MOBILE ROBOT LOCALIZATION AND MAPPING USING SPACE INVARIANT TRANSFORMS

Vivek Anand Sujan
Air Handling and Combustion Control Division, Cummins Engine Company
Columbus, IN 47201, USA
e-mail: vivek.a.sujan@cummins.com

Felipe Augusto Weilemann Belo
Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro
R. Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil
e-mail: felipe-belo@uol.com.br

Marco Antonio Meggiolaro
Department of Mechanical Engineering, Pontifical Catholic University of Rio de Janeiro
R. Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil
e-mail: meggi@mec.puc-rio.br

Abstract. *For many applications it is not usually practical to provide mobile 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 space invariant 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. By tracing its path from node to node, a service robot can navigate around its environment. Experimental results show the effectiveness of this algorithm.*

Keywords*: service robots, visual mapping, self-localization, information theory, Mellin transform*

1. Introduction

The market for medical robots, underwater robots, surveillance robots, demolition robots, cleaning robots and many other types of mobile robots for carrying out a multitude of services has grown significantly in the past decade. Presently most service robots for personal and private use are found in the areas of household robots, which include vacuum cleaning and lawn-mowing robots, and entertainment robots, including toy and hobby robots. The sales of mobile robots are projected to exceed the sales of factory floor robots by a factor of four (Lavery, 1996). Therefore the need to develop efficient localization and mapping algorithms for such systems.

Mobile robot environment mapping 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 et al., 1999; Castellanos et al., 1998; Tomatis et al., 2001). These algorithms have been implemented for several different sensing methods, such as stereo camera vision systems (Castellanos et al., 1998), laser range sensors (Tomatis et al., 2001), and ultrasonic sensors (Anousaki et al., 1999).

On the other hand, the existing SLAM algorithms typically require a relatively expensive sensor suite such as stereo vision systems. Commercial applications for service robots in large scale would only be possible after simplifications in hardware resources, which in turn would result in lower setup costs, maintenance costs, computational resources and interface resources.

In this work, an unknown environment exploration and modeling algorithm is developed to accurately map and navigate the robot agent in a well-illuminated flat-floored environment, using only information from a single monocular camera, wheel encoders and contact switches. Therefore, this algorithm has the potential to significantly reduce the cost of autonomous mobile systems such as indoor service robots. The proposed algorithm, named here IB-SLAM (Information-Based Simultaneous Localization and Mapping), is unique in that it utilizes the quantity of information (entropy) of the environment in order to predict high information yielding viewpoints from which to continue exploring the environment. This has been shown to result in a significantly more efficient and robust exploration process as compared to other conventional approaches (Sujan et al., 2003; Sujan et al., 2004). The algorithm is described next.

2. Simultaneous localization and mapping algorithm overview

The proposed environment exploration and modeling algorithm 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. 1. 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 (Fig. 1). 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.

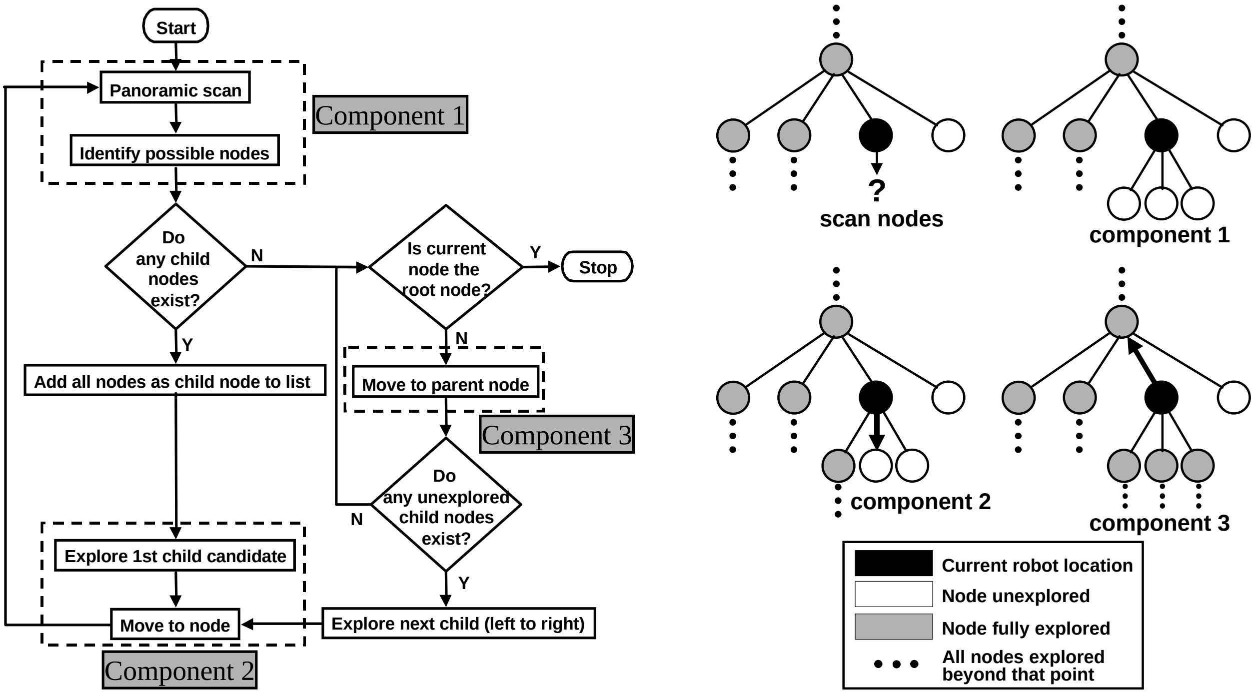

Figure 1. Overview of the exploration and modeling algorithm (left), with its key components (right)

The third component of the algorithm is to traverse to a parent node from the current node (Fig. 1). This is done when all the child nodes of the current node have been completely explored, allowing the other child nodes of the parent node to be explored. To return to a parent node the robot must first identify the direction of the parent node, which requires a process quite different than the process of moving to a child node. This is done using a Mellin transform (Alata et al., 1998) 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.

These three components – imaging and evaluation, branching and retracing its steps – continue until the robot has mapped the environment to a 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 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).

Experimental results are presented along each section, obtained from a mobile robot agent adapted from an ER-1 commercial system (Evolution Robotics, 2004), see Fig. 2. The system consists of a 2-wheel differential system driven by step motors, equipped with wheel encoders (for odometry), three bump switches for measuring contact, and a single color camera (640 × 480 pixels resolution) mounted to its base. The robot is controlled by a 1.5GHz Pentium IV notebook mounted on its structure. It is used to explore an apartment, generating the graph shown in Fig. 2 (right). Note that 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. The walls and obstacles were later added to this figure, because the robot cannot evaluate its distance from them due to the absence of range sensors or stereo cameras. Still, the robot can visually recognize the existence of such walls and localize itself within its surroundings using the generated nodal map. In the next sections, the details of the three key components of the algorithm are developed.

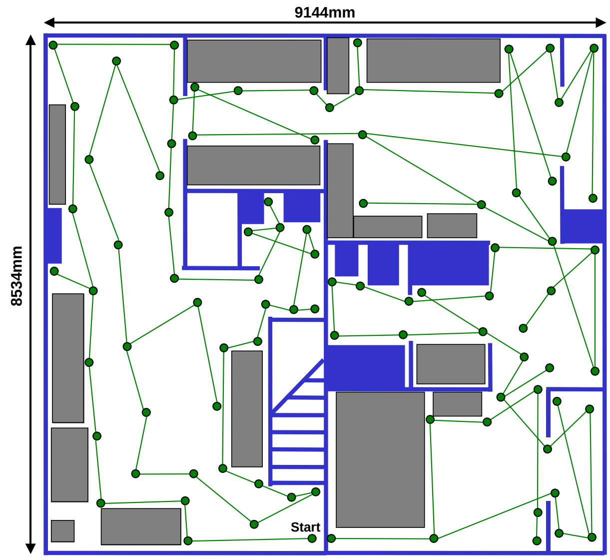

Figure 2. Mobile robot experimental system (left) and map of the environment with node locations (right)

3. Child node identification

The first component of the IB-SLAM 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. Conventional image joining operations can then be used to fuse the panoramic image together. For example, two images may be joined by identifying fiducials common to both to find a transformation between them. An example can be seen in Figure 3, taken from the experimental system.

Figure 3. Panoramic scan (360°) taken and joined by the experimental system

The key idea to identify potential child nodes is to reduce the acquired panoramic image using a quadtree decomposition method. In a quadtree decomposition, the 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.

But highly textured surfaces (such as furniture, carpets, curtains, wallpaper) or noise may also be identified as regions with high information content. Clearly these are regions that need not be explored and mapped. 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 (Sujan et al., 2003; Sujan et al., 2004) 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 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 k^{th} event. This definition of information may also be interpreted as the minimum number of states (bits) needed to fully describe a piece of data. Shannon's emphasis was in describing the information content of 1-D signals. In 2-D, the gray level histogram of an ergodic image can be used to define a probability distribution:

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

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 (Sujan et al., 2004). Thus, the information content is evaluated using the compressed version of the acquired panoramic image instead of the original image. The smooth gradient example would result in a much smaller compressed file than the random image, leading to a smaller information content, as expected.

3.2. Information-based node identification

The information theory developed above is used to identify potential child nodes. The algorithm breaks down the environment map into a quadtree of high information regions. The resulting image is divided into four quadrants. The information content of the compressed version of each quadrant is evaluated using a lossless compression algorithm and 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 cutoff threshold of information is based on a critical robot physical parameter (e.g. the wheel diameter).

After 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 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. The centroids of each blob 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.

Figure 4 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. 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, see Fig. 4. 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.

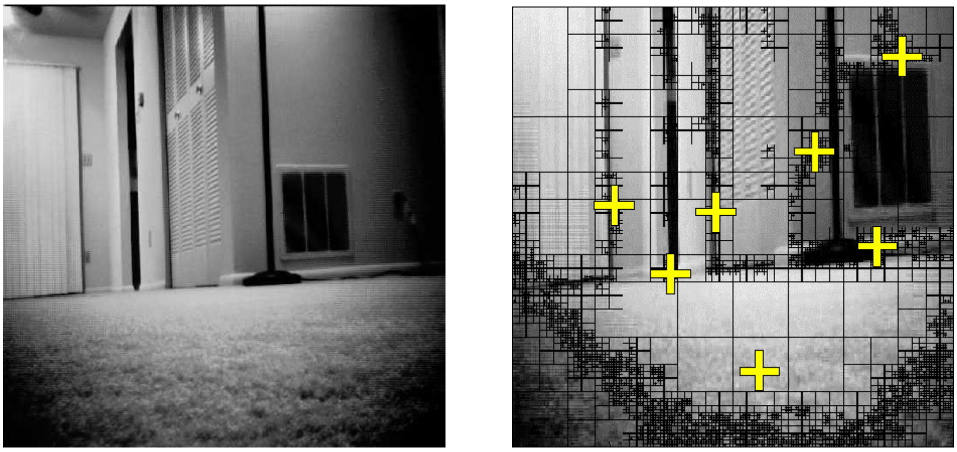

Figure 4. Raw image taken by mobile robot onboard camera (left) and processed image with trimmed elevation, showing quadtree decomposition and identified child nodes (right)

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. Traverse to unexplored node

The second component of the algorithm is to navigate the mobile robot to an unexplored child node from the current node. This is achieved by tracking one of the child (target) nodes obtained in the first component of the algorithm using a simple region growing tracker and a visual servo controller. The visual servo controller corrects for errors in heading of the mobile robot as it tracks the desired target. Target heading angle 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 image bottom (or top) edge, or the robot's motions are obstructed. At this point the child node is established, while its direction and odometric distance from the parent node are recorded. If the node cannot be reached by a straight line, then the point of obstruction is defined as the new child node, and 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.

Improved accuracy may be obtained by concurrently tracking multiple fiducials to determine the primary target heading, instead of tracking a single target point. This improved heading accuracy obtained through increased redundancy is a feature that can be turned on or off depending on how accurately one needs to move. In addition, an Extended Kalman Filter (Gelb, 1974) is used to maintain an estimate of the robot heading. Using both the estimate of target heading as well as an odometric measure on distance traversed, the mobile robot generates a graph similar to the one in Fig. 2. Although these measures 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. Traverse to parent node

The third component of the algorithm is to move back to a parent node. Moving to a parent node requires a process quite different from the process of moving to an unexplored child node. 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.

To return to a parent node, the robot must first identify the direction of this node. 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. 5.

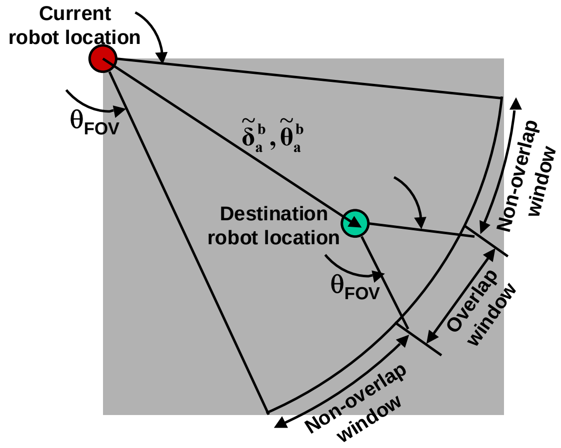

Figure 5. Overlap between the child and parent node views

There are several algorithms to correlate the images from the child and parent nodes. One approach is to use integral transform-based invariants. The number of robust invariant components is relatively large, which makes such transforms suitable for spread spectrum techniques. In addition, mapping to and from the invariant domain to the spatial domain is well-defined and it is in general not computationally expensive. The invariant used in this work is based on the Mellin transform (Alata et al., 1998). This transform is invariant to scale, therefore it is a good candidate to correlate views in the same direction taken from child and parent nodes, which are scaled versions of each other.

Let the image be a real valued continuous function f(x,y). Then f(x,y) can be discretized as an $N \times M$ matrix F with elements $f_{k,j}$, where $N \times M$ is the image size with pixel dimensions Δx and Δy, and

$$f_{k,j} \equiv f(k \cdot \Delta x,\ j \cdot \Delta y) \tag{3}$$

with k = 1, ..., N and j = 1, ..., M. The continuous Mellin transform $M(\omega_1, \omega_2)$ is a function of the frequencies ω_1 and ω_2. It can be discretized as an $U \times V$ matrix with elements $M_{u,v}$ considering frequency intervals $\Delta u/2\pi$ and $\Delta v/2\pi$ for ω_1 and ω_2 respectively,

$$M_{u,v} \equiv M(u \cdot \Delta u / 2\pi,\ v \cdot \Delta v / 2\pi) \tag{4}$$

with u = 1, ..., U and v = 1, ..., V, resulting in

$$M_{u,v} = \sum_{j=1}^{M} \sum_{k=1}^{N} \frac{1}{k\Delta x} e^{iu\Delta u \ln(k\Delta x)} f_{k,j} \frac{1}{j\Delta y} e^{iv\Delta v \ln(j\Delta y)} \tag{5}$$

The 2-D Mellin transform is then given by the product $T_x \cdot F \cdot T_y$, where

$$T_x = \frac{1}{\Delta x}\begin{bmatrix} e^{i\Delta u \ln \Delta x} & \frac{1}{2}e^{i\Delta u \ln 2\Delta x} & \cdots & \frac{1}{N}e^{i\Delta u \ln N\Delta x} \\ e^{i2\Delta u \ln \Delta x} & \frac{1}{2}e^{i2\Delta u \ln 2\Delta x} & \cdots & \frac{1}{N}e^{i2\Delta u \ln N\Delta x} \\ \cdots & \cdots & \cdots & \cdots \\ e^{iU\Delta u \ln \Delta x} & \frac{1}{2}e^{iU\Delta u \ln 2\Delta x} & \cdots & \frac{1}{N}e^{iU\Delta u \ln N\Delta x} \end{bmatrix} \quad T_y = \frac{1}{\Delta y}\begin{bmatrix} e^{i\Delta v \ln \Delta y} & e^{i2\Delta v \ln \Delta y} & \cdots & e^{iV\Delta v \ln \Delta y} \\ \frac{1}{2}e^{i\Delta v \ln 2\Delta y} & \frac{1}{2}e^{i2\Delta v \ln 2\Delta y} & \cdots & \frac{1}{2}e^{iV\Delta v \ln 2\Delta y} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{M}e^{i\Delta v \ln M\Delta y} & \frac{1}{M}e^{i2\Delta v \ln M\Delta y} & \cdots & \frac{1}{M}e^{iV\Delta v \ln M\Delta y} \end{bmatrix} \tag{6}$$

To identify the parent node direction, 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. 5 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 Mellin transforms of both images starts decreasing. The correlation process uses the normalized cross correlation R_M between the Mellin transform of a window of the observed image, M[I], and the Mellin transform of the expected image, M[t], to correlate them:

$$R_M = \frac{\sum_u \sum_v M_{u,v}[I] \cdot M_{u,v}[t]}{\sqrt{\sum_u \sum_v M_{u,v}[I]^2} \cdot \sqrt{\sum_u \sum_v M_{u,v}[t]^2}} \tag{7}$$

This term would be maximum if 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_M is then obtained, and the correspondent value of R_M 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_M 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.

Figure 6 shows three images considered in determining the correct direction to the parent node from a child node. Figure 6(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 6(b) shows a view from the child node not pointing in the correct direction to the parent node (with the robot slightly misaligned to the left). Figure 6(c) shows a view from the child node while pointing in the correct direction to the parent node.

(a) (b) (c)

Figure 6. Determining appropriate view to parent node

It is found that the correlation R_M between the Mellin transforms of Figures 6(a) and (c) is much higher than the one obtained from Figures 6(a) and (b). This can be seen by the peak value of the product $M_{u,v}[I]$ and $M_{u,v}[t]$, which is 2.26×10^{15} in the first case and only 1.15×10^{14} in the second, almost a factor of 20. It is clear from this example that the correct direction to the parent node shown in Fig. 6(c) may be accurately determined using Eq. (7).

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 R_M, this is a correlation between the entire image observed and the one that would be expected. The cross correlation value is then continuously evaluated as the robot moves on a straight line towards the parent node. When this correlation starts to decrease, it is expected that the robot has reached the target node.

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 third component of the algorithm to travel from parent to child nodes and vice-versa within the mapped environment.

Note that several nodes are identified, a few of them redundant. This can be calibrated by not exploring new child candidate nodes that are too close to previously identified nodes. Post-processing on the final graph can also be performed to eliminate some redundant nodes by bridging nodes. 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, but they are out of the scope of this work.

6. Experimental results

The results of the experimental exploration of a flat-floored apartment environment are shown in Fig. 2. Each node is marked and linked to its parent/child nodes. This map may then be used for navigation by the robot within its environment. Note that the walls and furniture were later added to the figure, since the absence of range sensors or stereo vision prevents the robot from identifying their exact location. However, the robot is able to recognize its environment, including walls and furniture, from the set of panoramic images taken at each landmark. 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, particularly since Mellin transforms are considerably fast and efficient. 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.

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. Experimental studies have been 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.

8. References

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

Anousaki, G.C., Kyriakopoulos, K.J., 1999, “Simultaneous localization and map building for mobile robot navigation”. IEEE Robotics & Automation Magazine, Vol.6, No.3, pp.42-53.

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

Evolution Robotics, 2004, homepage: www.evolution.com

Gelb, A., 1974, “Applied optimal estimation”. MIT press, Cambridge, Massachusetts, U.S.A.

Lavery, D., 1996, “The Future of Telerobotics”. Robotics World.

Reza, F.M., 1994, “An introduction to information theory”. Dover, New York.

Sujan, V.A., Dubowsky, S., Huntsberger, T., Aghazarian, H., Cheng, Y., Schenker, P., 2003, “Multi Agent Distributed Sensing Architecture with Application to Cliff Surface Mapping”. Proceedings of the 11th International Symposium of Robotics Research (ISRR), Italy.

Sujan, V.A., Meggiolaro, M.A., 2005, “Intelligent and Efficient Strategy for Unstructured Environment Sensing using Mobile Robot Agents”. In press at the Journal of Intelligent and Robotic Systems.

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

9. Responsibility notice