node is discussed next.

4. Algorithm component 2: Traverse to unexplored child node

As described before, the second key component of the algorithm is to navigate the mobile robot to the child node from

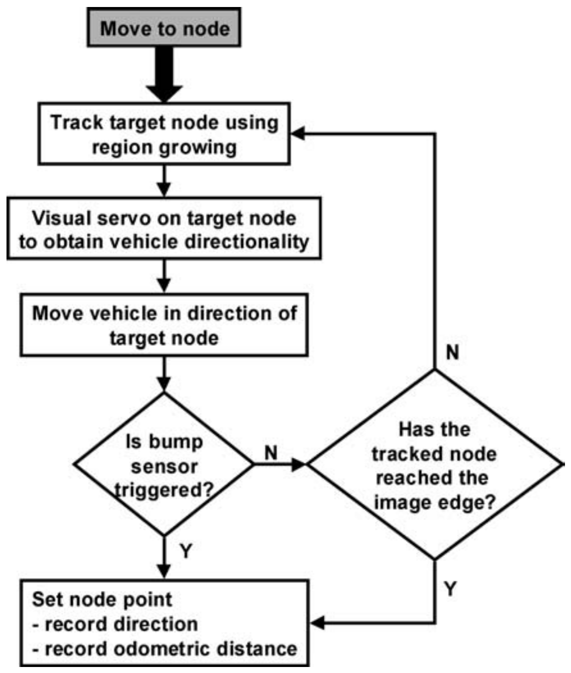

Move to node
Track target node using region growing
Visual servo on target node to obtain vehicle directionality
Move vehicle in direction of target node
Is bump sensor triggered?
N
Y
Has the tracked node reached the image edge?
N
Y
Set node point
- record direction
- record odometric distance

Fig. 6 Flow diagram to traverse to unexplored child node

the current node. This is achieved by tracking the target node using a simple region growing tracker and a visual servo controller. If the node cannot be reached by a straight line due to an obstruction, then the point of obstruction is defined as the new child node. The process is outlined in Fig. 6.

The visual servo controller, shown in Fig. 7, corrects for errors in heading of the mobile robot as it tracks the desired target. Target heading θ is determined based on the camera focal length and the position of the target node on the imaging plane.

The visual servo controller continues guiding the mobile robot until either the target node image vanishes across the

Fig. 7 Visual servo control on mobile robot heading

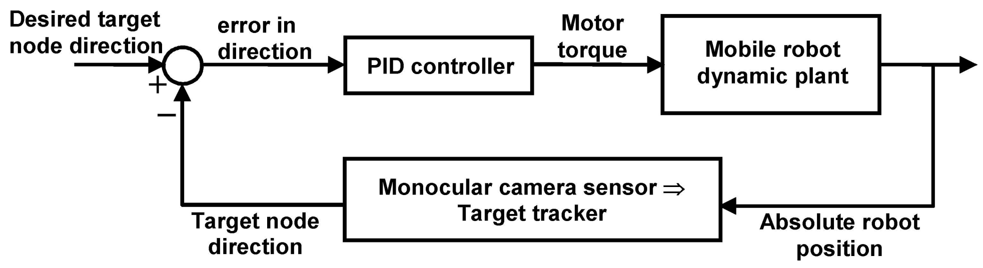

Desired target node direction
error in direction
+
−
PID controller
Motor torque
Mobile robot dynamic plant
Monocular camera sensor ⇒ Target tracker
Target node direction
Absolute robot position

image bottom (or top) edge, or its motions are obstructed. At this point the child node is established. If an obstruction has been identified, then the directionality of this obstruction is saved. When the child node list from Section 3 is compiled at this new node, then any nodes requiring motion in the direction of the obstruction are eliminated from the list.

It would be expected that tracking a single target point might not be as accurate as concurrently tracking multiple fiducials to determine the primary target heading. This increased redundancy results in improved heading accuracy, a feature that can be turned on or off depending on how accurately one needs to move. Let the measured/computed heading $\tilde{\theta}$ be related to the true heading $\hat{\theta}$ by a non-linear function, $h(\hat{\theta})$. The measurement vector is corrupted by a sensor noise v of known variance, R:

$$\tilde{\theta} = h(\hat{\theta}) + v \tag{3}$$

Assume that the measurement of the state vector $\hat{\theta}$ is done multiple times. In terms of the current measurement, a Jacobian matrix of the measurement relationship evaluated at the current state estimate is defined as:

$$H_k = \left.\frac{\partial h(\hat{\theta})}{\partial \hat{\theta}}\right|_{\hat{\theta}=\hat{\theta}_k} \tag{4}$$

The state (or computed heading) may then be estimated as follows:

$$\begin{aligned} K_k &= P_k H_k^T \left[H_k P_k H_k^T + R_k\right]^{-1} \\ \bar{\theta}_{k+1} &= \bar{\theta}_k + K_k \left[\tilde{\theta}_k - h(\hat{\theta}_k)\right] \\ P_{k+1} &= \left[1 - K_k H_k\right] P_k \end{aligned} \tag{5}$$

where P_k is the uncertainty associated with the heading after the kth fiducial is accounted for, and K_k is the gain associated with the kth fiducial. This estimate is known as the Extended Kalman Filter (Gelb, 1974). To maintain a measure of absolute uncertainty in heading of the robot at the current node with respect to the root node of the exploration tree, the uncertainty associated with each step in getting to the current node needs to be combined. This is achieved using a recursive method to determine the mean and uncertainty of $\bar{\theta}$ based on the previous i nodes as follows:

$$\begin{aligned} \bar{\theta}_{i+1} &= \frac{\left(i\bar{\theta}_i + \bar{\theta}_{i+1}\right)}{i+1} \\ P_{i+1}^{\bar{\theta}} &= \frac{iP_i^{\bar{\theta}} + \left[\bar{\theta}_{i+1} - \bar{\bar{\theta}}_{i+1}\right]\left[\bar{\theta}_{i+1} - \bar{\bar{\theta}}_{i+1}\right]^T}{i+1} \end{aligned} \tag{6}$$

Obtaining appropriate spatial points is now addressed. Spatial points are a visible set of fiducials that are tracked during sensor motion. As the sensor moves, the fiducials move relative to the sensor, eventually moving out of the sensor view. This requires methods to identify and track new fiducials. While traversing to the child node, fiducials are selected automatically from the image based on the visual contrast of the sampled point with its surroundings, giving a fiducial evaluation function (FEF) defined as:

$$\text{FEF} = g(C(u, v)) \tag{7}$$

where $g(C(u, v))$ is proportional to the contrast (C) multiplied by the window size (w), and contrast is defined as:

$$C(u, v) = \frac{I(x) - \bar{I}_w}{\bar{I}_w} \tag{8}$$

where $I(x)$ is the 2-D image intensity value of the potential fiducial at x, $\bar{I}_w$ is the average intensity of a window centered at the potential fiducial in the 2-D image, and w is the maximum window size after which the contrast starts to decrease. A penalty is added if a potential fiducial is too close to other identified fiducials or too far from the tracked target. Note however that closeness of the fiducials cannot be guaranteed because the single camera cannot provide 3-D data. Therefore, fiducials are automatically selected based on their apparent closeness in a 2-D sense. This is not an issue since fiducials are constantly being selected during the approach to the target node. Using the identified fiducials, sensor motion can be identified. Fiducials can be tracked with simple methods such as region growing or image disparity correspondence.

Figure 8 shows four steps in guiding the mobile robot to the target child node. The node is tracked using a simple region growing method. For the example shown in Fig. 8, in

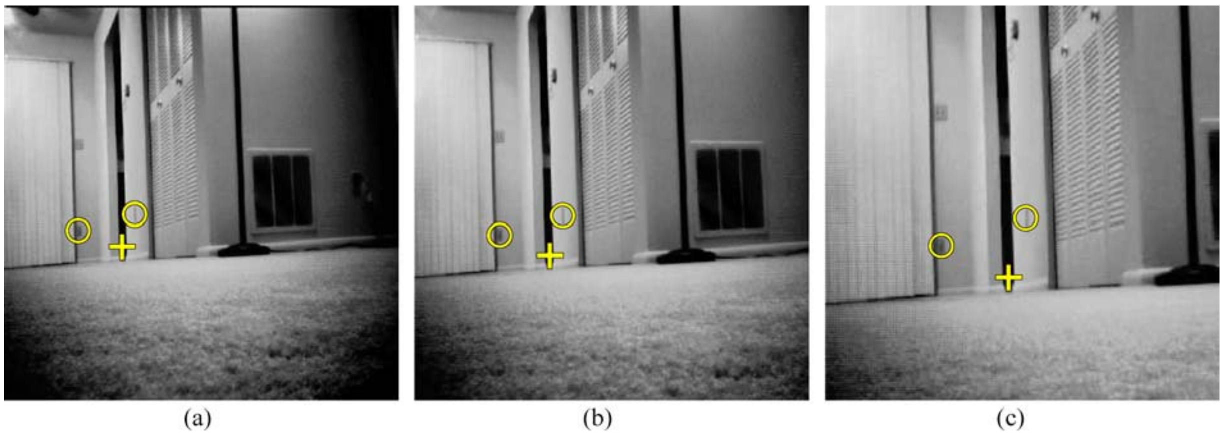


Fig. 8 Steps in tracking to an unexplored child node

addition to the primary target (+), two redundant fiducials are selected and tracked (O). These fiducials were selected close to the primary target. This permits fewer re-identifications of fiducials. However, this may result in higher inaccuracy in the visual tracking controller.

Using both the measure of target heading as well as an odometric measure on distance traversed, the mobile robot is able to generate a graph of the environment. Although the distance and direction estimates between parent and child nodes may be inaccurate, they serve as key parameters in estimating future movements along the graph. The process to return from a child node to its parent is described next.

5. Algorithm component 3: Traverse to parent node

Moving to a parent node requires a process quite different from the process of moving to an unexplored child node. The process is shown in Fig. 9. It is important to note that the process described in this section is used to approach any node that has been previously visited. Thus, once the map has been built, this process is used by the robot to navigate throughout the entire graph and around its environment. It can be used to traverse either from child to parent or from parent to child, as long as both locations (landmarks) include previously taken panoramic images to be compared.

5.1. Identifying parent node direction

To return to a parent node, the robot must first identify the direction of this node. As described above, using an Extended Kalman Filter, the mobile robot maintains a measure of the heading as it moves from node to node. Additionally, odometric measures of distance are also maintained. The measure of heading from the graph is used to determine the correct direction to point at. However, odometric errors would prevent the robot from performing accurate rotation without feedback. To compensate for these errors, an estimate of the expected image at the parent node facing away from the child node position may be generated from the panoramic scan previously obtained at that target node. In other words, if the robot were heading in the correct direction, then part of the image it "sees" should be a scaled version of the image that it would see if it were located at the target node, facing in the correct direction, see Fig. 10.

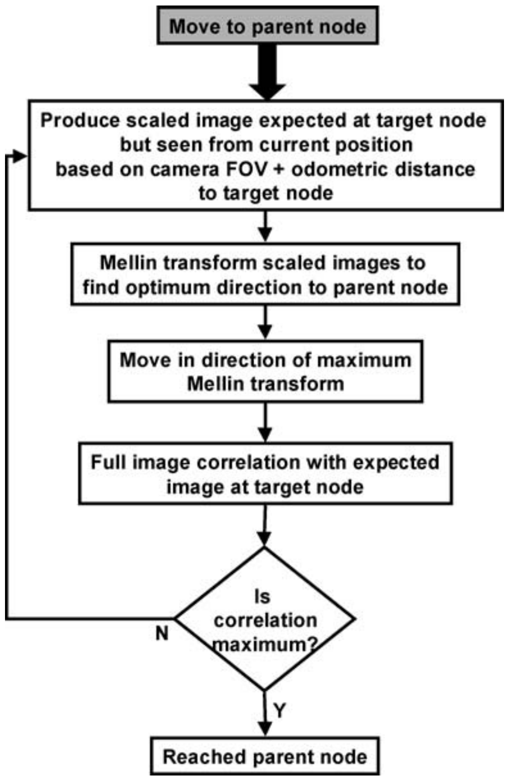

Move to parent node
Produce scaled image expected at target node but seen from current position based on camera FOV + odometric distance to target node
Mellin transform scaled images to find optimum direction to parent node
Move in direction of maximum Mellin transform
Full image correlation with expected image at target node
Is correlation maximum?
N
Y
Reached parent node

Fig. 9 Flow diagram to traverse to parent node

In general, these acquired images may be transformed relatively to the templates (rotated, scaled and/or translated). This can be resolved by either setting up a large set of templates that cover all possible transformations, or by finding an appropriate mathematical image transform that is invariant to image rotation, scale and/or translation. There are many different kinds of image invariants such as moment, algebraic and projective invariants (Casasent and Psaltis, 1978; Poularikas, 1998; Ruanaidh and Pun, 1997).

In this work, a Mellin transform is used to determine whether the image that the robot currently sees is what it would expect to see if it were pointed in the correct direction toward the target node. Note however from Fig. 10 that, without any range information, the overlap dimension of the images cannot be uniquely determined. The farther the destination node is from any wall or other feature in the environment, the smaller will be the difference between the views from the parent and child locations, leading to a larger overlap window. Because the robot does not have access to 3-D data, it cannot estimate its distance to the walls and therefore the scale of the overlap window. To overcome this limitation, a window growing approach is used. Here, a minimum sized overlap window is used for scale comparison. Progressively, this window size is increased until the correlation between the magnitude of the Mellin transforms of both images starts decreasing. The correlation process uses the Euclidean distance d_{MIt} between the magnitude of the Mellin transform of a window of the observed image, $M[I(\bar{x})]$, and the magnitude of the Mellin transform of the expected image, $M[t(\bar{x})]$, to correlate them:

$$d_{\mathrm{MIt}}^2 = \sum_{\bar{x}} \{M[I(\bar{x})] - M[t(\bar{x})]\}^2 \tag{9}$$

For a vector $\bar{x}$ of two dimensions, such as an image coordinate (u, v), it is meant that

$$\sum_{\bar{x}} \equiv \sum_{x=-M}^{M} \sum_{y=-N}^{N} \tag{10}$$

where M and N define the size of the expected image. If the magnitudes of the Mellin transforms of the expected

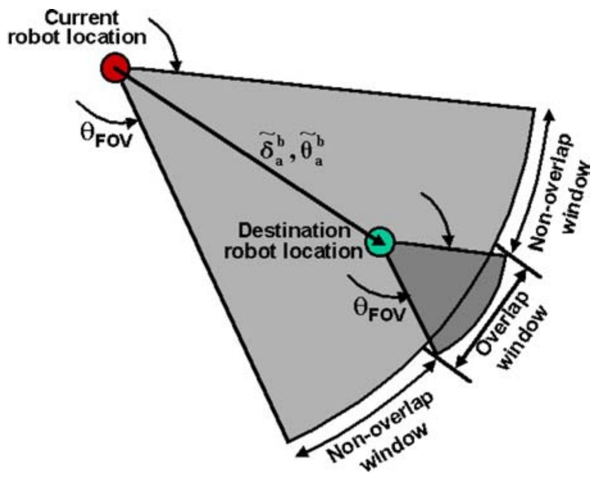


Fig. 10 Overlap between child and parent node views

and observed images are exactly the same, then the defined Euclidean distance is zero. Expanding the expression yields:

$$d_{\mathrm{MIt}}^2 = \sum_{\bar{x}} \{M[I(\bar{x})]^2 - 2 \cdot M[I(\bar{x})] \cdot M[t(\bar{x})] + M[t(\bar{x})]^2\} \tag{11}$$

The last term is constant and can be neglected. When $\sum M[I(\bar{x})]^2$ is approximately constant, it too can be neglected leaving the cross correlation between the magnitudes of the Mellin transforms of I and t:

$$R_{\mathrm{MIt}} = \sum_{\bar{x}} M[I(\bar{x})] \cdot M[t(\bar{x})] \tag{12}$$

This term would be maximum (and hence the distance measure is minimum) if the magnitudes of the Mellin transforms of the observed and expected images were the same.

The overlap window (centered at the observed image) that yields the maximum correlation R_{MIt} is then obtained, and the correspondent value of R_{MIt} stored. The robot then turns by a small predefined angle and repeats the overlap window correlation process. The process goes on as the robot searches in several directions within the tolerance of the odometric heading errors. The direction that yields the maximum correlation R_{MIt} between the Mellin transforms is determined to be the correction heading to the target node. In summary, gross adjusting of the robot heading is done using an Extended Kalman Filter and odometric measures of distance/rotation, however the actual (fine) tuning of the desired direction must be done using the Mellin transform. The main reason to use dead reckoning is to speed up the process by reducing the heading search space for the Mellin transform.

As an example of correlation of Mellin transforms, consider three simple images shown in Fig. 11: a white square, a circle and a scaled (large) version of the square, on a black background. Figure 12 shows the magnitude of the Mellin transforms of the three images. Note that the magnitude of the Mellin transforms of both squares are identical. It is found that the normalized cross correlation between Figs. 12(a) and (b) is 0.828, while the one between Figs. 12(a) and (c) is exactly 1.0, the maximum possible value. Therefore, the magnitudes of the Mellin transforms of the square and circle have a much lower cross correlation than the ones for both squares, as expected due to the invariance to scale property.

Figure 13 shows three images considered in determining the correct direction to the parent node from a child node. Figure 13(a) shows the view that the robot would see if it were at the parent node directed away from the child node, i.e., facing 180° from the direction of the child node. Figure 13(b) shows a view from the child node not pointing

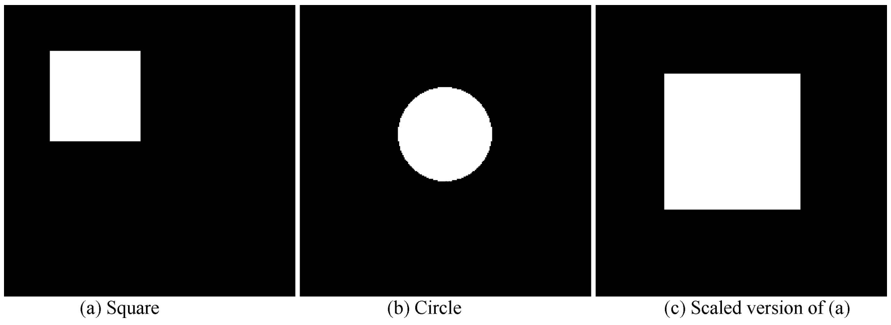


Fig. 11 Three basic images to exemplify the performance of Mellin transforms

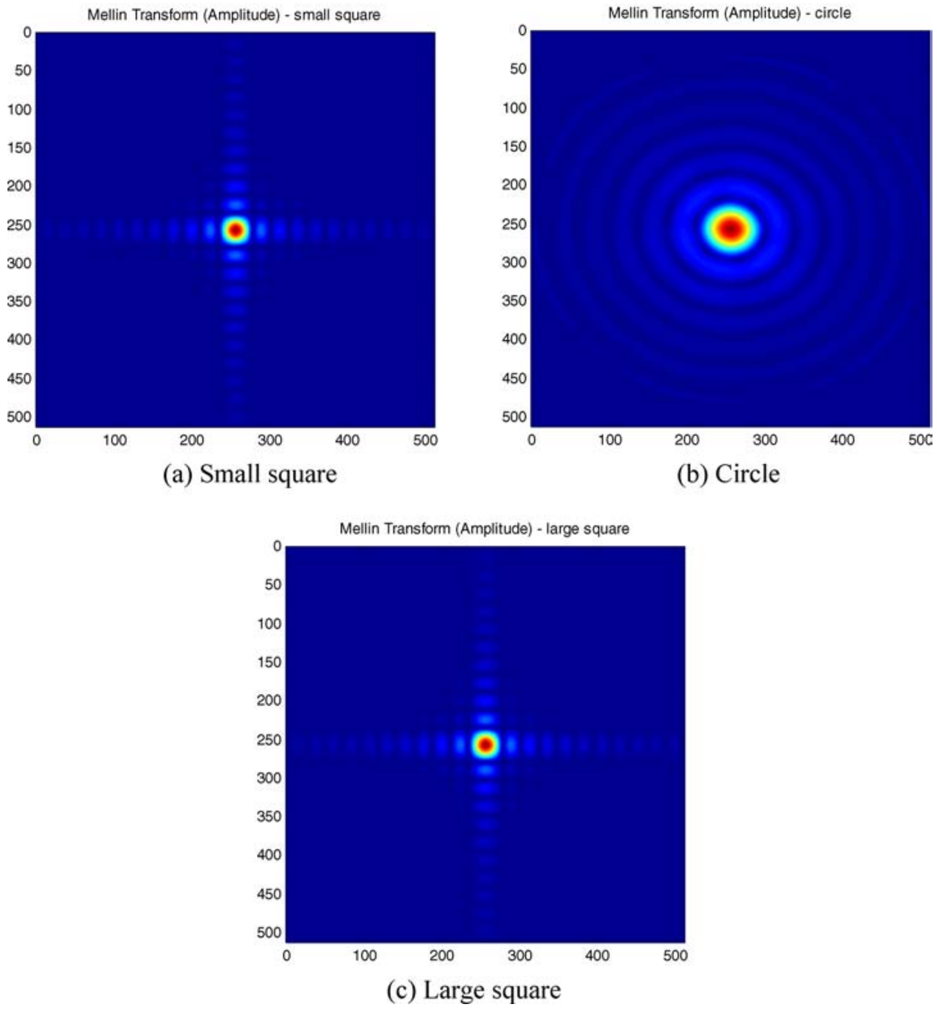


Fig. 12 Mellin transforms of the three basic images in Fig. 11

in the correct direction to the parent node (with the robot slightly misaligned to the left). Figure 13(c) shows a view from the child node while pointing in the correct direction to the parent node.

It is found that the correlation R_{MIt} between the magnitudes of the Mellin transforms of Figs. 13(a) and (c) is much higher than the one obtained from Figures 13(a) and (b). This can be seen e.g. by the peak value of the product $M[I(\bar{x})] \cdot M[t(\bar{x})]M$, which is 2.26×10^{15} in the first case and only 1.15×10^{14} in the second. It is clear from this example that the correct direction to the parent node, shown in Fig. 13(c), may be accurately determined.

An additional improvement can be implemented in the component 3. It has been shown that the amplitude of the Mellin transform is invariant to scale. In addition, it is well known that the amplitude of the Fourier transform is invariant to translation, while its scale is inversely proportional to the scale of the original function or image. Therefore, the amplitude of the Mellin transform of the amplitude of the Fourier transform of an image is invari-

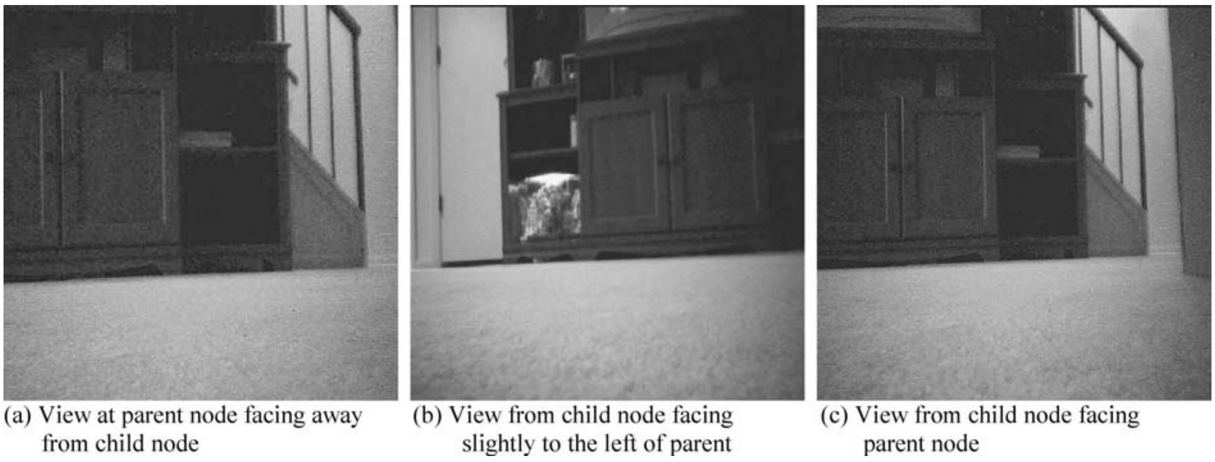

(a) View at parent node facing away from child node (b) View from child node facing slightly to the left of parent (c) View from child node facing parent node

Fig. 13 Determining appropriate view to parent node

ant to both scale and translation. This refinement has been tested on the experimental system, significantly improving its robustness.

The presented invariants work well when considering a flat-floored environment. However, even these ideal environments are subject to small image rotations caused by floor irregularities. An even more robust invariant can be obtained, which is invariant to scale, translation and rotation. This can be done by means of a log-polar mapping. Consider a point $(x, y) \in R^2$ and define

$$x = e^{\mu} \cos\theta, \quad y = e^{\mu} \sin\theta \tag{13}$$

where $\mu \in R$ and $0 \leq \theta < 2\pi$. One can readily see that for every point (x, y) there is a point (μ, θ) that uniquely corresponds to it. The new coordinate system is such that both scaling and rotation are converted to translations.

At this stage one can implement a rotation and scale invariant by applying a translation invariant such as the Fourier transform in the log-polar coordinate system. This transform is called a Fourier-Mellin transform. The amplitude of the Fourier-Mellin transform is therefore rotation and scale invariant.

Finally, to implement a translation, rotation and scale invariant, one just needs to consider the amplitude of the Fourier-Mellin transform of the amplitude of the Fourier transform of the image. This would improve even more the robustness of the algorithm component 3, however at the cost of a larger computational effort. For the experiments performed in this work on a flat-floored environment, the SLAM algorithm was successfull in finding all parent nodes within the graph without the need of such additional refinement.

5.2. Approach to parent node

Once the direction to the parent node is established, the robot moves toward the node. Again, visual servo control based on the correlation between the current image and the image the robot would see (if it were at the parent node pointed in the same direction) governs if the robot has reached the parent node. Unlike the correlation between Mellin transforms used in the previous section, this is a correlation between the entire image observed and the one that would be expected. This process uses the Euclidean distance d_{It} between the observed image, $I(\bar{x})$, and the expected image,$t(\bar{x})$, to correlate them:

$$d_{\mathrm{It}}^2 = \sum_{\bar{x}} [I(\bar{x}) - t(\bar{x})]^2 \tag{14}$$

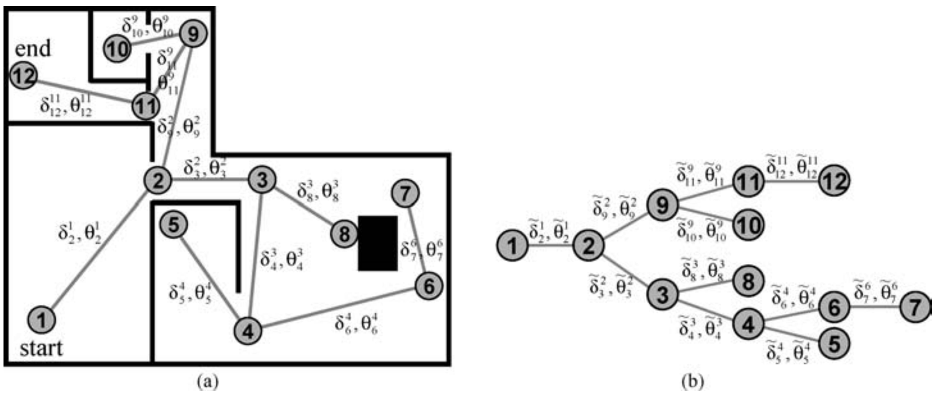
If the expected image and the observed image align exactly, then the distance between them is zero. For any misalignments, the distance is greater than zero. If the image intensity “energy” $\sum I(\bar{x})^2$ is approximately constant across the image, then minimizing Eq. (14) would be equivalent to maximizing the cross correlation between I and t:

$$R_{\mathrm{It}} = \sum_{\bar{x}} I(\bar{x}) \cdot t(\bar{x}) \tag{15}$$

The cross correlation value is then continuously evaluated as the robot moves on a straight line towards the parent node. When R_{It} starts to decrease, it is expected that the robot has reached the target node. To improve the visual servo control accuracy, fiducials taken from the desired view at the parent node could also be used if correspondent ones were to be found in the current view of the robot.

This component of the algorithm works very well if lateral slip can be ignored due to high friction surfaces (such as carpets or linoleum tiling in indoor environments). However, if significant drift occurs, then it is possible that R_{It} starts to decrease before the robot reaches the parent node, due to translational mismatches between the images. This can be dealt with by repeating the search for the correct heading (using Mellin transforms to recheck its directionality) as

Fig. 14 (a) Mapped environment with node locations, and (b) environment model



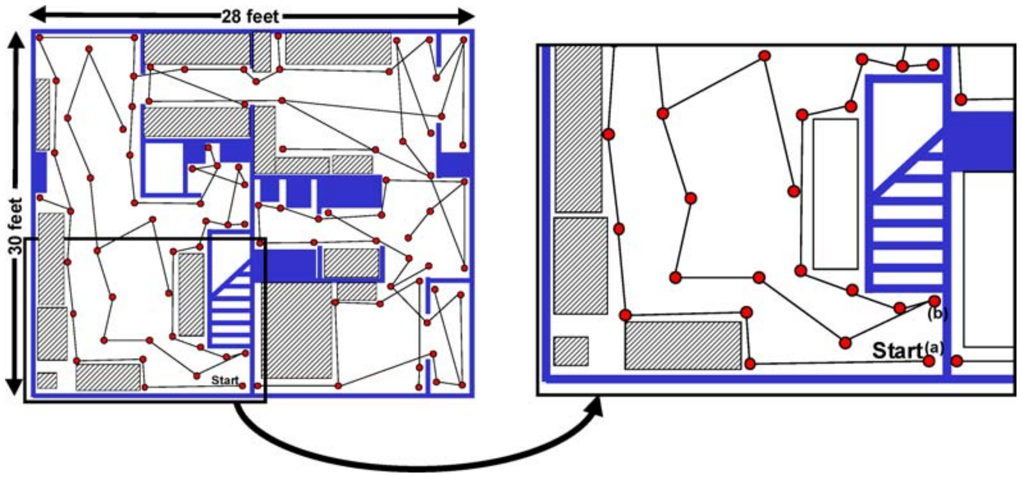


Fig. 15 Mapped environment with node locations

soon as R_{It} reaches its local maximum. If the current heading is in the same direction of maximum R_{MIt} recalculated from Eq. (12), within a user-defined tolerance, then the robot has reached the target node. Otherwise, the robot rotates to the direction of maximum R_{MIt} and, assuming it is still on the line segment connecting the child to parent nodes, the approach process resumes. This directionality verification significantly increases the robustness of the algorithm.

Finally, the robot finishes mapping the entire environment when all new potential child nodes have been explored. After that, the robot only needs to use the algorithm component 3 to travel from parent to child nodes and vice-versa within the mapped environment.

6. Experimental results

The proposed algorithm is applied to the exploration of two apartments by the mobile robot agent shown in Fig. 3. The results of the experimental exploration of the flat-floored environments are shown in Figs. 14 and 15. Each node is marked and linked to its parent/child nodes. These maps may then be used for navigation by the robot within its environment. Note in the figures that the node locations were estimated, however the actual trajectories between nodes during the exploration phase, which are irrelevant to the subsequent navigation tasks, were not stored. Straight lines were used to connect the nodes in the figures, which would be the ideal trajectory between any node pair.

The environment shown in Fig. 14(a) is a small 10.2 m by 8.0 m one-bedroom apartment. Each node on the graph consists of a panoramic scan at that node, while each branch on the graph consists of a heading and distance between the two connected nodes. In this map, each node j is located at a distance δ^i_j from its parent node i, at an angle θ^i_j (with respect to the original starting angle). However, in this algorithm absolute distances and angles cannot be measured accurately. Dead reckoning errors (e.g. due to wheel slip) do not allow for very accurate measurements in distance by a wheel odometer (even though these errors are significantly reduced by the applied Extended Kalman Filter). Thus the distance and angle of node j from its parent node i is represented by the estimates $\tilde{\delta}^i_j$ and $\tilde{\theta}^i_j$ respectively, see Fig. 14(b). Results for a larger apartment are shown in Fig. 15.

Note that the walls and furniture were later added to the figures, since the absence of range sensors or stereo vision prevents the robot from identifying their exact location. Still, the robot can visually recognize the existence of such walls and localize itself within its surroundings using the generated nodal map. This is an important feature of the proposed algorithm, which seeks an efficient map, and not a complete one. At the end, the robot does not have a standard floor plan of the environment, instead it has a very unique representation consisting solely of a graph with panoramic images taken at each node, which is enough to localize itself. This significantly reduces the memory requirements and ultimately the computational cost of such mobile robots. In addition, it is possible to include some human interaction during the mapping process, giving names to each environment the robot explores (such as "kitchen", "living room", etc.), which would improve the functionality of the system to be used as, e.g., a home service robot.

From the performed experiments it is found that in all cases the robot is able to localize itself, always arriving at the desired nodes, due to the efficiency of the Mellin transform. Note that the error in reaching a desired node is not a function of how far down the graph the node is, because the Mellin transform will be only considering the image that the target node has in memory. However, the accuracy will be a function of what the image is like, i.e., a more "busy" image will likely result in less position error as the multitude of features will help in the correlation. However, too busy an image will result in some degradation. Additionally, if all objects in the image are quite far away, then the image will not change much as the robot moves. This will result in degradation with distance. It has been found that the positioning error of the robot is directly proportional to its average distance to the walls and obstacles in its field of view, and inversely proportional to the camera resolution. For several test runs on our experimental system in indoor environments, with distance from walls ranging in average between one and ten meters, and with a (limited) camera resolution of 176×144 pixels, an average positioning error of 60 mm (RMS) has been obtained. This positioning accuracy can certainly be improved with a better camera resolution.

An important limitation may be observed here. When navigating in the environment, the robot would use the graph to find a path from one node to another. Although simple search methods may be applied to find the optimum path through the graph, this path may in fact be spatially circuitous due to the manner in which the graph was developed. For example, to get from node (a) to (b) in Fig. 15 by traversing the graph developed, the robot would take a long path through the environment, whereas a short direct path is obtained by inspection. Current work to develop an algorithm that bridges nodes in a graph while avoiding obstacles between them has been performed, based on genetic algorithms. The key idea is to find a set of new link candidates (bridges) to be added that would most efficiently reduce the average traveled distance between any two nodes in the graph. This set would form a cromossome in a classical genetic algorithm, where the fitness function would be inversely proportional to the average traveled distance, along the graph, between each and every node pair. Then each new link candidate from the optimal set is explored by the robot to check for the presence of walls or furniture. If the path between such nodes is clear, then a new link is added to the graph. Experimental results have validated this approach. Remolina and Kuipers (2004) have also described other techniques to bridge nodes in a graph, which is the equivalent of building a topological map.

Another issue is that there is no guarantee that the robot will explore the entire workspace. The robot will only explore a certain region if there is enough information content in that area. This issue can be significant if the considered area lacks texture, since texture is the key to information content. However, this can be dealt with by calibrating the cutoff threshold of the information content in the algorithm that identifies potential child nodes. This can be done automatically by repeating the first component of the algorithm until a minimum pre-defined number of child nodes is identified, in average, at each panoramic image.

Finally, the algorithm can also be improved to handle issues such as "kidnapping" of the robot followed by a random placement within the environment, at a point that in general is not a landmark with an associated panoramic view. If the robot is randomly placed in an environment that it has properly mapped, it needs to first make a panoramic map of its location. Then it needs to work through the list of mapped nodes and compare what it sees to what the other nodes have seen. It actually needs to do what it would do in the algorithm component 3, except that it needs to check all nodes to find a best match as to which direction to move. This improvement will be addressed in future work in algorithm optimization.

7. Conclusions

In this work, an information-based iterative algorithm has been presented to plan the visual exploration strategy of an autonomous mobile robot to enable it to most efficiently build a graph model of its environment using a single base-mounted camera. The algorithm is based on determining the information content present in sub-regions of a 2-D panoramic image of the environment as seen from the current location of the robot. Using a metric based on Shannon's information theory, the algorithm determines potential locations of nodes from which to further image the environment. Using a feature tracking process, the algorithm helps navigate the robot to each new node, where the imaging process is repeated. A Mellin transform and tracking process is used to guide the robot back to a previous node, using previously taken panoramic images as a reference. This imaging, evaluation, branching and retracing its steps continues until the robot has mapped the environment to a pre-specified level of detail. The set of nodes and the images taken at each node are combined into a graph to model the environment. By tracing its path from node to node, a service robot can navigate around its environment. Experimental studies were conducted to demonstrate the effectiveness of the entire algorithm. It was found that this algorithm allows a mobile robot to efficiently localize itself using a limited sensor suite (consisting of a single monocular camera, contact sensors, and an odometer), reduced memory requirements (only enough to store one 2-D panoramic image at each node of a graph), as well as modest processing capabilities (due to the computational efficiency of Mellin transforms and the proposed information-based algorithm). Therefore, the presented approach has a potential benefit to significantly reduce the cost of autonomous mobile systems such as indoor service robots.

References

Alata, O., Cariou, C., Ramananjarasoa, C., and Najim, M. 1998. Classification of rotated and scaled textures using HMHV spectrum estimation and the Fourier-Mellin transform. In *Proceedings of the International Conference on Image Processing ICIP 98*, Vol. 1, pp. 53–56.

Anousaki, G.C. and Kyriakopoulos, K.J. 1999. Simultaneous localization and map building for mobile robot navigation. *IEEE Robotics and Automation Magazine*, 6(3):42–53.

Borenstein, J. and Koren, Y. 1990. Real time obstacle avoidance for fast mobile robots in cluttered environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 572–577.

Casasent, D. and Psaltis, D. 1978. Deformation invariant, space-variant optical pattern recognition. *Progress in Optics*, Vol. XVI, North-Holland, pp. 289–356.

Castellanos, J.A., Martinez, J.M., Neira, J., and Tardos, J.D. 1998. Simultaneous map building and localization for mobile robots: A multisensor fusion approach. In *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1244–1249.

Evolution Robotics, 2005. homepage: www.evolution.com.

Gelb, A. 1974. *Applied optimal Estimation*. MIT Press, Cambridge, Massachusetts, USA.

Huntsberger, T., Sujan, V.A., Dubowsky, S., and Schenker, P. 2003. Integrated System for Sensing and Traverse of Cliff Faces. In *Proceedings of the SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls: Symposium on Unmanned Ground Vehicle Technology V*, Orlando, Florida, USA.

Khatib, O. 1999. Mobile manipulation: The robotic assistant. *Journal of Robotics and Autonomous Systems* 26:175–183.

Kruse, E., Gutsche, R., and Wahl, F.M. 1996. Efficient, iterative, sensor based 3-D map building using rating functions in configuration space. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1067–1072.

Kuipers, B. 2000. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233.

Lavery, D. 1996. *The Future of Telerobotics*. Robotics World, Summer 1996.

Lawitzky, G. 2000. A navigation system for cleaning robots. *Autonomous Robots*, 9:255–260.

Leonard, J.J. and Durrant-Whyte, H.F. 1991. Simultaneous map building and localization for an autonomous mobile robot. *IEEE 1991 International Workshop on Intelligent Robots and Systems*, Vol. 3, pp. 1442–1447.

Nister, D. 2003. Preemptive RANSAC for live structure and motion estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 1, pp. 199–206.

Poularikas, A.D. 1998. *The Handbook of Formulas and Tables for Signal Processing*. CRC Press and IEEE Press.

Remolina, E. and Kuipers, B. 2004. Towards a general theory of topological maps. *Artificial Intelligence*, 152:47–104.

Reza, F.M. 1994. *An Introduction to Information Theory*. Dover, New York.

Roumeliotis, S.I. and Rekleitis, I.M. 2004. Propagation uncertainty in cooperative multirobot localization: Analysis and experimental results. *Autonomous Robots*, 17:41–54.

Ruanaidh, J.O. and Pun, T. 1997. Rotation, scale and translation invariant digital image watermarking. In *Proceedings IEEE International Conference on Image Processing (ICIP 97)*, Vol. 1, Santa Barbara, CA, USA, pp. 536–539.

Se, S., Lowe, D.G., and Little, J. 2002. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758.

Simhon, S. and Dudek, G. 1998. Selecting targets for local reference frames. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 2840–2845.

Smith, S.W. 1999. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd edn. California Technical Publishing, San Diego, CA.

Sujan, V.A., Dubowsky, S., Huntsberger, T., Aghazarian, H., Cheng, Y., and Schenker, P. 2003. Multi agent distributed sensing architecture with application to cliff surface mapping. In *Proceedings of the 11th International Symposium of Robotics Research (ISRR),* Siena, Italy.

Sujan, V.A. and Meggiolaro, M.A. 2005. On the visual exploration of unknown environments using information theory based metrics to determine the next best view. In: *Mobile Robots: New Research*, edited by X.J. Liu, Nova Publishers, USA, 2005.

Taylor, C.J. and Kriegman, D. 1998. Vision-based motion planning and exploration algorithms for mobile robots. *IEEE Trans. on Robotics and Automation*, 14(3):417–426.

Thrun, S., Burgard, W., and Fox, D. 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, San Francisco, CA, pp. 321–328.

Thrun, S. 2003. Robotic mapping: A survey. In: *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1–35.

Tomatis, N., Nourbakhsh, I., and Siegwar, R. 2001. Simultaneous localization and map building: A global topological model with local metric maps. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 421–426.

Wong, S., Coghill, G., and MacDonald, B. 2000. Natural landmark recognition using neural networks for autonomous vacuuming robots. In *Proceedings of the 6th International Conference on Control, Automation, Robotics and Vision, ICARCV'00,* Singapore.

World Robotics, 2001. *Statistics, Market Analysis, Forecasts, Case Studies and Profitability of Robot Investment*. Produced by the United Nations Economic Commission for Europe (UNECE) in cooperation with the International Federation of Robotics (IFR). No. GV.E.01.0.16 or ISBN No. 92-1-101043-8.