Hierarchical Image-based Localisation for Mobile Robots with Monte-Carlo Localisation

Emanuele Menegatti, Mauro Zoccarato
Enrico Pagello
Intelligent Autonomous Systems Laboratory
Dept. of Information Engineering (DEI)
Fac. of Engineering, The Univ. of Padua
Via Gradenigo 6/a, I-35131 Padova, Italy

Hiroshi Ishiguro

Department of Adaptive Machine Systems

Osaka University
Suita, Osaka, 565-0871 Japan

Abstract

This paper extends our previous works on image-based localisation for mobile robot. The image-based localisation consists in matching the current view experienced by the robot with the reference views stored in the visual memory of the robot. The original idea was to use the Fourier components as signatures for the omnidirectional images acquired by the robot. The extensions proposed in this paper are: the possibility to have a localisation with different accuracy while the robot navigates (called hierarchical localisation) and the introduction of a Monte-Carlo localisation technique to increase the robustness of the system in environments where the image-based localisation can be mislead. Experiments demonstrated the feasibility of the hierarchical localisation and the robustness of the implementation of our Monte-Carlo localisation.

1. Introduction

Localisation is a basic task for a robot that moves in an environment. Usually, the robot is provided with a geometrical map of the environment and it will use some kind of sensors to locate itself in this map. The sensors' reading is noisy, but with a good management of the uncertainty and with reliable sensors like laser range scanner, good results have been achieved [14] [3].

If the map of the environment is not available, building the geometrical map of the environment can be time consuming. A different approach that does not use a map is the **image-based localisation** [1, 5, 9, 11, 13, 7]. There is experimental evidence that the image based localisation is used by very simple animals like bees and ants [6] and it is a every-day experience that is used by humans as well. For instance, it happened to everyone to get lost in an unfamiliar place and to be able to recover its position recognising a *view* previously experienced.

Figure 1: An omnidirectional image taken at a reference location.

Figure 2: The panoramic cylinder created by the omnidirectional image of Fig. 1.

In the image based localisation approach, the agent is provided with a set of *views* of the environment taken at several locations in the environment. These locations are called **reference locations** because the robot will refer to them to locate itself in the environment. The corresponding images are called **reference images**. When the robot moves, it can compare the current view with the reference images stored in its visual memory. When the robot finds which one of the reference images is more similar to the current view, it

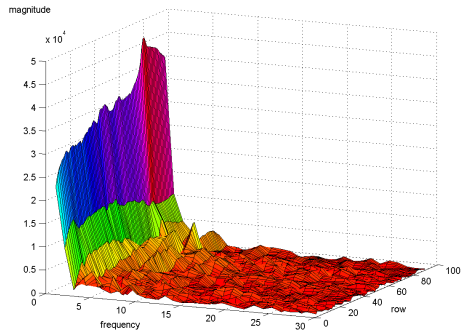


Figure 3: The power spectrum of the Fourier transform of the image in Fig. 2. Note that only the first 30 components are shown and components after the 15th have very small values and so can be neglected in the calculation of the similarity function.

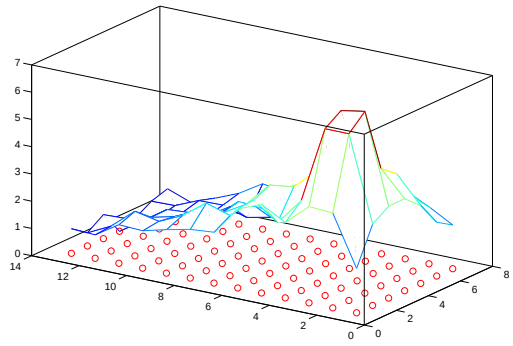


Figure 4: The values of the similarity functions calculated at every reference point for the current image. The empty circles on the XY plane represent the reference images. The full circle represents the actual position of the current image. The hight of the surface at every reference location is proportional to the degree of similarity of the reference image with the current image.

can infer its position in the environment. In the image based localisation, the problem of finding the position of the robot in the environment is reduced to the problem of finding the best match for the current image among the reference images. Most of the cited works on image-based localisation just stop here. In this paper, we propose to extend the image based localisation with two additional features: the hierarchical localisation and the statistical approach to the robot's localisation. Let us see the hierarchical localisation first.

When a robot navigates in a real world environment, it does not need to know its position with the highest accuracy at any time. The actual accuracy needed depends on the environment's structure and on the current action the robot is performing. If the robot is crossing a wide open space, it can tolerate a quite large uncertainty in its position. While if the robot enters a door, the accuracy on the localisation must be high. This is similar to the behaviour we experience walking in a street of an unknown town using a map. When we are following the high-street we do not need to know our exact position on the map, but when we have to take a detour or to enter a building we need to shrink down the uncertainty in our position, maybe looking for additional environmental clues. We called this process **hierarchical localisation**. In the body of the paper we will see how it is possible to calculate the robot's localisation with different accuracies using the Fourier components of the panoramic image.

The image based localisation approach has a flaw. It does not work in environments with periodical structures, i.e. in case of a structure that repeats several times in the environment (for instance: a corridor with a set of doors equally spaced with the same appearance). In this case, the appearance of the world is the same in different places, so the current view will match not only to the corresponding reference image but also all the reference images that are similar to the current one. This is a case of *perceptual aliasing*, i.e. the reading of the sensor is the same at different locations and the use of the vision sensor alone is not able to discriminate between the different places that looks the same. This is similar to what happens to humans when they get lost in a large building, because every place looks the same. To understand where they are humans need additional clues like asking directions or reading signboards. A robot could use additional sensors able to discriminate between the two points, e.g. GPS sensors or other non-vision sensors. But what about if these additional sensors are not available? The robot needs to manage situations, maybe transitory situations, in which it has evidence of being located at two distinct points at the same time. It needs a tool to manage the uncertainty about its position. The solution we adopted here is to use a Monte-Carlo Localisation process to manage this uncertainty [14]. This technique is able to manage multi-modal distribution of probability. Therefore, also in situations where the current image matches more than one reference image, the robot can correctly estimate its position.

In the next section we will outline the algorithm used to assess the similarity of two images. In Section 3, the problem of the hierarchical localisation is introduced and the solution we propose is detailed. In Section 4, we sketch our implementation of the statistical approach to handling the belief about the robot's position and we present the sucessful experiments performed in a real-world environment. In the last section, conclusions are drawn.

Image	Memory Required (in bit)	Memory Required
omnidirectional	$640 \times 480 \times 24$	7.3 Mbit
panoramic cylinder	$512 \times 80 \times 24$	980 Kbit
Fourier signature	$80 \times 15 \times 2 \times 8$	19 Kbit

Table 1: The different memory requirements illustrating the impressive memory saving introduced by the Fourier signature.

2 Image Matching

In the image based localisation approach, the main problem is how to store and to compare the reference images, which for a wide environment can be a large number.

In this paper we have fully developed a method we proposed in a previous work [9]. The robot is equipped with an omnidirectional camera [8, 12] and takes a set of omnidirectional images at the reference locations, then it compares the current omnidirectional image with the reference images. In order to store and match efficiently a large number of images, we transform each omnidirectional view into a compact representation. From the omnidirectional image we create a **panoramic cylinder**, i.e. a new image obtained unwarping the original omnidirectional image, as depicted in Fig. 2. The panoramic cylinder is expanded row by row into Fourier series. The agent memorises each view by storing the Fourier coefficients of the low frequency components of the panoramic cylinder, depicted in Fig. 2. We called the set of the stored coefficients **Fourier signature** of the omnidirectional image. This drastically reduces the amount of memory required to store a view at a reference location. With this approach also the matching of the current view against the visual memory is computationally inexpensive. For more details on this procedure, please refer to [9].

Let us highlight the advantages of our approach with respect to the work of others authors. The use of the omnidirectional camera with respect to the use of a perspective camera reduces the number of images required to fully describe the environment. In fact, if a perspective camera is used, the view of the environment from a certain location changes with the direction of gaze. A solution can be to constrain the movements of the robot in order to have the perspective camera pointing always at the same direction [5], but this strongly limits the motion of the robot. An alternative approach can be to extract from the perspective images some features that reduce the amount of required memory while retaining a unambiguous description of the image. A good example of this is reported in [15], where 936 images where store in less than 4MB. Nonetheless, collecting such a big number of images is tedious and time consuming.

Several authors, exploited omnidirectional cameras in image based localisation. In order to reduce the amount of memory required, they extracted a set of eigenimages from the set of reference images and projected the images into eigenspaces. The drawback of such systems is that they need to preprocess the images they created from the omnidirectional image in order to obtain the rotational invariance as in [1, 10, 7] or to constrain the heading of the sensor as in [11].

On the contrary, our technique, which uses the Fourier transform of the panoramic cylinder (i.e. the *Fourier Signature*, is the natural representation for implementing a *rotational invariance*, as detailed in [9]. The reduction in the memory requirement associating every omnidirectional image to its Fourier signature is large as detailed in Table 1. The similarity between two images is calculated as the $L1$ norm of the Fourier signature of the corresponding panoramic cylinders, see again [9]. The similarity linearly decreases with the distance within a short range and after a certain distance it will saturate, see Fig. 4. This happens because when the two images are taken at points far apart there is no correlation at all between the two images.

In the next section we will introduce the concept of the hierarchical memory-based localisation and how this can be easily calculated from the signature associated to the omnidirectional images. The hierarchical memory-based localisation is one of the advantages of our method based on the *Fourier signature*, with respect to other method of omnidirectional image based localisation.

3 Hierarchical Memory-based Localisation

As we introduced in Section 1, a robot needs to have a different accuracy in the estimation of its position in the environment depending on the structure of the environment. We called this process *hierarchical localisation*.

This idea of the need of different accuracy depending on the navigation task of the robot, was spotted also by other authors, like Santos-Victor [7]. In that work, the robot uses two different image-based navigation strategies: *topological* navigation and *visual-path following* navigation switching between one and the other depending on the kind of motion required to the robot. The drawback of this solu-

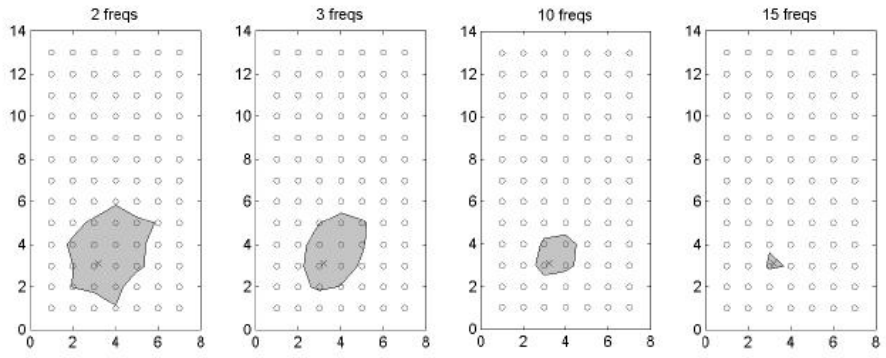


Figure 5: An example of hierarchical localisation. The number of Fourier components used to calculate the similarity function increases from left to right. The empty circles represent the reference images. The full circle represents the actual position of the current image and the grey area represents the calculated possible locations of the robot.

tion is that it requires the design of two different navigation behaviours. Moreover, the visual-path following technique requires handmade design and an accurate control system.

On the contrary, we satisfied the need of different localisation accuracies within the frame of image-based localisation. The approach proposed in this section exploit the technique described in Section 2, and actually improves this technique, reducing its computational cost. In fact, the $L1$ norm used to calculate the similarity between the two function (Eq. 1) is stopped for $k << m$ when a broad localisation of the robot is enough.

$$Sim(O_i, O_j) = \sum_{y=0}^{l-1} \sum_{k=0}^{m-1} |F_{iy}(k) - F_{jy}(k)| \qquad (1)$$

To understand the hierarchical localisation process, we need to spend some more words on the properties of the Fourier transform of an image and in particular on the Fourier signature associated with the panoramic cylinder.

When we calculate the Fourier transform of a brightness signal, like one row of the panoramic cylinder, we are actually decomposing this signal into its component on a set of basis functions. These basis functions are related to the spatial brightness variations. The first basis function, the one with zero frequency, is the constant brightness signal (no variation) and the coefficient associated to it is giving the level of brightness. The basis functions of higher frequency are associated to variations of brightness of higher frequency, i.e. to brightness pattern of higher and higher spatial frequency. When we are calculating the similarity function for two images, using Eq. 1, we are summing up all the contribution from the different frequency components.

When confronting two images, we can see the average brightness of the images changes very slowly when increasing the distance between the two images, while the distribution and the presence of brightness patterns (representing the objects in the environment) changes much faster. Therefore, we can expect that the low frequency components of the Fourier transform of the two images are more similar in a larger interval of distances than the higher frequency components. This means that if in the calculation of the similarity function we stop the calculation of the sum in Eq. 1 at the first Fourier components our current image will match not only the closest reference image, but also a larger number of reference images distributed in the surroundings of the current position.

As a result, we can have a hierarchical localisation just by choosing the number of Fourier components to compare with the similarity function. In other words, if the robot needs only a broad localisation it does not need to calculate the inner sum in Eq. 1 for every value of k, it can just stop at the very first values. The result is to match the current view not only with the closest view but also with other reference views close to it. When a more precise localisation is needed, as in a situation in which the robot has to manoeuvre in a cluttered environment, the sum can be extended to higher values of k in order to have a more strict matching against only one reference image.

In Fig. 5 is depicted a graphical representation of the hierarchical localisation achieved with our system. The empty circles represent the reference images. The full circle represents the actual position of the current image. The possible position of the robot, as calculated by the system, is represented by the grey area. The number of Fourier components used to calculate the similarity function increases from left to right, consequently the grey area showing the possible localisation of the robot is shrinking. In this test the reference images were taken on a 25 cm grid in a office environment cluttered with many pieces of furniture, as you can see from

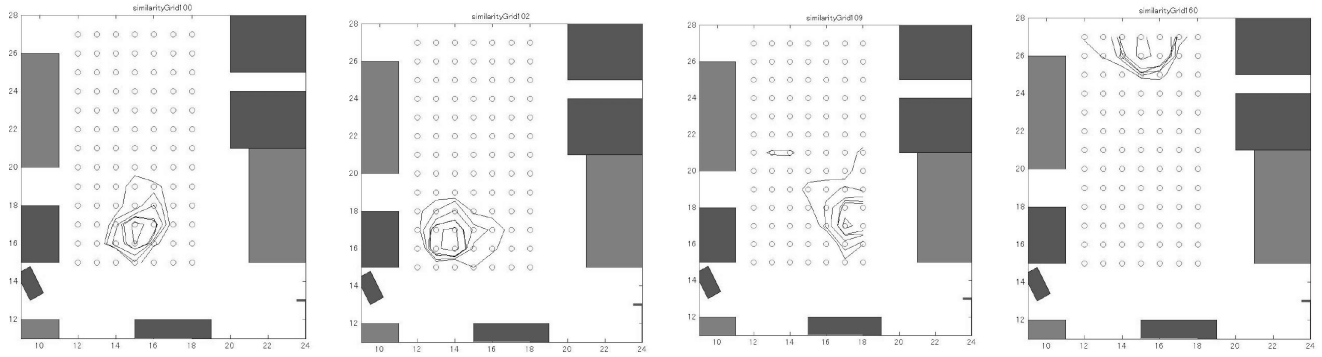


Figure 6: Several examples of hierarchical localisation at different places in the environment. The layout of the room in which experiments were performed is shown and the boxes represent the objects in the environment. Lighter boxes represent lower objects, darker boxes represent taller objects.

pictures in Fig. 1 and Fig. 2.

In Fig. 6, we present the hierarchical localisation obtained at different locations in the same environment. In the figure is sketched also a rough map of the test environment. In wich the objects present in the environment are schetched as boxes of different colour. Lighter boxes represent lower objects (like desks or chairs), darker boxes represent taller objects (like filers or shelves). At the moment of writing, we are investigating the relation between the shape of the localisation areas and the disposition of the objects in the environment.

In summary, our method provide a direct way of calculating the hierarchical localisation for the robot by comparing the frequency spectrum of the current image with the frequency spectrum of the set of reference images. Broad localisation is provided at minimal computational cost, just comparing very few frequency components. When higher accuracy in the localisation is needed, the system will use a little extra computational power.

4 Monte-Carlo Localisation

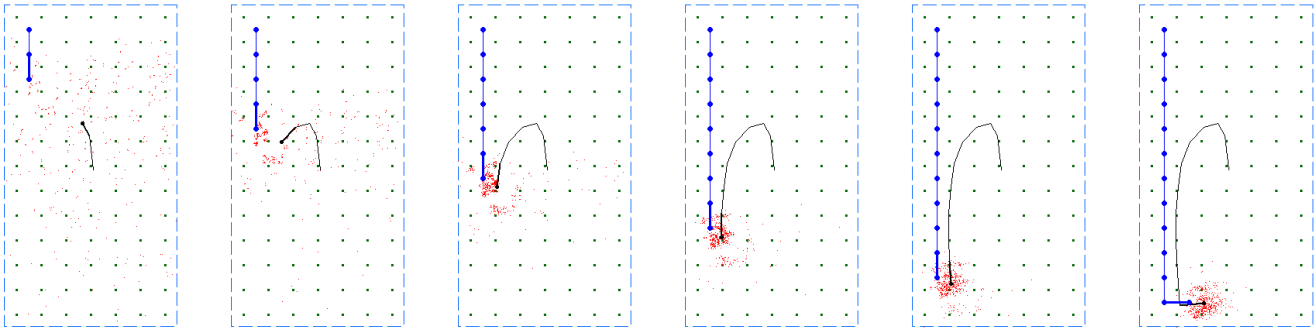
As we stated in the introduction, the image-based localisation approach is mislead in situation in which the apparence of the environment is the same from two different locations. In this paper we overcame this problem by exploiting a well-known probabilistic approach in order to estimate the correct position of the robot. This general method, known as **Bayesian filtering** (also known as *Markov localisation* in robotics) [2, 15], updates recursively the belief about robot position. The belief about the robot position is updated every time the robot makes a new measurement (i.e. it grabs a new image). As the state space is continuous, we used Monte-Carlo Method to represent the belief with a set of weighted samples.

The approach we used is very similar to the one proposed by Burgard in [15]. Every time the robot moves, it grabs a new image. The grabbed image is compared with the reference images in the memory of the robot and a similarity value is calculated for every reference image. These similarity values are used to assign a weight to samples used in the Monte-Carlo localisation process. The pose of the robot is estimated with a standard Monte-Carlo approach. For more details please refer to [2, 15, 4].

There are three main differences with respect to Burgard's work. The first is that we used an omnidirectional vision sensor, so that at every position in the environment is associated one and only one image (with a drastical reduction of the number of required images). The second is that we do not associate any *visibility region* to the reference images in the memory database: this is possible because we used an omnidirectional sensor that supplies a 360° view of the environment; this also means that we rely only on the similarity computation to distinguish various localisation hypothesis with a gain in minor complexity of the algorithm. The third difference is that we implemented the MCL system on an holonomic robot, called Barney. The peculiarity of this robot is that it can move in any direction without the need of a previous rotation. This property suits perfectly an omnidirectional camera that has no need to change robot's heading before grabbing an image. In general this reduces odometric error.

4.1 Experiments

We tested the system in the unmodified small real-world of Fig. 1 and Fig. 2, that is described in Section 3. The tests were performed on the system in off-line mode using a simulator completely parameterisable. The inputs of the simulator are the series of the *reference omnidirectional images*



(1) (2) (3) (4) (5) (6)

Figure 7: An example of Monte-Carlo localisation. These are some snapshot of our system while performing a Global Localisation (1,2,3) and then Position Tracking (4,5,6).

and the sequences of the *current images* presented one by one every time the robot takes a new step. The aim of the experiments is to demonstrate the system is able to reliably localise the robot and to recover from localisation errors.

We carried out several experiments of Global Localisation, Position Tracking and Kidnaped Robot, introducing different amount of noice on the odometers' data. The system is able to successfully localise the robot in any situation. The test environment is rather small (about $5m \times 2m$) and it has the simple shape of a rectangle. This environment was chosen just as first testbed to test the system. At the time of writing we are experimenting in a much more complex environment.

In Fig. 7 we present an example of Monte-Carlo localisation based on the similarity between the current image and the reference images. There the small red dots are the samples generated by the Monte-Carlo filter, the grid dots are the reference locations, the blue line is the actual path of the robot and the black curve is the extimated path of the robot. The estimated position of the robot is calculated as the average position of the samples and is marked with a black square.

Global Localisation

Our system is able to localise the robot without any prior information on the robot's position after processing about 5-6 images. The correct localisation is achieved even if we use a very low number of samples. Experimentally, we observed that the minimal number of samples to obtain a reliable localisation is about the same number of the reference images. In few iterations the error on the position decreases below the grid size. Some screen shots of the experiments are presented in Fig. 7 (1)(2)(3). Note that in (3) the real position of the robot is locked.

Position Tracking

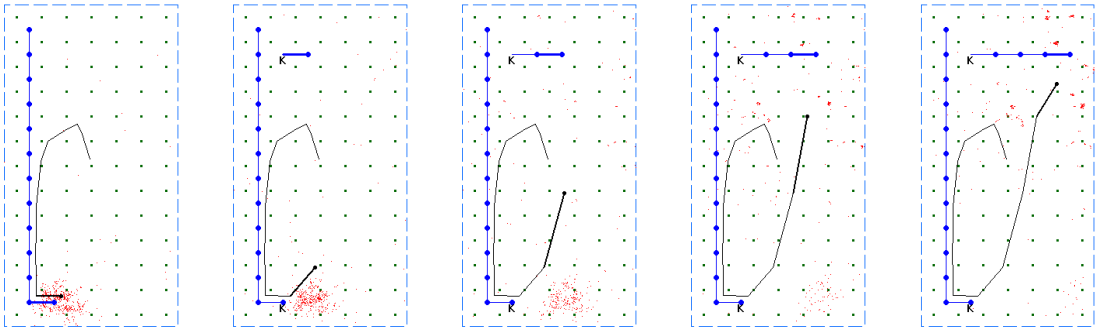
In Fig. 7 (4)(5)(6), the position tracking experiment is presented. The system is able to keep track of robot's position also when the robot takes long steps (about 200 cm) in dead-reckoning mode (i.e. it travels for a while whithout tacking any picture) with large odometric errors. In this condition the system is able to correct the misleading induced by odometry.

Kidnaped Robot

We tested our system on the kidnapped robot problem, i.e. after the robot acquired its position, it is lifted and moved to a different location, see Fig. 8 and Fig. 9.

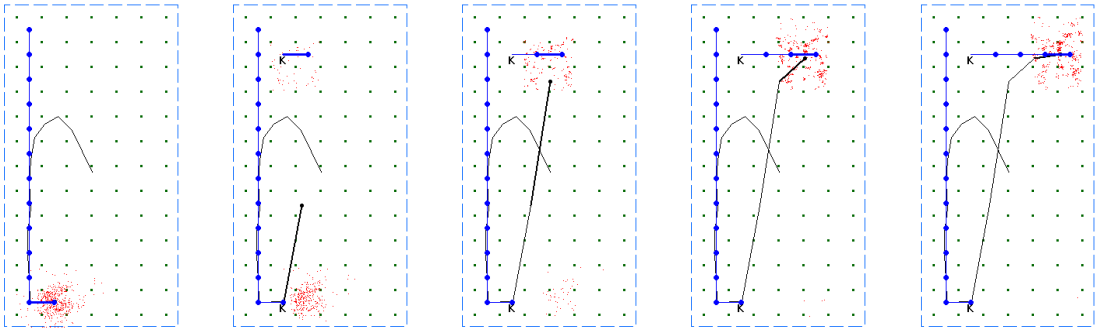
The main difficulty faced by Monte-Carlo localisation methods in the kidnapped robot problem is that when the robot has a good localisation, the samples are generated only in a tight cloud close to the estimated position of the robot. Once the robot is moved in a new position without perceiving this motion (kidnapping), it will continue to generate the samples around the position it thinks is still occupying, without generating any sample around the new unknown position. If the robot does not have any samples around the new position it will never recover from the localisation error.

The standard approach replaces a certain number of samples with others randomly drawn in the entire environment [15]. The result obtained with our system implementing this technique are depicted in Fig. 8. This technique is robust, but in general requires many steps to re-localize the robot. Instead, we exploited the *topological localisation* and the *hierarchical localisation* gave by our image-based localisation approach. At each step a number of samples (10% of samples) are replaced with new samples drawn around the topological localisations (i.e. the reference locations of the



(1) (2) (3) (4) (5)

Figure 8: The kidnaped robot problem with the standard solution of generating a certain number of samples (about 50) in random positions in the whole environment. Note the robot needs many steps in order to recover from the kidnapping.



(1) (2) (3) (4) (5)

Figure 9: The kidnaped robot problem with the proposed *kidnaped strategy* that uses the *topological localisation* and the *hierarchical localisation*. Note how the robot recovers the correct position after 3-4 steps.

most similar reference images), see Fig. 9. This *kidnapping strategy* assures to concentrate the searching for the new robot position only around probable locations. We notice that this approach is possible, because we use an omnidirectional sensor that provide, in a single observation, a more accurate topological localisation than a prospective sensor.

The proposed *kidnapped strategy* can be applied also to the standard Global Localisation with a significative speed-up (almost twice faster) of the convergence of the estimated position to the real position.

At the time of writing we are testing our system in more challenging environments: the first is a long indoor corridor with a loop where we face many false hypothesis; the second is a large outdoor environment where we are investigating the different nature of brigthness variations and its consequences in localisation process.

5 Conclusions

In this paper, we presented the two additional step we took toward a robust image-based localisation system that can operate in every type of environment. We presented our approach to the problem of lowering the computational and memory requirements posed by the image-based localisation. This solution uses the Fourier trasforms of the omnidirectional images grabbed by the robot. We discussed the advantages of this solution with respect to the solutions devised by other authours. We focused on the possibility offered by this representation to implement a hierarchical localisation of the robot. To overcome the limitation of the image-based localisation systems, i.e. the lack of robustness in case of perceptual aliasing, we introduced a Monte-Carlo localisation technique. We showed that this system is able to track the position of the robot while moving and it is able to estimate the position of the robot without any prior knowledge on the real position. At the moment of writing, we are testing robustness of our system in more com-

plex and large environments. We are experimenting within a large outdoor environment and within an indoor environment with high perceptual aliasing due to a long corridor with loop and several identical doors and junctions.

References

[1] H. Aihara, N. Iwasa, N. Yokoya, and H. Takemura. Memory-based self-localisation using omnidirectional images. In A. K. Jain, S. Venkatesh, and B. C. Lovell, editors, *Proc. of the 14th International Conference on Pattern Recognition*, volume vol. I, pages 1799–1803, 1998.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.

[3] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2000.

[4] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. In *IEEE Proc. Radar, Sonar and Navigation*, volume vol. 146, 1999.

[5] R. Cassinis, D. Duina, S. Inelli, and A. Rizzi. Unsupervised matching of visual landmarks for robotic homing unsing fourier-mellin transform. *Robotics and Autonomous Systems*, 40(2-3), August 2002.

[6] T. Collett, E. Dillmann, A. Giger, and R. Wehner. Visual landmarks and route following in desert ants. *Journal of Comparative Physiology A*, 170:pp. 435–442, 1992.

[7] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transaction on Robotics and Automation*, Vol 16(number 6), December 2000.

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

[9] H. Ishiguro and S. Tsuji. Image-based memory of environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-96)*, pages pp. 634–639, 1996.

[10] M. Jogan and A. Leonardis. Robust localization using panoramic view-based recognition. In *Proc. of the 15th Int.Conference on Pattern Recognition (ICPR00)*, volume 4, pages pages 136–139. IEEE Computer Society, September 2000.

[11] B. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appareance-based robot localization. *Image and Vision Computing*, vol. 19(6):pp. 381–391, April 2001.

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

[13] T. Pajdla and V. Hlaváč. Zero phase representation of panoramic images for image based localization. In F. Solina and A. Leonardis, editors, *8-th International Conference on Computer Analysis of Images and Patterns*, number 1689 in Lecture Notes in Computer Scinence, pages 550–557, Tržaška 25, Ljubljana, Slovenia, September 1999. Springer Verlag.

[14] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. In *International Journal of Robotics Research*, volume Vol. 19, pages pp. 972–999, November 2000.

[15] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.