Selectively Using Landmarks in Online SLAM with Omnidirectional Vision

Vitor Campanholo Guizilini, Jun Okamoto Jr.

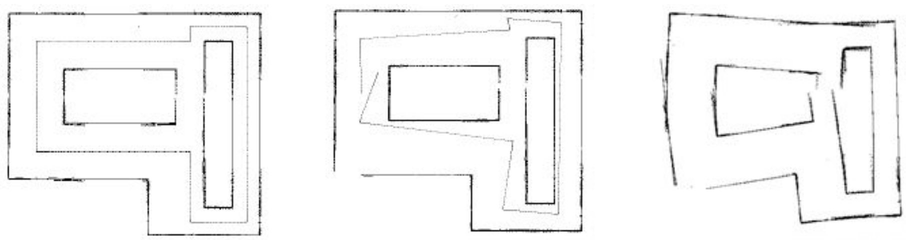
Advanced Perception Laboratory
Department of Mechatronics and Mechanical Engineering
Escola Politécnica da Universidade de São Paulo, SP, Brazil
vitor.guizilini@poli.usp.br, jokamoto@usp.br

Abstract. 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 the other by eliminating errors inherent to the sensors. One common solution consists of establishing landmarks in the environment that 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 in omnidirectional images for online 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 propense 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.

1 Introduction

A solution to the problem of SLAM 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 [1] have successfully dealt with both problems individually, such as mapping given the robot's exact position at each instant [2] or localization given a precise map of the environment [3]. 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 [4–7] but still lacks a closed, efficient and truly generic solution. The classic approach to the problem of SLAM, first described in [9] and implemented

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205



(a) Exact Map (b) Localization estimates using odometry (c) Mapping using odometric estimates

Fig. 1. Influence of sensor errors in localization and mapping estimates [8]

in [10], is based on detection and recognition of landmarks in the environment that can be used as reference points to eliminate odometric 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 [11, 6] and data association problems [12, 13].

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 [14], vision sensors are attractive equipments 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 [7, 15]. Among visual sensors, the omnidirectional vision [16] introduces several properties that are very desirable in most navigational tasks [17], including in the SLAM problem discussed above [18]. 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 costs 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 landmark extraction that is based on clustering features

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205

directly from omnidirectional images, without any prior knowledge of the environment and thus that can in theory be applied in any situation. We start by briefly describing the problem of SLAM and the use of landmarks to ensure localization precision. After that the proposed method of landmark selection is described, along with modifications in the matching step and landmark management. Finally, we show results obtained in a real SLAM situation that indicate a significative gain in quality and efficiency over a common approach of landmark selection.

2 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 by its coordinates in the $x - y$ plane and its orientation θ relatively to the x axis. This position evolves in time according to a probabilistic distribution known as the *motion model*:

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

Where u_t is the control vector used for navigation. The robot's environment is composed by a set of K static landmarks with locations denoted as θ_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 observers only one landmark at each instant) governed by a probabilistic distribution known as the *measurement model*:

$$p(z_t|s_t, \theta, n_t) \tag{2}$$

Where $\theta = (\theta_1, ..., \theta_N)$ is the entire set of landmarks and n_t is the correspondence value that indicates which landmark θ_n is observed by z_t. Most theoretical work on SLAM assumes that all correspondences $n^t = (n_1, ..., n_t)$ are known correspondence is known, and thus the problem of SLAM becomes one of determining the location of all landmarks θ and robot poses s_t from measurements $z^t = (z_1, ..., z_t)$ and controls $u^t = (u_1, ..., u_t)$. In other words:

$$p(s^t, \theta|z^t, u^t, n^t) \tag{3}$$

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 probabilistic distribution, which indicates the probability of each measurement corresponding to each landmark. Most approaches use maximum likelihood algorithms, with thresholds that determine if measurement should be matched with a landmark already stored or considered as a new landmark.

$$p(n_t|z^t, u^t) \tag{4}$$

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205

3 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 in [19], and methods for a better landmark selection in specific environments are shown in [20] and [21].

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 [22] to obtain the initial feature set. The SIFT algorithm has become very popular in several robotics applications, as it can be seen in [18, 23, 24], and introduces several properties of invariance that are especially useful when extracting features directly from omnidirectional images, as it is the case in this paper. Rotational invariance is important because objects detected 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 [25].

The first stage of SIFT is composed by 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 (Eq. 5) is computed from the difference of two nearby scaled images $L(x, y, \sigma)$, obtained convoluting the original image $I(x, y)$ with Gaussian kernels $G(x, y, \sigma)$ separated by a 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 considered 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 subscale and sub pixel precision, as shown in [26]. A main orientation (Eq. 6) and magnitude (Eq. 7) are assigned to each remaining feature candidate using an orientation histogram obtained from pixel differences in the closest smoothed image $L(x, y, \sigma)$. Each pixel orientation is added to the histogram weighted by its magnitude and by a circular Gaussian to decrease the influence of distant portions of the image.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{6}$$

$$\theta(x, y) = tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{(L(x+1, y) - L(x-1, y)}\right) \tag{7}$$

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205

This gradient information is then divided into sub-windows, each one with its own orientation histogram, that is obtained relatively to the main orientation to ensure rotational invariance. The magnitude value of each orientation in each histogram is added to the final descriptor, and to obtain a partial invariance to luminosity this descriptor is normalized, so global changes in intensity will not affect the result.

4 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 in [24], 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 unviable in omnidirectional images. The scale and translation invariances are removed for topological localization with omnidirectional images in [15] 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 dimensionalities [23] compromise 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 increase computational cost is the volume of information generated, most of it redundant and non-representative of the environment, characterizing background structures and noise that is 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 [14], 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.

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 are promoted to landmarks and used by the robot to increase its knowledge of the environment. Position estimates of each landmark are updated individually according to the SLAM algorithm used, but they share the same object 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 object index is also used to eliminate features that are consistently not matched in the

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205

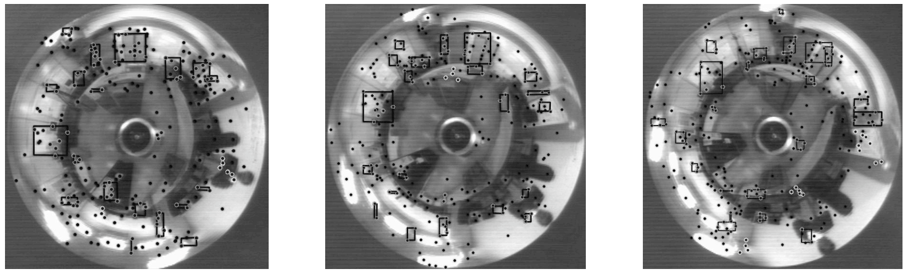


Fig. 2. Landmark selection in sequential frames (5 seconds apart) using the proposed method. Black dots indicate SIFT features, and circled black dots indicate landmarks that were clustered into single objects (rectangles). Darker circles are landmarks that were matched from previous frames and lighter circles are landmarks just added to the map.

environment, liberating space for new features. The result is less landmarks per image (lower computational costs), but these landmarks will be more representative of the environment and will be more distinguished (less data association problems).

4.1 Feature Clustering

The two image properties restraints 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 color contrast in the image and be at a reasonable distance between each other. Each restraint 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:

$$p_d(f^m, f^n) = \eta\left(\sqrt{(f_x^m - f_x^n)^2 + (f_y^m - f_y^n)^2}, \sigma_d\right) \tag{8}$$

$$p_c(f^m, f^n) = \eta\left(f_c^m - f_c^n, \sigma_c\right) \tag{9}$$

And $\eta(\mu, \sigma)$ is a Gaussian distribution function. Each restraint 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 already discarded. The ones within reasonable probability move to the second restraint, 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 object index that is used in the matching stage and also allows landmark elimination.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205

4.2 Varying σ_d

The previous section assumes that the standard deviation σ_d is constant throughout the image, but in omnidirectional images this is not the case, 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 [25]). This change of resolution affects the space represented by each pixel, 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:

$$p_d(f^m, f^n) = p_r(f^m, f^n).p_\theta(f^m, f^n) \tag{10}$$

$$p_r(f^m, f^n) = \eta\left(\sqrt{(f_x^m - x_c)^2 + (f_y^m - y_c)^2} - \sqrt{(f_x^n - x_c)^2 + (f_y^n - y_c)^2}, \sigma_r\right) \tag{11}$$

$$p_\theta(f^m, f^n) = \eta\left(tan^{-1}\left(\frac{f_y^m - y_c}{f_x^m - x_c}\right) - tan^{-1}\left(\frac{f_y^n - y_c}{f_x^n - x_c}\right), \sigma_\theta\right) \tag{12}$$

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, as shown below:

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

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

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:

$$\sigma_r = g_r(r) \quad , \quad \sigma_\theta = g_\theta(r) \tag{13}$$

$$r = \left(\sqrt{(f_x^m - x_c)^2 + (f_y^m - y_c)^2} + \sqrt{(f_x^n - x_c)^2 + (f_y^n - y_c)^2}\right)/2 \tag{14}$$

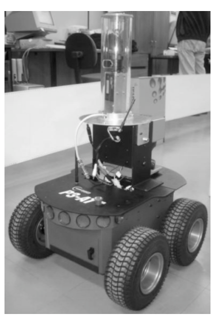
4.3 Matching

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 object is calculated, using the index number of each landmark. If a minimum percentage of landmarks in each object 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.

Every landmark has a counter n_{ftr} that indicates the amount of times it has been matched, and likewise every object 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 object 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 object. If this number is below a certain threshold new features can be incorporated as landmarks to the object using the clustering process presented earlier, and if no new features are available the whole object can be eliminated.

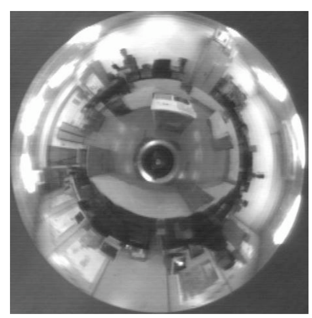
5 Experimental Results

The landmark selection algorithm presented in this paper was tested in a real SLAM situation, using a Pioneer 3AT (Fig. 3a) 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. 3b) positioned on the rotation axis of the robot. The omnidirectional images (Fig 3c) collected were 640x480 grayscale and processed using a Pentium Core 2 Duo 2.0 GHz.



(a) Pioneer 3AT

(b) Vision System



(c) Omnidirectional Image

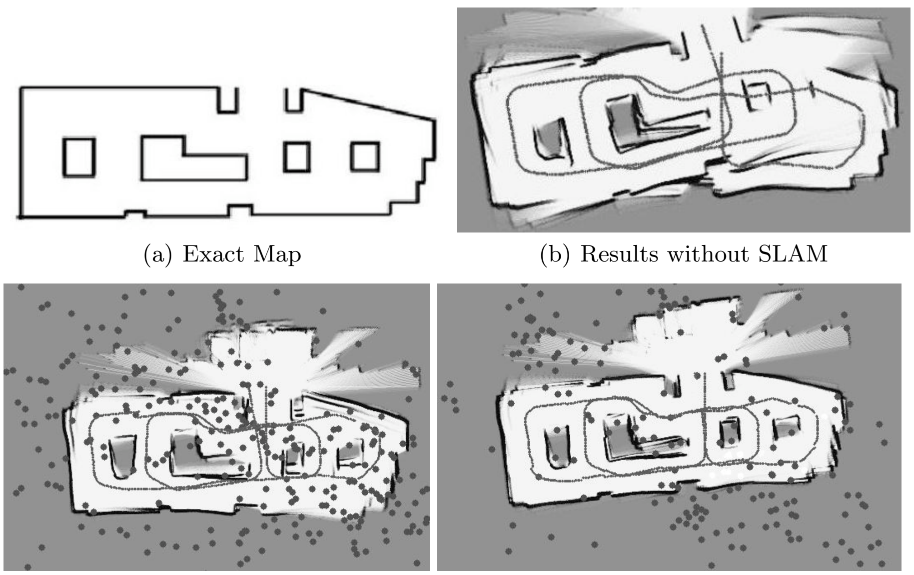
Fig. 3. Equipment used in the experimental tests.

The SLAM algorithm used to incorporate the information obtained from the omnidirectional vision system was FastSLAM [6], chosen due to its efficiency in dealing with large amounts of landmarks and data association problems. A particle filter [27] is used to model the robot's localization uncertainty, and each particle also keeps an independent mapping hypothesis, which is updated using

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205

an Extended Kalman Filter [28]. Each landmark is updated individually according to the independency notion stated in [29] 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 online 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 an omnidirectional image collected. During this stage its localization uncertainty increases, and when the processing is done the landmark information is incorporated to the estimates and the uncertainty decreases.



(a) Exact Map

(b) Results without SLAM

(c) SLAM with Individual Landmarks

(d) SLAM with Proposed Method

Fig. 4. Results obtained in the experimental tests.

An environment of corridors and obstacles (the robot could see above the walls, detecting landmarks outside its limits) was constructed (Fig. 4a) and the robot navigated through it in trajectories of roughly 70m, 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. Fig. 4b 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. Figs. 4c and 4d 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 is a higher number of landmarks positioned over the robot's trajectory, indicating poor estimates.

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.

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

	Individual Landmarks	**Proposed Method**	**%**
Features per frame	299.31	297.32	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

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 methods 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.

6 Conclusion

We presented here a method of landmark selection for online 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 restraints 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 as 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 may be changed as to increase performance in different environments and with different camera geometries.

7 Acknowledgements

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

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
2. Thrun, S.: Robotic mapping: A survey. In: Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann (2002)
3. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: International Conference on Robotics and Automation. (May 1999)
4. Csorba, M.: Simultaneous Localization and Map Builing. PhD thesis, University of Oxford, Robotics Research Group (1997)
5. Bailey, T.: Mobile Robot Localization and Mapping in Extensive Outdoor Environments. PhD thesis, University of Sydney (2002)
6. Montemerlo, M.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association. PhD thesis, Robotics Institute, Carnegie Mellon University (July 2003)

7. Fitzgibbons, T.: Visual-Based Simultaneous Localization and Mapping. PhD thesis, University of Sydney (2004)
8. Guizilini, V., Okamoto, J.J., Grassi, V., Correa, F.: Estudo e implementao do dp-slam. In: SBAI. (2007)
9. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. Autonomous Robot Vehicles (1990) 167–193
10. Moutarlier, P., Chatila, R.: Stochastic multisensory data fusion for mobile robot localization and environment modeling. In: 5th Simposium on Robotics Research. (1989)
11. Leonard, J., Feder, H.: A computationally efficient method for large-scale concurrent mapping and localization. In: International Symposium of Robotics Research. (1999)
12. Thrun, S., Fox, D., Burgard, W.: A probabilistic approach to concurrent mapping and localization for mobile robots. In: Machine Learning. Volume 31. (1998)
13. Leonard, J., Rickoski, R., Newman, P., Bosse, M.: Mapping partially observable features from multiple uncertain vantage points. International Journal of Robotics Research **21**(10-11) (2002) 943–975
14. Press, P., Austin, D.: Approaches to pole detection using ranged laser data. In: Proceedings of Australasian Conference on Robotics and Automation. (2004)
15. Andreasson, H., Duckett, T.: Topological localization for mobile robots using omnidirectional vision and local features. In: 5th Symposium on Intelligent Autonomous Vehicles. (July 2004)
16. Zhu, Z.: Omnidirectional stereo vision. In: Proceedings of IEEE, 10th International Conference on Advanced Robotics. (2001)
17. Gaspar, J.: Omnidirectional Vision for Mobile Robot Navigation. PhD thesis, Universidade Tecnica de Lisboa, Instituto Superior Tecnico (2003)
18. Se, S., Lowe, D., Little, J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: International Conference on Robotics and Automation. (May 2001) 2051–2058
19. Tuytelaars, T., Mikolajczyk, K.: A survey on local invariant features. (May 2006)
20. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of IEEE, Conference on Computer Vision and Pattern Recognition. (Jun 1994)
21. Olson, C.F.: Selecting landmarks for localization in natural terrain. In: Autonomous Robots. Volume 12. (2002) 201–210
22. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision. Volume 20. (2004) 91–110
23. Se, S., Lowe, D., Little, J.: Vision-based mobile robot localization and mapping for mobile robots. In: IEEE Transactions on Robotics. Volume 21. (June 2005)
24. Ledwich, L., Williams, S.: Reduced sift features for image retrieval and indoor localization. In: Australian Conference on Robotics and Automation. (2004)
25. Grassi, V.J., Okamoto, J.J.: Development of an omnidirectional vision system. In: Journal of the Brazilian Society of Mechanical Sciences and Engineering. Volume 28. (2006) 58–68
26. Brown, M., Lowe, D.: Invariant features from interest point groups. In: British Machine Vision Conference. (2002) 656–665
27. Rekleitis, I.: A particle filter tutorial for mobile robot localization. In: International Conference on Robotics and Automation. (2003)
28. Welch, G., Bishop, G.: An introduction to the kalman filter. Technical report, University of North Carolina, Department of Computer Science (1995)
29. Murphy, K.: Bayesian map learning in dynamic environments. In: Neural Information Processing Systems. (1999)

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 194-205