On-line SLAM Using Clustered Landmarks with Omnidirectional Vision

Jun Okamoto Jr.
jokamoto@usp.br
Escola Politécnica da Universidade de São Paulo
Department of Mechatronics and Mechanical Systems Engineering
Av. Prof. Mello Moraes, 2231
05508-030 São Paulo, SP, Brazil

Vitor Campanholo Guizilini
v.guizilini@acfr.usyd.edu.au
University of Sydney
Australian Centre for Field Robotics
96 City Road, Chippendale
NSW 2008 Sydney, Australia

The problem of SLAM (simultaneous localization and mapping) is a fundamental problem in autonomous robotics. It arises when a robot must create a map of the regions it has navigated while localizing itself on it, using results from one step to increase precision in another by eliminating errors inherent to the sensors. One common solution consists of establishing landmarks in the environment which are used as reference points for absolute localization estimates and form a sparse map that is iteratively refined as more information is obtained. This paper introduces a method of landmark selection and clustering in omnidirectional images for on-line SLAM, using the SIFT algorithm for initial feature extraction and assuming no prior knowledge of the environment. Visual sensors are an attractive way of collecting information from the environment, but tend to create an excessive amount of landmarks that are individually prone to false matches due to image noise and object similarities. By clustering several features in single objects, our approach eliminates landmarks that do not consistently represent the environment, decreasing computational cost and increasing the reliability of information incorporated. Tests conducted in real navigational situations show a significant improvement in performance without loss of quality.
***Keywords**: SLAM, SIFT, omnidirectional vision, mobile robot control*

Introduction

A solution to the problem of SLAM (Simultaneous Localization And Mapping) would be of inestimable value in robotics as it would lead to truly autonomous robots, capable of navigating safely at unknown locations in unknown environments using nothing but embedded equipment. Information from sensors cannot be used directly because they are inherently inaccurate, due to phenomena that cannot be modeled, as they are too complex or unpredictable. Probabilistic approaches (Thrun et al., 2005) have successfully dealt with both problems individually, such as mapping given the robot's exact position at each instant (Thrun, 2002) or localization given a precise map of the environment (Dellaert et al., 1999). However, in situations where neither one is known in advance the robot must estimate both simultaneously, a problem that is largely discussed in the autonomous robotic literature (Csorba, 1997; Bailey, 2002; Montemerlo, 2003; Fitzgibbons, 2004), but still lacks a closed, efficient and truly generic solution. Figure 1 illustrates these situations: Fig. 1(a) shows the robot trajectory and the map built with no error in the robot localization; Fig. 1(b) shows the robot localization error in a known map and Fig. 1(c) shows the map generated using only odometry estimates for the robot localization.

The classic approach to the problem of SLAM, first described by Smith et al. (1990) and implemented by Moutarlier and Chatila (1989), is based on detection and recognition of landmarks in the environment which can be used as reference points to eliminate odometry errors accumulated over time. A feature map of such landmarks is iteratively built by comparing new landmarks with the ones already stored in search for matches. If a match is found, this information is used to increase precision in both localization and mapping estimates; otherwise, it is added to the map for future correspondence. A substantial amount of research has been conducted to overcome some of the limitations in this approach, such as computational complexity and scalability (Leonard and Feder, 1999; Montemerlo, 2003) and data association problems (Thrun et al., 1998; Leonard et al., 2002).

A robot's ability to detect and recognize landmarks is limited by its sensors and how they interact with structures in the environment. Although a number of approaches have been proposed to address the problem of SLAM using range sensors (Press and Austin, 2004), vision sensors are well suited devices for an autonomous mobile robot, because they are information-rich and rarely have restrictions in range and applications. Recent increases in computational power and algorithm efficiency have led to numerous implementations of visual systems in many fields of robotics (Fitzgibbons, 2004; Andreasson and Duckett, 2004). Among visual sensors, the omnidirectional vision (Zhu, 2001) introduces several properties that are very desirable in most navigational tasks (Gaspar, 2003), including in the SLAM problem discussed above (Se et al., 2001). A larger field of view means ability to detect a higher number of landmarks, increasing characterization of environment as a whole by avoiding blind spots and poor angles for triangulation. Each landmark will also remain visible for a larger period of time, increasing number of matches and providing more information for improving localization and mapping estimates.

However, the high characterization of environments provided by visual sensors can also be a drawback due to the large amount of information obtained from a single image. This leads to a high computational cost necessary to process, maintain and access all this data, and also causes data association problems due to redundancy and image noise, generating estimates that do not correspond to reality and increase uncertainty of results. We describe in this paper a method for selective perform landmarks extraction that is based on clustering features directly from omnidirectional images, without any prior knowledge of the environment and therefore can be applied to any situation where visual sensors are capable of providing useful information (i.e. feature-rich scenarios). We start by briefly describing the problem of SLAM and the use of landmarks to ensure localization precision. After that the proposed method of landmarks selection is described, along with modifications in the matching step and landmarks management. Finally, we show results obtained in a real SLAM situation that indicate a significant gain in quality and efficiency over a common approach of landmark selection.

Nomenclature

- a = *hyperbole parameter*
- b = *hyperbole parameter*
- C = *camera lens focal distance*
- D = *Difference of Gaussian (DoG) function*
- d = *distance between hyperbolical mirror focus and camera focus*

Paper accepted August, 2010. Technical Editor: Glauco A. de P. Caurin

f = *distance from the image plane to the camera focus*
f^m = *SIFT feature*
G = *Gaussin function*
I = *image*
i_t = *incidente light ray*
K = *total number of landmarks*
L = *image*
m = *SIFT vector magnitude*
n_{ftr} = *landmark feature counter*
n_{obj} = *object counter*
n_t = *correspondence value between landmark and observation*
p = *probability distribution*
r_t = *reflected light ray*
s_t = *robot position in the x-y plane*
t = *time*
T = *absolute temperature, K*
u_t = *control vector*
x = *coordinate of robot position in plane, image coordinate*
x_c = *center coordinate of the omnidirectional image*
y = *coordinate of robot position in plane, image coordinate*
y_c = *center coordinate of the omnidirectional image*
z_t = *observation vector*

Greek Symbols

α = *reflected light ray in the hyperbolical mirror*
β = *SIFT vector orientation*
η = *Gaussian distribution*
μ = *mean*
$\boldsymbol{\rho}$ = *set of all landmarks*
ρ_k = *location of landmark, k*
σ = *variance*
θ = *robot orientation in plane*
φ = *incident light ray angle in mirror from a point in space*

Subscripts

c relative to center
ftr relative to a feature
k relative to a landmark
t relative to a moment in time
obj relative to an object

The problem of SLAM

The problem of localization and mapping in robotics can be described as a probabilistic Markov Chain, where the hidden variables are both the robot's localization and the map components. At a given time t we will denote the robot's position (assuming one-plane navigation) as $s_t = (x, y, \theta)$, composed of its coordinates in the $x - y$ plane and its orientation θ relatively to the x axis. This position evolves in time according to a probability distribution known as the *motion model*:

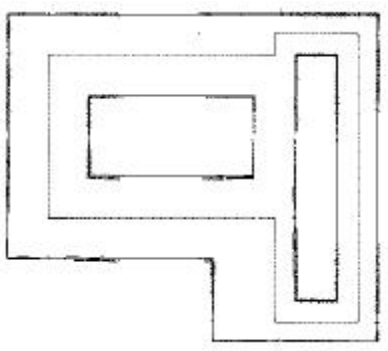
$$p(s_t \mid u_t, s_{t-1}) \tag{1}$$

where u_t is the control vector used for navigation. The robot's environment is composed of a set of K static landmarks with locations denoted as ρ_k. With its sensors the robot is capable of detecting these landmarks and measuring their positions relatively to itself (i.e. through range and bearing information). Each measurement is given by the observation vector z_t (we assume without loss of generality that the robot observes only one landmark at each instant) governed by a probability distribution known as the *measurement model*:

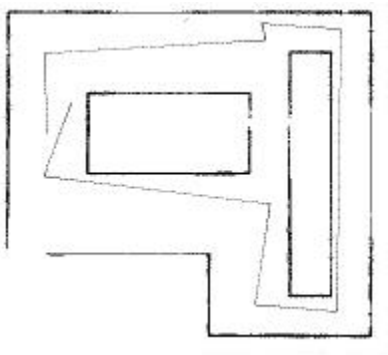
$$p(z_t \mid s_t, \rho, n_t) \tag{2}$$

where $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N)$ is the entire set of landmarks and n_t is the correspondence value that indicates which landmark ρ_n is observed by z_t. Most theoretical work on SLAM assumes that all correspondences $\mathbf{n} = (n_1, \ldots, n_t)$ are known, and thus the problem of SLAM becomes the one of determining the location of all landmarks $\boldsymbol{\rho}$ and robot poses s_t from measurements $z^t = (z_1, \ldots, z_t)$ and controls $u_t = (u_1, \ldots, u_t)$. So, we can write:

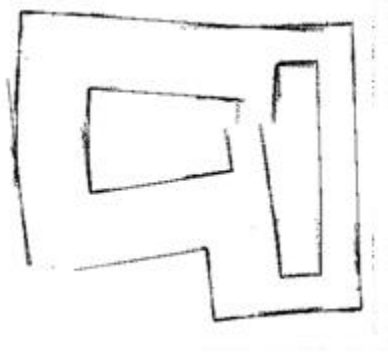
$$p(s_t, \boldsymbol{\rho} \mid \mathbf{z}, \mathbf{u}, \mathbf{n}) \tag{3}$$



(a) Exact map



(b) Localization estimates with odometry



(c) Mapping using odometric estimates

Figure 1. Influence of sensor errors in localization and mapping estimates (Guizilini et al., 2007).

In practical applications this is, however, not the case, as landmarks will never be truly unique in the environment, due to imprecision in the measurement or natural ambiguities. In this case,

we have to consider another probability distribution, Eq. (4), which indicates the probability of each measurement corresponding to each landmark. Most approaches use maximum likelihood algorithms, with thresholds that determine whether a measurement should be matched with a landmark already stored or considered as a new landmark.

$$p(n_t \mid \mathbf{z}, \mathbf{u}) \tag{4}$$

Feature extraction

A feature represents a piece of relevant information that can be obtained from the data collected. In computer vision, an image can provide both global features, where information contained in all image is used for feature extraction, and local features, where only a region of the image is used. Due to the necessity of detecting and recognizing particular objects in the image, local features are more commonly used in autonomous robotics to represent the environment. An extensive survey on local features is conducted by Tuytelaars and Mikolajczyk (2006), and methods for a better landmark selection in specific environments are shown by Shi and Tomasi (1994) and Olson (2002).

Although the method proposed in this paper can be used as a complement for any feature extraction method, we propose here the use of the SIFT algorithm as described by Lowe (Lowe, 2004) to obtain the initial feature set. The SIFT algorithm has become very popular in several robotics applications, as it can be seen in Se et al. (2001), Se et al. (2005), Ledwich and Williams (2004), and introduces several properties of invariance that are specially useful when extracting features directly from omnidirectional images, as it is the case in this work. Rotational invariance is important because detected objects can appear in any orientation depending on the angle between them and the robot, and so is scale invariance since resolution rapidly decreases in the outer ring of the image, changing the apparent size of observed objects. The high dimensionality of the SIFT descriptor provides some robustness regarding the deformation caused by the omnidirectional geometry, partially eliminating the need for rectification (Grassi and Okamoto, 2006).

The SIFT algorithm

The first stage of SIFT is composed of a search for local extrema over different scale spaces (ensuring scale invariance), constructed using a Difference of Gaussian (DoG) function $D(x, y, \sigma)$. This function (5) is computed from the difference of two nearby scaled images $L(x, y, \sigma)$, obtained by the convolution of the original image $I(x, y)$ with Gaussian kernels $G(x, y, \sigma)$ defined by their mean $\mu = (x, y)$ and variance σ, separated by a multiplicative factor k:

$$\begin{aligned} &D(x,y,\sigma) = \\ &= ((G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y)) \\ &= (L(x,y,k\sigma) - L(x,y,\sigma)) \end{aligned} \tag{5}$$

Pixels in any scale are reconsidered extrema if they represent a local maximum or minimum considering its neighbors in the same scale and in the ones directly above or below. These extrema are filtered according to two other criteria (contrast and ratio of main curvatures) for more stable matches and localized to sub-scale and sub pixel precision, as shown in Brown and Lowe (2002). A main orientation β (7) and magnitude m (6) are assigned to each remaining feature candidate using an orientation histogram obtained from pixel differences in the closest smoothed image $L(x, y, \sigma)$.

$$m(x,y) = \sqrt{\Delta x^2 + \Delta y^2} \tag{6}$$

$$\beta(x,y) = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) \tag{7}$$

where $\Delta x = L(x + 1, y) - L(x - 1, y)$ and $\Delta y = L(x, y + 1) - L(x, y - 1)$. This orientation histogram has usually 36 bins, covering 360° in intervals of 10°. Each point is added to the histogram weighted by its magnitude and by a circular Gaussian with \sigma variance that is 1.5 times the scale of the smoothed image used, to decrease the influence of distant portions of the image (Fig. 2). Additional feature candidates are generated where there are multiple dominant peaks, and dominant peaks are interpolated with their neighbors for a more accurate orientation assignment.

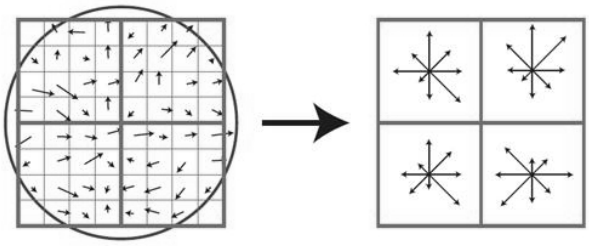


Figure 2. Example of SIFT descriptor with a total of 32 bins (Lowe 2004).

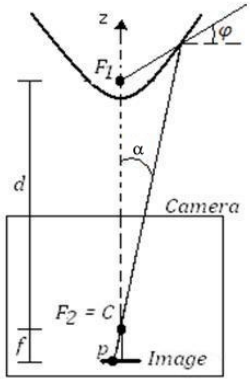
The gradient information used in the histogram is also used to create the feature descriptor. First, the gradient information is rotated and aligned with the feature's main orientation, creating relative measurements that ensure rotational invariance in further correspondences. This relative data is again weighted by a Gaussian as to decrease influence of distant portions of the image and is separated in sub-windows. Each sub-window has its own orientation histogram, composed of usually 8 bins, and each component of each sub window is added to the final descriptor. In the example of Fig. 2, the final descriptor would have 32 different values. To obtain a partial invariance to luminosity this descriptor is normalized, so global changes in intensity will not affect the result.

Omnidirectional vision

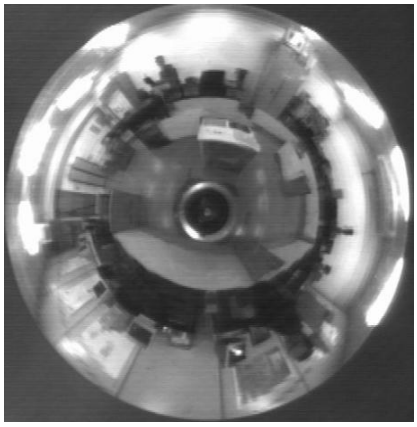
Omnidirectional vision sensors represent a family of visual sensors that are capable of obtaining simultaneously information regarding the entire environment around the camera (Zhu, 2001). Besides truly omnidirectional cameras (Nalwa, 1996), there are several other ways of obtaining this omnidirectional property, such as multiple cameras (Peleg and Ben-Erza, 1999) and special mirrors (Baker and Nayar, 1997).

Omnidirectional systems composed of special mirrors are usually more compact and without moving parts, thus being more suitable for applications in autonomous navigation. There are a number of possible mirror geometries, such as spherical, conical, parabolic or hyperbolic, each one with its own set of properties. Between these possible geometries the hyperbole has the property of single focus projection that allows the use of regular cameras in the omnidirectional vision system Svoboda and Pajdla (2002). Figure 3(a) presents a scheme of a hyperbolic mirror omnidirectional vision system, and Fig. 3(b) shows an example of omnidirectional image obtained using this configuration.

The camera is placed vertically and points to the mirror fixed above it, in a distance that ensures coincidence between the inferior hyperbole focus F_2 and the camera focus C. Light from the environment is reflected by the mirror and sent to the camera, converging to its focus and creating the omnidirectional image. The radial distance of the pixel p to the center of the image defines the angle α between the reflected light ray and the vertical axis.



(a) Scheme



(b) Image example

Figure 3. Scheme of a single-lobed hyperbolic mirror omnidirectional vision system.

The angle α also defines the angle φ between the incident light ray and the horizontal axis (Eq. (8), where a and b are the hyperbole parameters). As α increases, so does φ, and when $\varphi = 0$ that pixel will be observing the infinite.

$$\varphi = \tan^{-1}\left(\frac{d + f - \frac{ab}{\sqrt{a^2 - b^2\tan^2\theta}}}{\tan\alpha\,\frac{ab}{a^2 - b^2\tan^2\theta}}\right) \tag{8}$$

The value of φ as a function of α is shown in Fig. 4. When $\alpha = 0$ we have $\varphi = -\pi/2$, as expected, and when α increases so does φ, in a ratio that is proportional to the mirror curvature. This ratio defines radial resolution of pixels at each portion of image, which is high in the inner portions (small curvature) and low in the outer portions (high curvature). As a result, in these external areas even a small error in pixel coordinate estimation may result in a large error in feature position.

Another characteristic of omnidirectional systems is the deformation of objects, due to the projection of the mirror surface in the bi-dimensional surface of the image (Zhu, 2001). Most computational vision algorithms perform well in conventional geometries, so it is common that omnidirectional images are first rectified (Torii and Imiya, 2004) before utilized. However, the rectification process does not add information to the omnidirectional image, only rearranges it, and in the process incurs extra computational cost.

Therefore, we aim here for the direct extraction of information from omnidirectional images. In the next sections we describe the methods used for feature extraction and landmark selection, along with the matching and triangulation steps necessary to the use of this information in the SLAM algorithm previously presented.

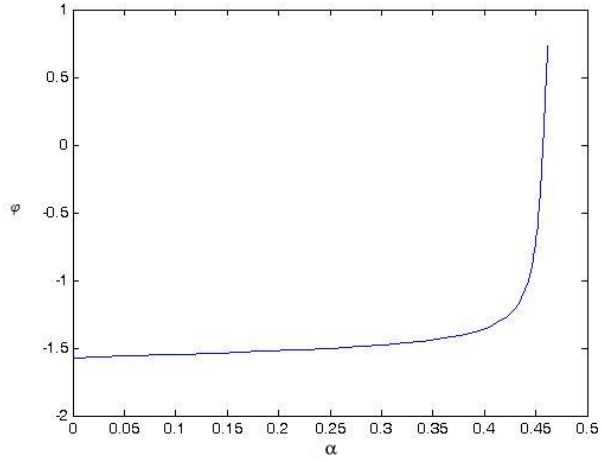
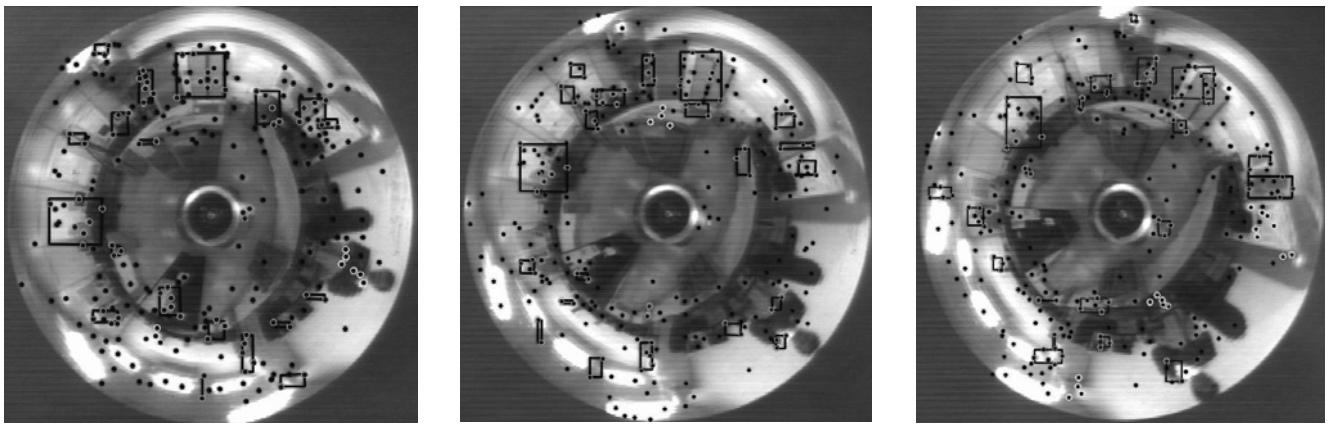


Figure 4. Decrease in resolution (Eq. (8)) in the radial axis of an omnidirectional image.

Selecting landmarks in omnidirectional images

The main drawback of SIFT features compared to other image descriptor is their high computational cost. A way of reducing computational cost in SIFT by removing its rotational invariance is presented by Ledwich and Williams (2004), but it assumes a conventional camera mounted parallel with the ground in a flat environment in order to create a stable point of view, which is not viable in omnidirectional images. The scale and translation invariances are removed for topological localization with omnidirectional images in Andreasson and Duckett (2004), because features should only be observed in the vicinity of the region where the image was obtained, but this compromises the robot's ability to recognize landmarks in different points of view. Lower descriptor dimensionality (Se et al., 2005) compromises object recognition in different distances from the robot due to image deformation. In resume, SIFT's invariance properties are important for generic feature extraction and landmark selection in different environments, especially in omnidirectional images, and therefore should not be eliminated.

Another limitation in SIFT features that increases computational cost is the volume of information generated, most of it redundant and non-representative of the environment, characterizing background structures and noise which are not matched between images that share a common view. Additionally, the local aspect of individual SIFT features generates data association problems in situations where there is object similarity. One possible solution to this problem is the use of feature database representing the objects that should be used as landmarks (Press and Austin, 2004), taking advantage of natural organization in certain kinds of environments. But this approach both limits the applicability of the solution in different environments, as it can only be used where these predetermined structures exist, and discards potentially useful information from other objects and structures not considered in the database.



(a) (b) (c)

Figure 5. Landmark selection in sequential frames (5 seconds apart) using the proposed method. Black dots indicate SIFT features and circled black dots indicate landmarks that were clustered into single clusters (rectangles). Darker circles are landmarks that were matched from precious frames and lighter circles are landmarks just added to the map.

We propose here the grouping of features from a single omnidirectional image into clusters based solely on image properties, and therefore can be determined equally in any kind of environment. Clusters without a minimum number of features are discarded and their features are not used, while others have their features promoted to landmarks and used by the robot to increase its knowledge of the environment. Position estimates of each landmark are still updated individually according to the SLAM algorithm used, but now they share the same unique cluster index, which is used in the correspondence step for more reliable matches, since the probability of one false match is higher than the probability of several false matches. This cluster index is also used to eliminate features that are consistently not matched in the environment, liberating space for new features. The result is fewer landmarks per image (lower computational costs), but these landmarks will be more representative of the environment and will be better distinguished (less data association problems).

Feature clustering

The two image properties constraints used in this paper were distance and intensity difference between pixels. We assume that features from the same object in the environment will have similar contrast in the image and be at a reasonable distance between each other. Each constraint has its own independent standard deviation σ_d and σ_c, and the probability of two features f_m and f_n be part of the same object is given by $p\,(f_m,\ f_n) = p_d\,(f_m,\ f_n)\ .\ p_c\,(f_m,\ f_n)$, where

$$p_d(f^m, f^n) = \eta\left(\sqrt{(f_x^m - f_x^n)^2 + (f_y^m - f_y^n)^2}, \sigma_d\right) \quad (9)$$

$$p_c(f^m, f^n) = \eta(f_c^m - f_c^n, \sigma_c) \quad (10)$$

and $\eta\,(\mu,\ \sigma)$ is a Gaussian distribution function. Each constraint is treated independently to decrease computational costs by applying each one separately. First, every two features of the image are compared according to pixel distance, and the ones with low probability are readily discarded. The ones within reasonable probability move to the second constraint, and if the final probability is high enough they are clustered as part of the same object. After all features in the image are compared, the ones that don't have a minimum of peers are discarded, while the other ones are promoted to landmarks and used by the robot as representative of the environment. Each landmark is treated independently, but shares the same cluster index that is used in the matching stage and also allows landmark elimination.

In omnidirectional images it is not correct to assume that the standard deviation σ_d is constant throughout the image as resolution varies in the radial axis (we assume here an omnidirectional vision system composed of an hyperbolic mirror and a conventional camera as shown in Grassi and Okamoto (2006)). This change of resolution affects the space represented by each pixel (Fig. 6), and in a different way for radial and angular distances, dividing σ_d into two distinct standard deviations, σ_r and σ_θ. The probability $p_d\,(f^m, f^n)$ of features f^m and f^n sharing the same object becomes:

$$p_d(f^m, f^n) = p_r(f^m, f^n) \cdot\ p_\theta(f^m, f^n) \quad (11)$$

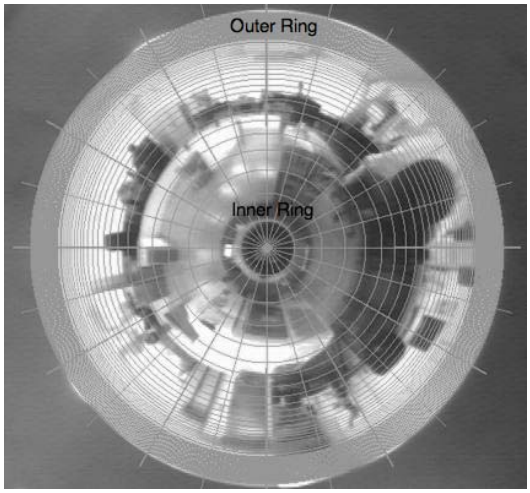
$$p_r(f^m, f^n) = \eta\left(\sqrt{(f_x^m - x_c)^2 + (f_y^m - y_c)^2} - \sqrt{(f_x^m - x_c)^2 + (f_y^n - y_c)^2}.\sigma_r\right) \quad (12)$$

$$p_\theta(f^m, f^n) = \eta\left(\tan^{-1}\left(\frac{f_y^m - y_c}{f_x^m - x_c}\right) - \tan^{-1}\left(\frac{f_y^n - y_c}{f_x^n - x_c}\right), \sigma_\theta\right) \quad (13)$$

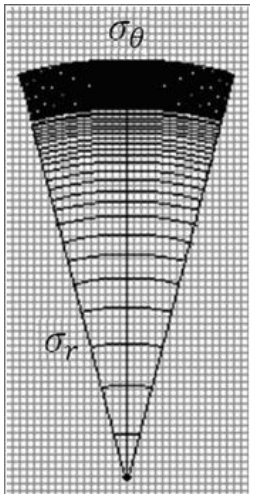
where x_c and y_c are the center coordinates of the omnidirectional image. Furthermore, the values of σ_r and σ_θ change differently according to the radial distance of the feature to the center of the image (see Fig. 6), as shown below:

- **Inner Ring**: σ_r increases and σ_θ decreases
- **Outer Ring**: σ_r decreases and σ_θ increases

In the inner ring of the image there are lesser pixels to represent angular intervals, so each pixel covers a larger angular distance (decreasing σ_θ). At the same time, since the mirror curvature is still small, radial intervals are represented by a higher number of pixels, increasing σ_r. In the outer ring of the image there are more pixels to represent each angular interval, which increases σ_θ, and each pixel has to cover a larger radial portion of the environment because of the higher mirror curvature, decreasing σ_r.



(a) Pixel grid



(b) Segment of pixel grid

Figure 6. Representation of radial resolution change throughout an omnidirectional image.

So, σ_r and σ_θ become functions $g_r(r)$ and $g_\theta(r)$ of the distance r between the features and the center of the omnidirectional image, determined by the system's parameters and geometry. Since two features will most likely have different distances, one straightforward way of determining an effective r is to find the arithmetic mean between each individual r. So

$$\sigma_r = g_r(r) \quad , \quad \sigma_\theta = g_\theta(r) \tag{14}$$

$$r = \left(\sqrt{(f_x^m - x_c)^2 + (f_y^n - y_c)^2} + \sqrt{(f_x^n - x_c)^2 + (f_y^n - y_c)^2}\right)/2 \tag{15}$$

Matching and Triangulation

The likelihood of matching between two features f_m and f_n is given by the Euclidean distance between its descriptors (Eq. (16)). The closer they are in the K-dimensional space, the higher is the matching probability. The matching set from one image is obtained minimizing the distance between their landmark set and a given particle's map (each map may have a different matching set).

$$d(f^m, f^n) = \sqrt{\sum_{i=0}^{K} (v_i^m - v_i^n)^2} \tag{16}$$

The high dimensionality of SIFT descriptors makes exhaustive search computationally intractable, so traditional approaches to this problem use probabilistic algorithms such as the Best Bin Fit (Beis and Lowe, 1997), which is capable of finding the optimal match within 95% of certainty, with a computational gain in two orders of magnitude.

During the matching step, each landmark stored on the robot's map is first compared directly to the features obtained from the omnidirectional image (without previous object clustering) using regular matching process, such as Best Bin Fit for SIFT. After this process, the number of successful matches in each cluster is calculated, using the index number of each landmark. If a minimum percentage of landmarks in each cluster are not matched all its matches are discarded, otherwise, they are assumed correct and their information is used to refine the robot's localization and mapping estimates.

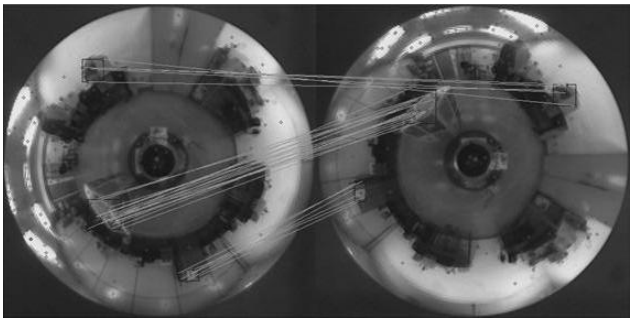


Figure 7. Matching in omnidirectinal images. Groups of features were selected in the left image and the line indicates matchings in the right image.

Every landmark has a counter n_{ftr} that indicates the amount of times it has been matched, and likewise every cluster has a counter n_{obj} to indicate the amount of time it has been successfully matched. If the ratio n_{obj}/n_{ftr} becomes too large it indicates that the cluster is being consistently matched without the need for that specific feature. This landmark can then be eliminated from the robot's map, decreasing the number of features representing that cluster. If this number is below a certain threshold new features can be incorporated as landmarks to the object using the same process presented earlier, and if no new features are available the whole cluster can be eliminated.

The position of each matched landmark in the environment is obtained through triangulation, using (Fig. 8) the position $(x, y)_1$ of the robot when the landmark was last observed and the position $(x, y)_2$ where the current omnidirectional image was obtained. The coordinates in each omnidirectional image of the feature that originated the landmark (p_1 and p_2) provide bearing information for the triangulation, and the mirror geometry allows the transformation from the reflected rays r_t to incident rays i_t. The position $(x, y, z)_n$ of the landmark in the environment is the point where i_1 and i_2 intersect. Since these measurements will be inevitably noisy, in real applications it is possible (and most likely) that i_1 and i_2 do not intersect, thus rendering the triangulation impossible. One simple solution, and the one used in this work, is to calculate the triangulation using solely the projection of both rays in the $x-y$

plane in the triangulation. The *z* coordinate is then calculated as the average position of both rays at the point where they intersect in the projection.

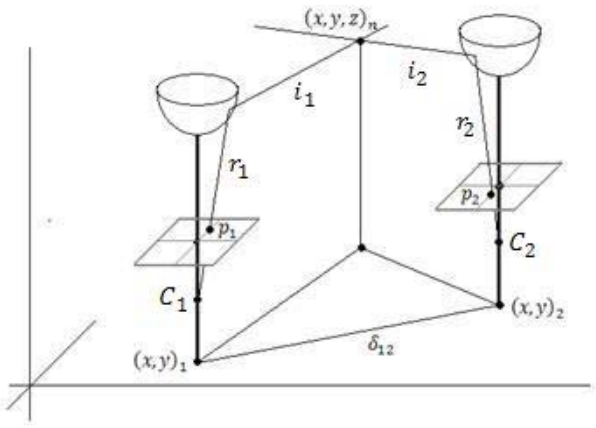


Figure 8. Ideal triangulation with a single omnidirectional vision system camera. The robot navigated between points $(x, y)_1$ and $(x, y)_2$ and observed the same landmark in the coordinates p_1 and p_2 of the images obtained at each instant.

Experimental Results

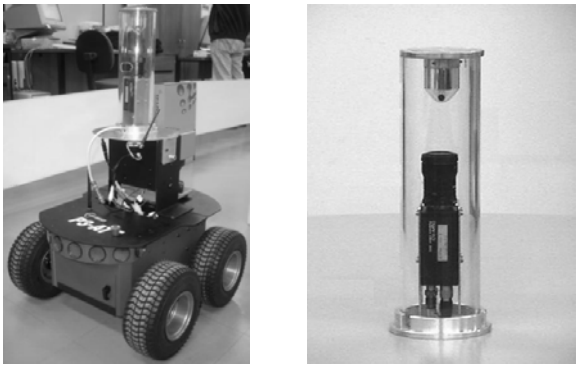
The landmark selection algorithm presented in this paper was tested in a real SLAM situation, using a Pioneer 3AT (Fig. 9(a)) equipped with an odometry system for incremental localization estimates, a laser scanner used solely to build a metric map of the environment, and an omnidirectional vision system composed of a hyperbolic mirror and a vertically placed camera (Fig. 9(b)) positioned on the rotation axis of the robot. The omnidirectional images collected were 640 x 480 gray scale and processed using a Pentium Core 2 Duo 2.0 GHz.

The SLAM algorithm used to incorporate the information obtained from the omnidirectional vision system was FastSLAM (Montemerlo, 2003), chosen due to its efficiency in dealing with large amounts of landmarks and data association problems. A particle filter (Rekleitis, 2003) is used to model the robot's localization uncertainty, and each particle also keeps an independent mapping hypothesis, which is updated using an Extended Kalman Filter (Welch and Bishop, 1995). Each landmark is updated individually according to the independency notion stated in Murphy (1999) and held true if the robot's position is assumed known, which is possible within each particle's hypothesis. Landmark position estimates were obtained through triangulation using matching information from two different instants.

We aim for an on-line solution to the problem of SLAM (with an update rate of 10 Hz), and the SIFT algorithm has a processing time far greater than this. So, we parallelized FastSLAM and SIFT, allowing the robot to navigate blindly while processing a collected omnidirectional image. During this stage its localization uncertainty increases, and when the processing is done the landmark information is incorporated to the estimate and the uncertainty decreases. Even though this update is based on past information, due to the parallelization, all particle position estimates can be tracked back and forwards over time (a characteristic of FastSLAM as a solution to the Full SLAM problem), and so the update can be easily propagated to the current instant of navigation.

An environment of corridors and obstacles (the robot could see above the walls, detecting landmarks outside its limits) was constructed (Fig. 10(a)) and the robot navigated through it in trajectories of roughly 70 m, with a maximum speed of 0.2 m/s. Initially the robot navigated without error correction, directly using odometry measurements to localize itself while building the metric mapping. Figure 10(b) shows the results of localization and metric mapping in this situation, where the errors accumulated during navigation can be clearly perceived through repetition and misalignment of structures and the inability of the robot to close the final loop and return to its starting position. The same path was then repeated using FastSLAM, and we tested the landmark selection method proposed by comparing it to the directly approach of using all features detected as landmarks. Figures 10(c) and 10(d) show the results of localization and metric mapping along with landmarks detected during navigation (gray circles plotted in the plane of navigation) using the direct and the proposed method, respectively. The structures in the environment were in no way modified prior to the navigation, and although there was no change in the environment during navigation, people could walk freely outside the established corridors. This behavior creates spurious landmarks that will not be matched in posterior images, providing a way of testing our method's landmark elimination process.

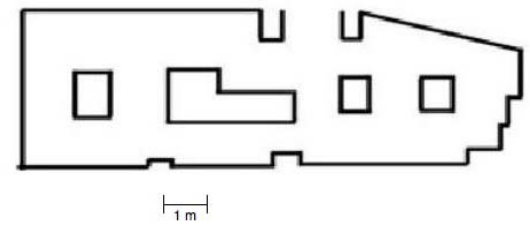
It is possible to see a substantially larger amount of landmarks in the direct approach compared to the landmark selection method proposed. These landmarks were also much more spread throughout the environment, while in the proposed method landmarks have a tendency of clustering in regions of high characterization according to SIFT. It is also possible to notice that in the direct approach there are a higher number of landmarks positioned over the robot's trajectory, indicating poor estimates.



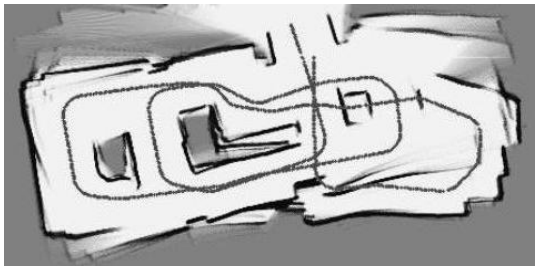
(a) Pioneer 3AT **(b) Omnidirectional vision system**

Figure 9. Equipment used in the experimental procedure.

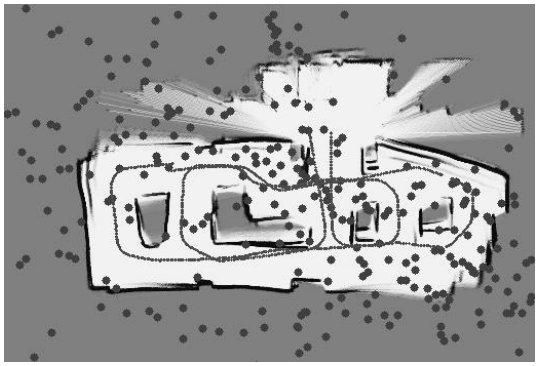
Also, the visual results of metric mapping show a better alignment and definition of corridors in the case where the proposed method was used, while some residual errors were maintained while using the direct approach. We attribute these residual errors to spurious landmarks and false matches caused by the large amount of data incorporated at each iteration. A larger amount of data also implies in a larger computational cost, which is reflected in the amount of time between image acquisition and information incorporation, when the robot navigates blindly in the environment and accumulates localization errors. Table 1 compares values regarding the use of each approach for landmark selection.



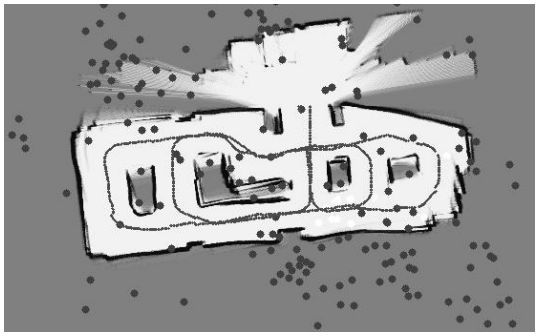
(a) Exact map



(b) Results without SLAM



(c) SLAM with individual landmarks



(d) SLAM with proposed method

Figure 10. Results obtained in the experimental tests. The total area is approximately 4 m by 10 m. The total path length is approximately 70 m.

In fact, we see that the proposed method can process an omnidirectional image, obtaining the final landmark set in approximately 40% of the time necessary when using the features directly as landmarks. During navigation the proposed method was capable of analyzing 251 images, while the direct approach could process only 104, indicating a much higher period of blind navigation and a longer distance of navigation between matches, compromising landmark recognition and increasing error accumulation between each update stage of FastSLAM.

Each image provided a smaller number of landmarks in the proposed method, due to the features discarded as not part of any object. Logically, the amount of matches was also smaller, but proportionally it was able to match a higher amount of landmarks (53.56% against 19.46% on the direct approach). This indicates a higher percentage of information used over information obtained, characterizing higher efficiency in landmark selection. There are no statistics for number of landmarks correctly matched, since the features were obtained automatically, but the metric mapping results shown earlier indicate a better matching in the proposed method due to elimination of residual errors. Again, no quantitative statistics are provided for the localization estimates in different situations, because there is no ground-truth data for comparison (the navigation took place indoors, where there was no GPS signal).

Table 1: Comparative results using the direct approach and the proposed method.

	Individual Landmarks	Proposed Method	%
Features per frame	299.31	297.31	99.3
Frames processed	104.81	251.34	239.8
Processing time (s)	4.91	1.95	39.7
Total of landmarks per frame	299.31	78.69	26.3
Landmarks matched per frame	58.24	42.15	72,4

Conclusion

We presented here a method of landmark selection and clustering for on-line SLAM in omnidirectional images that does not require any prior knowledge of the environment, and thus can be in theory used equally in any situation. We use image properties such as pixel distance and contrast to create constraints that cluster features that are used by the SLAM algorithm as landmarks. This approach decreases computational cost by eliminating non-relevant landmarks and increases reliability of matches by corresponding groups of landmarks instead of individually. Results show improvement both in landmark selection efficiency and in quality of localization and mapping estimates when compared to a common approach of using all features and landmarks. The restraints used to cluster features, along with the threshold for landmark promotion, may be changed as to increase performance in different environments and with different camera geometries. Future work will include larger loop-closures, which should not pose as a big challenge since landmarks are uniquely identified by its features, without any spatial constraint. Also, other sensors will be included, such as laser and IMU, as a way to improve results in situations where visual information is not enough to solve the SLAM problem.

Acknowledgements

We would like to thank the financial supporting agency FAPESP for this work, under grants #07/05293-2 and #07/07104-5.

References

Akihiko, T. and Imiya, A., 2004, “A panoramic image transform of omnidirectional images using discrete geometry techniques”, 2nd International Symposium on 3D Data Processing Visualization and Transmission (DPVT).

Andreasson, H. and Duckett, T., 2004, "Topological localization for mobile robots using omnidirectional vision and local features", 5th Symposium on Intelligent Autonomous Vehicles.

Bailey, T., 2002, "Mobile Robot Localization and Mapping in Extensive Outdoor Environments", PhD thesis, University of Sydney.

Baker, S. and Nayar, S., 1998, "A theory of catadioptric image formation", International Conference for Computer Vision, pp. 35-42.

Beis, J. and Lowe, D., 1997, "Shape indexing using approximate nearest-neighbour search in high dimensional spaces", Conference on Computer Vision and Pattern Recognition, pp. 1000-1006.

Brown, M. and Lowe, D., 2002, "Invariant features from interest point groups", British Machine Vision Conference, pp. 65-665.

Csorba, M., 1997, "Simultaneous Localization and Map Builing", PhD thesis, University of Oxford, Robotics Research Group.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S., 1999, "Monte carlo localization for mobile robots", International Conference on Robotics and Automation.

Fitzgibbons, T., 2004, "Visual-Based Simultaneous Localization and Mapping", PhD thesis, University of Sydney.

Gaspar, J., 2003, "Omnidirectional Vision for Mobile Robot Navigation", PhD thesis, Universidade Técnica de Lisboa, Instituto Superior Tecnico.

Grassi, V.J. and Okamoto, J.J., 2006, "Development of an omnidirectional vision system", *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vo. 28, pp. 58-68.

Guizilini, V., Okamoto, J.J., Grassi, V., and Correa, F, 2007, "Implementação do dp-slam em tempo real para robôs móveis usando sensores esparsos", SBAI.

Ledwich, L. and Williams, S., 2004, "Reduced sift features for image retrieval and indoor localization", Australian Conference on Robotics and Automation.

Leonard, J. and Feder, H., 1999, "A computationally efficient method for large-scale concurrent mapping and localization", International Symposium of Robotics Research.

Leonard, J., Rickoski, R., Newman, P., and Bosse, M., 2002, "Mapping partially observable features from multiple uncertain vantage points", *International Journal of Robotics Research*, 21(10-11):943-975.

Lowe, D., 2004, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, Vo. 20, pp. 91-110.

Montemerlo, M., 2003, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association", PhD thesis, Robotics Institute, Carnegie Mellon University.

Moutarlier, P. and Chatila, R., 1989, "Stochastic multisensory data fusion for mobile robot localization and environment modeling", 5th Simposium on Robotics Research.

Murphy, K., 1999, "Bayesian map learning in dynamic environments", Neural Information Processing Systems.

Nalwa, V., 1996, "A true omnidirectional viewer", Technical Report, Bell Lab. Olson, C.F., 2002, "Selecting landmarks for localization in natural terrain", *Autonomous Robots*, Vo. 12, pp. 201-210.

Peleg, S. and Ben-Erza, M., 1999, "Stereo panorama with a single camera", Computer Vision and Pattern Recognition, pp. 395-401.

Press, P. and Austin, D., 2004, "Approaches to pole detection using ranged laser data", Proceedings of Australasian Conference on Robotics and Automation.

Rekleitis, I., 2003, "A particle filter tutorial for mobile robot localization", International Conference on Robotics and Automation.

Se, S., Lowe, D., and Little, J., 2001, "Vision-based mobile robot localization and mapping using scale-invariant features", International Conference on Robotics and Automation, pp. 2051-2058.

Se, S., Lowe, D., and Little, J., 2005, "Vision-based mobile robot localization and mapping for mobile robots", IEEE Transactions on Robotics, Vol. 21.

Shi, J. and Tomasi, C., 1994, "Good features to track", Proceedings of IEEE, Conference on Computer Vision and Pattern Recognition.

Smith, R., Self, M., and Cheeseman, P., 1990, Estimating uncertain spatial relationships in robotics. Autonomous Robot Vehicles, pages 167-193.

Svoboda, T. and Pajdla, T., 2002, "Epipolar geometry for central catadioptric cameras", *International Journal of Computer Vision*, Vol. 49.

Thrun, S., 2002, "Robotic mapping: A survey", Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann.

Thrun, S., Burgard, W., and Fox, D., 2005, "Probabilistic Robotics. MIT Press".

Thrun, S., Fox, D., and Burgard, W., 1998, "A probabilistic approach to concurrent mapping and localization for mobile robots", Machine Learning, Vol. 31.

Tuytelaars, T. and Mikolajczyk, K., 2006, "A survey on local invariant features".

Welch, G. and Bishop, G., 1995, "An introduction to the kalman filter", technical report, University of North Carolina, Department of Computer Science.

Zhu, Z., 2001, "Omnidirectional stereo vision", Proceedings of IEEE, 10th International Conference on Advanced Robotics.