Hindawi Publishing Corporation
Journal of Robotics
Volume 2012, Article ID 797063, 13 pages
doi:10.1155/2012/797063

Research Article

Visual Odometry through Appearance- and Feature-Based Method with Omnidirectional Images

David Valiente García, Lorenzo Fernández Rojo, Arturo Gil Aparicio, Luis Payá Castelló, and Oscar Reinoso García

Departamento de Ingenierıa de Sistemas y Automática, Universidad Miguel Hernández, Avda. de la Universidad s/n, 03202 Elche (Alicante), Spain

Correspondence should be addressed to David Valiente García, dvaliente@umh.es

Received 30 March 2012; Revised 16 July 2012; Accepted 31 July 2012

Academic Editor: Bojan Nemec

In the field of mobile autonomous robots, visual odometry entails the retrieval of a motion transformation between two consecutive poses of the robot by means of a camera sensor solely. A visual odometry provides an essential information for trajectory estimation in problems such as Localization and SLAM (Simultaneous Localization and Mapping). In this work we present a motion estimation based on a single omnidirectional camera. We exploited the maximized horizontal field of view provided by this camera, which allows us to encode large scene information into the same image. The estimation of the motion transformation between two poses is incrementally computed, since only the processing of two consecutive omnidirectional images is required. Particularly, we exploited the versatility of the information gathered by omnidirectional images to perform both an appearance-based and a feature-based method to obtain visual odometry results. We carried out a set of experiments in real indoor environments to test the validity and suitability of both methods. The data used in the experiments consists of a large sets of omnidirectional images captured along the robot's trajectory in three different real scenarios. Experimental results demonstrate the accuracy of the estimations and the capability of both methods to work in real-time.

1. Introduction

The problems of robot localization and SLAM (simultaneous localization and mapping) are of paramount importance in the field of mobile autonomous robots, since a model of the environment is often required for navigation purposes. To the present days, the estimation of the robot motion through the environment has been a task of large interest, since it provides a helpful tool when it lacks any previous knowledge about the environment. Normally, this motion is computed incrementally at each step of the robot's trajectory. With this aim, several types of sensors such as GPS, IMS, 3D laser data, and wheel encoders (robot odometry) have been used to acquire valid information to compute an estimation. However, visual sensors arise as an emerging tendency for such purpose as they present several benefits such as low power consumption, low weight, relative low cost, and the richness of the information from the environment they offer. Likewise, the term of visual odometry involves the determination of the current pose of the robot and the path covered by retrieving a motion transformation between images, which are captured at every pose traversed by the robot. This process is performed in an incremental manner by processing consecutive images. The estimation is inferred from the motion effects revealed by the apparent changes on the scene. Figure 1 establishes the relative motion parameters, θ, ϕ, and ρ, between two consecutive poses.

As it will be detailed in Section 2, different types of visual sensors may be used to estimate a visual odometry. Our approach is based on a single omnidirectional camera. Due to its capability to represent the scene with a maximum field of view, we expect this sensor to generate a robust representation of the environment and to provide us with more information compared to other common camera sensors when variations in large-scale scenarios are considered. Therefore, the harmful instability effects on the scene, which appear

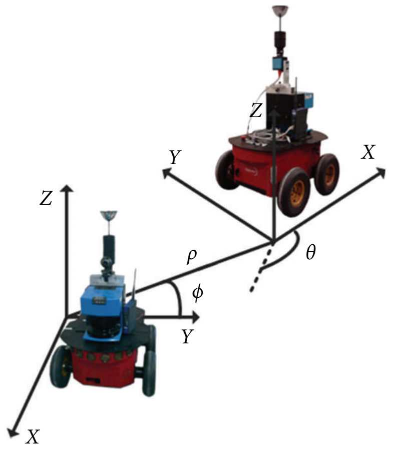


Figure 1: Two consecutive poses of the robot and the relative motion constraints between them.

in large uneven scenarios, may be reduced by exploiting omnidirectional image benefits. For instance, in presence of occlusions or big lighting changes, the appearance of a perspective image could be dramatically modified, whereas this effect is diminished in the case of an omnidirectional image since its field of view is wider. In this sense omnidirectional images achieve to get successful results when localizing the robot in long-distance runs. Moreover we take advantage from the versatility of the omnidirectional images to process the information with appearance-based and feature-based methods. A set of experiments with real data have been carried out by using an indoor robot equipped with an omnidirectional camera. Experimental results reveal the validity of both types of methods. The feasibility of both schemes is analyzed in terms of accuracy and computational cost in order to demonstrate its validity for real-time applications.

The paper is organized as follows. Section 2 introduces some related work in the field of visual odometry. Next, Sections 3 and 4 describe the appearance-based and feature-based methods respectively. Section 5 presents the results gathered from the experiments carried out with real data. Finally, Section 6 states a brief discussion to analyze the results obtained.

2. Related Work

Approaches to visual odometry can be classified according to the kind of data sensor used to estimate the trajectory of the robot. For instance, stereo cameras have been extensively used to estimate a visual odometry. In [1] an orientation sensor was introduced together with a stereo pair to outperform previous visual odometry results [2] obtained by taking advantage of Moravec's work [3]. In [4] a precedent was set in terms of real-time working and error removal. However, single cameras, as well denoted as monocular visual sensors, have also achieved successful results despite the fact that they only can recover a 2D relative motion. There exists application of both pinhole cameras [5, 6] and omnidirectional cameras [7, 8]. First, single camera examples introduced a *RANSAC* algorithm for outlier removal, whereas the last omnidirectional camera examples went further through the obtention of the car's trajectory along large scenarios.

Some distinctions may also be defined depending on the image-processing model. Main classifications are referred to appearance-based methods and feature-based methods. Appearance-based methods concentrate their efforts on the information extracted from the pixel intensity. For instance in [9] a Fourier-Mellin transform is applied to omnidirectional images in order to obtain a visual descriptor for a visual odometry application. In the same manner, [10] implements a Fourier signature [11] with the aim of retrieving a visual odometry with panoramic images. An other example is [12] where an appearance method based on orientations histograms is implemented to get the ego-motion of an autonomous vehicle.

Regarding feature-based methods, these approaches try to detect distinctive and robust points or regions between consecutive images. In [13] monocular feature point detection is combined with optical flow to retrieve a visual odometry. In [14], a search of repeatable points is set up with the same aim. In [15] some robust points are also extracted to obtain a relative motion transformation intended for SLAM applications. After the statement of [16] where an efficient algorithm was presented for outlier removal, most parts of feature-based methods are sustained by modifications of it, such as the approach presented in [7].

In the next sections we present our methods for the appearance- and feature-based odometry.

3. Appearance-Based Visual Odometry

In this section, we face the visual odometry problem as an appearance-based relative camera pose-recovering problem, using the whole appearance of the panoramic images, without any feature extraction process. We describe a real-time algorithm for computing an appearance-based map through visual and robot odometry.

3.1. Global Appearance Descriptor.

When we have to build on an appearance descriptor, this must be robust against changes in the environmental lighting conditions. It has to be built in an incremental way to allow us to obtain the visual odometry while the robot is going through the environment. Moreover, the computational cost to obtain the descriptor must be low, to permit working in real time. We build our descriptor using the Fourier signature over a set of previously filtered omnidirectional images as it presents all these features [11, 17].

3.1.1. Fourier Signature.

To date, different description methods have been used in the context of omnidirectional robot vision. The advantages of the Fourier signature are its simplicity, computational cost, and the fact that it exploits better the invariance against ground-plane rotations using panoramic images. This transformation consists in the expansion of each row of the panoramic image $\{a_n\} = \{a_0, a_1, \ldots, a_{N_y-1}\}$ using the discrete Fourier transform into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \ldots, A_{N_y-1}\}$. With this transform, the most relevant information is concentrated in the low-frequency components of each row, so it is possible to work only with the information from the k first columns in the Fourier transform. Also, this feature presents rotational invariance when working with panoramic images. If each row of the original panoramic image is represented by the sequence of complex number $\{a_n\}$ and each row of the rotated image by $\{a_{n-q}\}$ (being q the amount of shift), it is possible to prove that when the Fourier transform of the shifted sequence is computed, we obtain the same amplitudes A_k than in the nonshifted sequence, and there is only a phase change, proportional to the amount of shift q

$$F\left[\left\{a_{n-q}\right\}\right] = A_k \exp - j\frac{2\pi qk}{N_y}; \quad k = 0, \dots, N_y - 1. \quad (1)$$

Taking advantage of this property, we can compute the robot orientation using only the phase information of the Fourier signature.

3.1.2. Homomorphic Filtering. When we work in a real environment, it can be expected that small changes in the appearance of the environment occur. These changes are mainly due to the lighting conditions and changes in the position of some objects. This way, the descriptor must be robust against these small variations in the environment.

In a previous work [18], we show how it is possible to increase the accuracy of the localization of a robot in a previously created map by applying homomorphic filtering techniques on the panoramic images captured. In this work, we have used such techniques to filter the images as a preprocessing step before extracting the Fourier signature.

This filtering technique allows us to filter separately the luminance and reflectance components of an image [19]. Accordingly, it is possible to control the influence of each component on the image appearance. From the natural logarithm operator on the image, the homomorphic operator separates the components of luminance and reflectance:

$$\begin{aligned} I(x, y) &= l(x, y) \times r(x, y), \\ z(x, y) &= \ln(I(x, y)), \\ z(x, y) &= \ln(l(x, y)) + \ln(r(x, y)), \end{aligned} \quad (2)$$

where the panoramic image $I(x, y)$ is represented as a multiplication of the luminance $l(x, y)$ and the reflectance component $r(x, y)$ of the image. Once the components have been separated and we have applied the 2D discrete Fourier transform on the panoramic image, we apply a filter on the frequency domain image:

$$\begin{aligned} F[\{z(x, y)\}] &= F[\{\ln(l(x, y))\}] + F[\{\ln(r(x, y))\}], \\ F[\{z'(x, y)\}] &= F[\{z(x, y)\}] \cdot H(u, v), \end{aligned} \quad (3)$$

where the filter transfer function in the frequency domain is represented by $H(u, v)$. Due to the fact that the low-frequency components are associated with the illumination of the image and the high-frequency ones with the reflectance, we apply a high-pass filter constructed from a Butterworth low pass filter to reduce the effects of changes in the lighting of the scenes [19].

3.2. Visual Odometry Estimation. To carry out the relative camera pose recovering, we have to establish some relationships between the stored poses. It can be shown that when we have the Fourier signature of two panoramic images that have been captured in two points that are geometrically close in the environment, it is possible to compute their relative orientation using the shift theorem (1).

Taking into account that the Fourier Signature is invariant to ground-plane rotations, we can establish a relationship between the phases of the Fourier signature of a panoramic image captured at one position and the phases of the Fourier signature of another panoramic image captured at the same point but with a different orientation. We can expand this property and get the approximate rotation $\phi^{t+1,t}$, between two panoramic images taken on two consecutive poses. To obtain this angle, we convolute the phases of the Fourier signature of the panoramic images of the two poses. This operation implies a comparison between the first image and a rotated versions of the second one. The number of rotations equals the row size of the image. Once we have the Euclidean distance for each comparison, we retain the image with minimum distance and calculated the angle using the number of pixels for the rotated image.

Once we have shown how the relative rotation between two consecutive images has been computed, we have to establish the method employed to build a visual odometry for a robot's trajectory. We build a poses-based graph where when a new image is captured, a new pose is added to the graph, and the topological relationships with the previous pose are computed using the global appearance information of the scenes. With our procedure, this computation is made online as the robot is going through the environment, in a simple and robust way.

In this case we consider that the trajectory that the robot traverses is composed of a set of poses $L = \{l^1, l^2, \dots, l^N\}$, where each pose l^j is represented by an omnidirectional image $I^j \in \Re^{N_x,N_y}$ and a Fourier descriptor composed of a modules matrix $d^j \in \Re^{N_x,k_1}$ and a phases matrix $p^j \in \Re^{N_x,k_2}$. Therefore, with the algorithm used, we can obtain the position (l_x^j, l_y^j) and the orientation l_ϕ^j of each pose of the trajectory with respect to the previous one; thus $l^j = \{(l_x^j, l_y^j, l_\phi^j), d^j, p^j, I^j\}$.

The robot captures a new image at time t+1, and then, the Fourier descriptors d^{t+1} and p^{t+1} are computed. Comparing p^{t+1} with the descriptor of the previously captured image p^t, we can find the relative position between these two poses. We obtain the Euclidean distance $D^{t+1,t}$ between two consecutive poses by using data supplied by the robot's internal odometry, denoted as odo_x^t and odo_y^t referring to the x and y cartesian coordinates at the instant t. On the other hand, thanks to the visual compass implemented, we can estimate the relative orientation between images. After this process, the position of the current pose is computed from the previous one as

$$\begin{aligned} l_x^{t+1} &= l_x^t + D^{t+1,t} \cdot \cos(\phi^{t+1,t}), \\ l_y^{t+1} &= l_y^t + D^{t+1,t} \cdot \sin(\phi^{t+1,t}), \end{aligned}$$

$$l_\phi^{t+1} = l_\phi^t + \left(\phi^{t+1,t}\right),$$

$$D^{t+1,t} = \sqrt{\left(\mathrm{odo}_x^{t+1} - \mathrm{odo}_x^t\right)^2 + \left(\mathrm{odo}_y^{t+1} - \mathrm{odo}_y^t\right)^2}. \tag{4}$$

These relationships are shown graphically in Figure 2.

4. Feature-Based Visual Odometry

The objective of this section is to present the procedure to obtain a visual odometry by using specific feature points extracted from the omnidirectional images. As already mentioned, the horizontal field of view of omnidirectional cameras is maximum so that images provide a large information of the current scene captured. This fact allows us to obtain a set of significative pixel points in the image plane that correspond to a set of physical 3D points. These points are expected to show high saliency, concept that measures their visual relevance on the image, and in addition it is used to classify them as interest points whether they fulfill a specific metric applied to the pixel and its local surrounding. This visual significance of each interesting point is encoded into a visual descriptor; therefore later comparisons can be established when looking for corresponding points between images.

4.1. Detection of Interest Points. Traditionally, some tracking techniques have been used to find corresponding points [1], provided that there exists a short scale between consecutive poses from where images were taken. On the other hand, current approaches [4] use global searches at the whole image together with similarity descriptor tests in order to find correspondences. We have adopted a modification of this later scheme because of our purpose in computing motion in large-scale scenarios, where image appearance may vary considerably from one image to another. In this context, different combination of feature point detectors and descriptors has been commonly used, such as Harris corner detector [20], SIFT features [21], and more recently SURF features [22]. In [23], SURF features outperform other detection methods and description encodings in terms of invariance and robustness of the detected points and their descriptors. In addition, SURF features have been successfully performed [24] with omnidirectional images. Consequently, these facts suggested us the use of SURF features to detect points and describe their visual appearance. Particularly we transform the omnidirectional images into a panoramic view in order to increase the number of valid matches between images due to the lower appearance variation obtained with this view.

4.2. Visual Odometry Estimation. Once a set of interest points have been detected and matched in two images, it is necessary to compute the relative motion of the robot. As shown in Figure 1, this motion is formed by a rotation and a translation, which are stated by two relative angles θ and ϕ and a scale factor ρ. The same relation can be expressed into the image plane, which is observed in Figure 3 with some correspondences indicated. Retrieving a certain camera rotation and translation to recover a relative motion has been generally performed by applying epipolar constraints to solve a 6DOF problem, such as [16, 25, 26], whereas in our case, according to the planar motion on the XY plane, we simplify the estimation to a 3DOF problem.

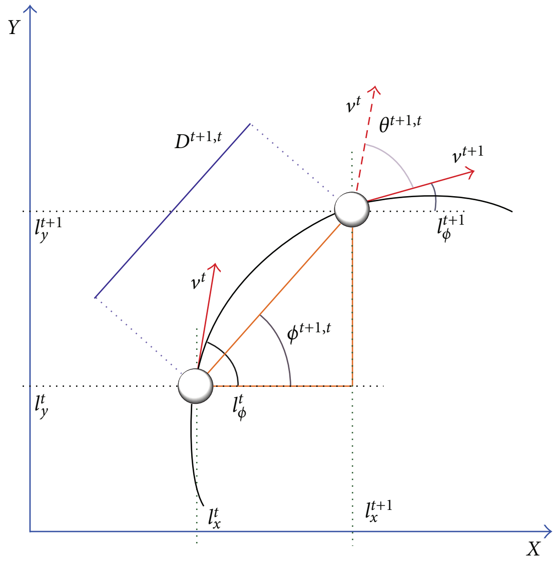


Figure 2: Position and orientation of the new pose in the map computed incrementally from the previous pose.

The same point detected in two images can be expressed as $p = [x, y, z]^T$ and $p' = [x', y', z']^T$ in the first and second camera reference system, respectively. Then, the epipolar condition establishes the relationship between two 3D points p and p' seen from different poses:

$$p'^T E p = 0, \tag{5}$$

where matrix E is denoted as the essential matrix. Notice that E is composed by a certain rotation R and a translation T (up to a scale factor) that relate both images, being $E = R \cdot T_x$. Therefore, according to R and T, the essential matrix presents the following structure:

$$E = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

$$R = \begin{bmatrix} 0 & 0 & \sin(\phi) \\ 0 & 0 & -\cos(\phi) \\ \sin(\theta-\phi) & \cos(\theta-\phi) & 0 \end{bmatrix}, \tag{7}$$

$$T = \rho\left[\cos(\phi) \quad \sin(\phi) \quad 0\right]^T. \tag{8}$$

Being ϕ and θ the relative angles that define a motion transformation between two different poses are shown in Figures 1 and 3.

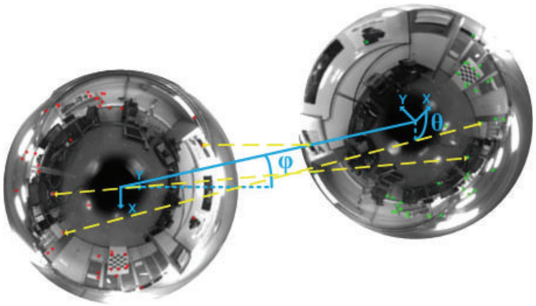


Figure 3: Real omnidirectional images acquired at different poses of the robot with the correspondences and relative motion constraints indicated.

Matrix E can be computed by directly applying the epipolar condition to the N detected points in the following manner:

$$\begin{bmatrix} p_1' \\ p_2' \\ \cdots \\ p_N' \end{bmatrix}^T \cdot \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_N \end{bmatrix} = \vec{0}. \tag{9}$$

This last equation leads to a common N-linear equation system:

$$D\vec{e} = 0 \tag{10}$$

being $\vec{e} = \{e_{13}, e_{23}, e_{31}, e_{32}\}$ and D the coefficient matrix. Please note the simplification due to the planar motion characteristics, since only 4 variables are required to solve (10). Thus, we face the overdetermined system (10) by using only $N = 4$ corresponding points which suffice to solve the problem; however we use a larger number of correspondences in order to obtain a reliable solution in presence of outliers. A singular value decomposition (SVD) has been adopted to solve (10). Then, following [27, 28] it is easy to retrieve R and T. Next subsection deals with the error point considerations and introduces our approach to avoid outliers.

4.3. Error Propagation and Outlier Removal. In the suggested framework, the problem of finding correspondences between images is troublesome. An error propagation term is always associated to each motion transformation. This error is not only introduced by the sensor deviation when detecting feature points. Besides, the number of outliers considered as true matched points are error points that deviate the estimated solution, but also the current uncertainty of the pose of the robot corrupts the estimation. As long as the robot moves, each relative motion transformation (from now on, denoted as M_j) introduces an error term that deviates the robot's pose accumulatively, so that the uncertainty of the pose increases. In this situation, methods as *RANSAC* [8], *histogram voting* [14], or *epipolar matching* [29] have been used cope with uncertainties. In this approach, we alternatively suggest the use of a gaussian distribution to model the error and propagate it to the computation of M_j in a more realistic manner. Furthermore, this propagation becomes into a helpful tool in order the restrict the search for valid correspondences.

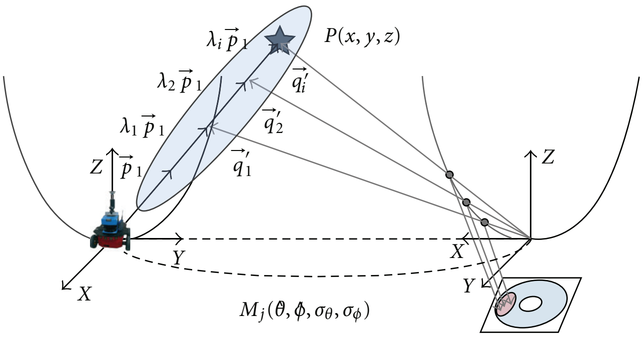


Figure 4: Given a detected point $\vec{p}_1$ in the first image reference system, a point distribution is generated to obtain a set of multiscale points $\lambda_i\vec{p}_1$. By using the predicted motion transformation $\widehat{M}_j$, they can be transformed to $\vec{q}_i'$ in the second image reference system through $\widehat{M}_j \sim N(\hat{\theta}, \hat{\phi}, \sigma_\theta, \sigma_\theta)$. Finally $\vec{q}_i'$ are projected into the image plane to determine a restricted area where correspondences have to be found.

In an idealistic case, the epipolar restriction (5) may equal a fixed threshold, meaning that the epipolar curve between images always presents a little static deviation. On a realistic approach we should consider that this threshold depends on the existing uncertainty of the pose. To do that, we assume that the robot's odometry is known, which allows to predict a motion transformation $\widehat{M}_j$. Eventually the epipolar condition becomes

$$p'^T \hat{E} p < \delta\left(\widehat{M}_j\right), \tag{11}$$

where $\delta(\widehat{M}_j)$ expresses the uncertainty on the pose of the robot, since it is directly correlated with the predicted transformation $\widehat{M}_j$.

This approach enables a reduction in the search for corresponding points between images. Figure 4 depicts the procedure. A detected point $P(x, y, z)$ is represented in the first image reference system by a normalized vector $\vec{p}_1$ due to the unknown scale. To deal with this scale ambiguity, we suggest a generation of a Gaussian point distribution to get a set of multiscale points $\lambda_i\vec{p}_1$ for $\vec{p}_1$. Please note that the current uncertainty of the pose of the robot has to be propagated along the procedure. Next, since we obtained a predicted motion transformation $\widehat{M}_j$, we can transform the distribution $\lambda_i\vec{p}_1$ into the second image reference system, obtaining $\vec{q}_i'$. We propagate the uncertainty of the pose to redefine the motion transformation through a normal distribution, being $\widehat{M}_j \sim N(\hat{\theta}, \hat{\phi}, \sigma_\theta, \sigma_\phi)$, and thus implying that $\vec{q}_i'$ is a gaussian distribution correlated with the current uncertainty of the pose. Once $\vec{q}_i'$ are obtained, they are projected into the image plane of the second image, represented as crosses in Figure 5. This projection of the normal multiscale distribution defines a predicted area in the omnidirectional image which is drawn with a curved continuous

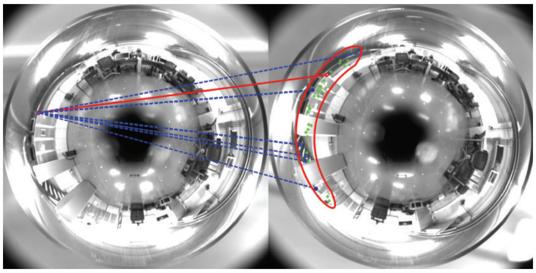


FIGURE 5: Correspondences for a specific point in the first image are searched for in the predicted area, which is projected into the second image. Crossed points represent the projection of the normal point distribution for the multiscale points that determine this area. Dash lines show the candidate points in the second image which are inside the predicted area. Continuous line represents the correct correspondence for a certain point in the first image. Curved continuous line shows the shape of the predicted area.

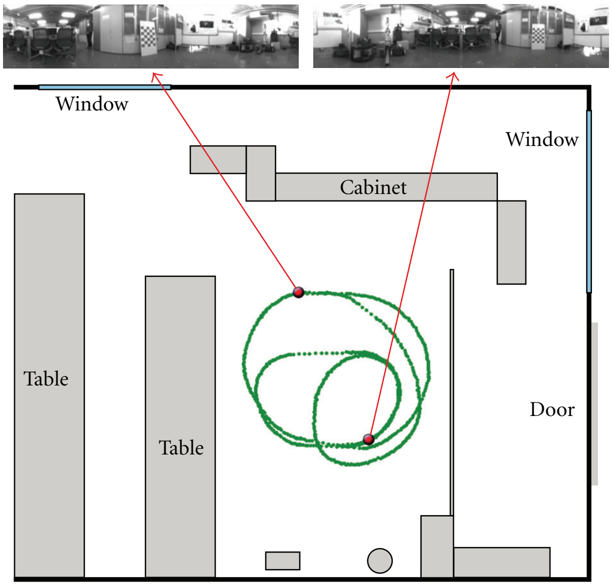


FIGURE 6: Trajectory followed by the robot when capturing *data set* 1.

line. This area establishes the specific image pixels where correspondences for $\vec{p}_1$ must be searched for. The shape of this area depends on the error of the prediction, which is directly correlated with the current uncertainty of the pose of the robot. Dash lines represent the possible candidate points located in the predicted area. Therefore, the problem of matching is reduced to finding the correct corresponding point for $\vec{p}_1$ from those candidates inside the predicted area by comparing their visual descriptor, instead of searching for them at the whole image. Notice that using this technique, the number of false correspondences is reduced since we avoid them through this restricted matching process, limited to a specific image areas as explained above.

5. Experimental Settings

5.1. Description of the Environment. In order to acquire the necessary data for the experiments, we have used a Pioneer P3-AT mobile robot, equipped with a DMK 21BF04 omnidirectional camera, an internal odometer, and a SICK LMS-200 laser (Figure 1). The robot performed three different trajectories in three different indoor environments, capturing a new omnidirectional image, odometry, and laser data whenever it traversed a distance equal to 0.1 m. The characteristics of the three paths traversed by the robot are shown in Table 1. The paths followed by the robot and some examples of panoramic images for each set are shown in Figures 6, 7, and 8, respectively.

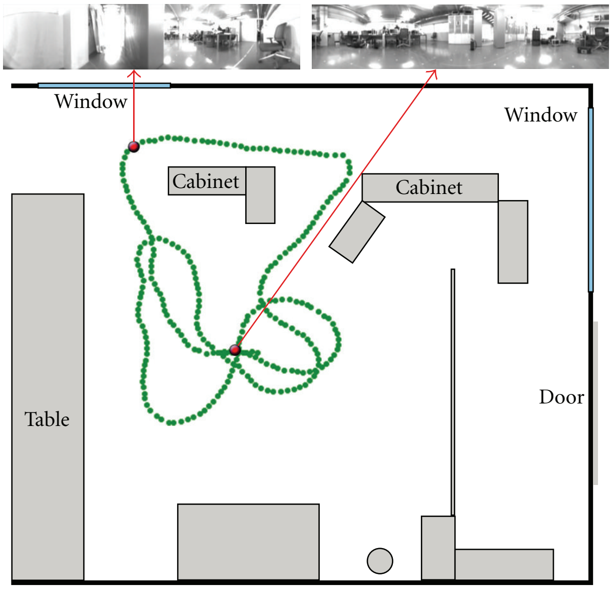


FIGURE 7: Trajectory followed by the robot when capturing *data set* 2.

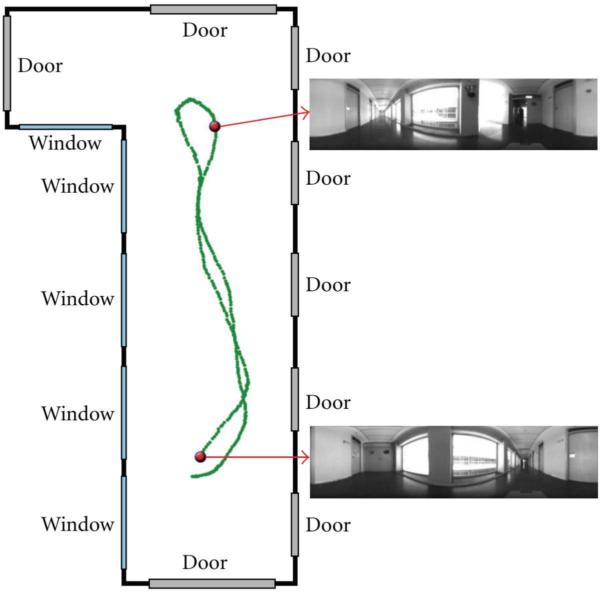
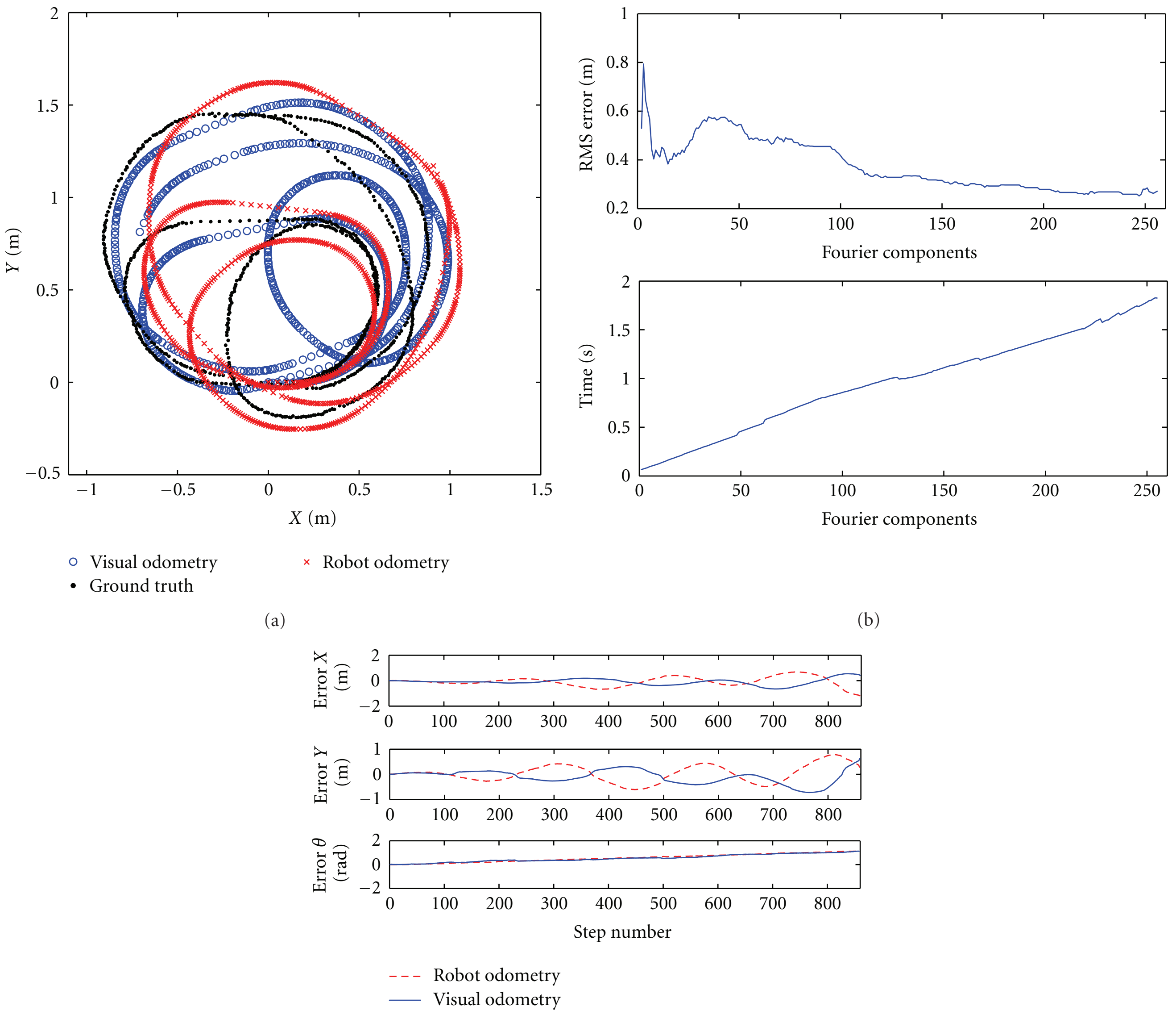


FIGURE 8: Trajectory followed by the robot when capturing *data set* 3.



(a)

(b)

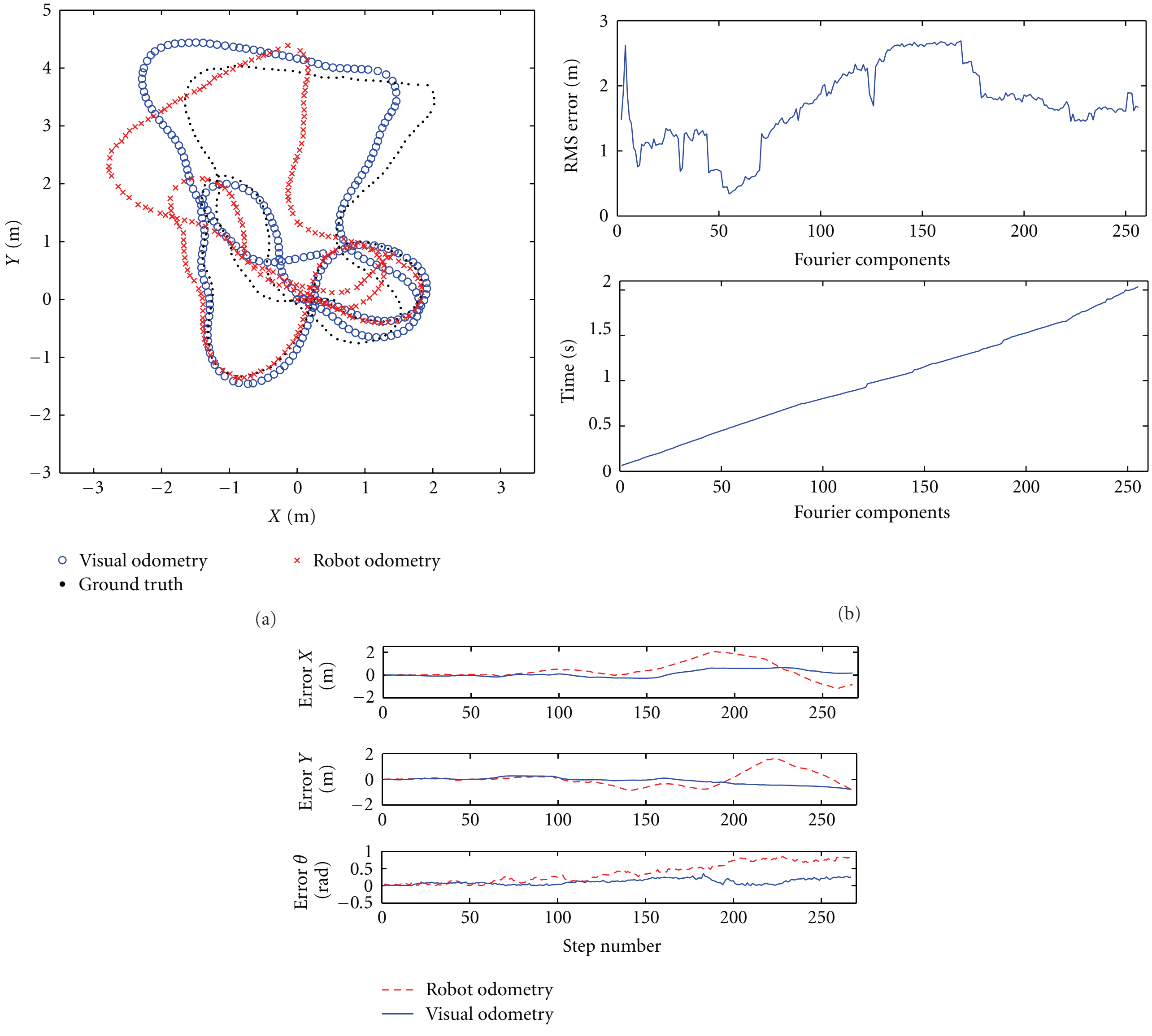
(c)

Figure 9: (a) Example of an experiment using the data set 1 and the appearance-based method, using 64 Fourier components for each row of the Fourier signature. (b) RMS error and step time versus the number of Fourier components for each row of the Fourier signature using the appearance-based method. (c) X, Y and θ error for each step of the visual odometry estimation process using the appearance-based method and using 64 Fourier components for each row of the Fourier signature.

5.2. Experimental Results. To test the performance and compare our appearance-based and feature-based visual odometry methods, we have carried out a series of simulation experiments using the sets of images described in the previous subsection. To demonstrate the feasibility of the procedures we have computed the *RMS* (root mean squared) error in the robot position along the trajectory, the error in x, the error in y, and the error in θ for each step of the process and the computational time necessary to calculate each pose of the robot. In each case, we test the influence of the main parameters of the descriptors used.

The *RMS* error has been obtained taking as a reference the real path (Figures 9(a), 10(a), and 11(a). To get the real path, we have used a SICK LMS range finder in order to compute a ground truth using the method presented in [30]. We must take into account the RMS error of the robot odometry data comparing with the ground truth is 0.665 m for the *data set* 1, 0.768 m for the *data set* 2 and 1.112 m for the *data set* 3.

Figures 9(a), 10(a), and 11(a) show an example of the visual odometry (black dots) computed, the ground truth (blue circles) that represent the real path, and the robot odometry (red crosses), for the first, the second, and the third set of images, respectively. This visual odometry has been built using the appearance-based method and a number of Fourier components for each row of the Fourier signature



(a)

(b)

(c)

Figure 10: (a) Example of an experiment using the data set 2 and the appearance-based method, using 64 Fourier components for each row of the Fourier signature. (b) RMS error and step time versus the number of Fourier components for each row of the Fourier signature using the appearance-based method. (c) X, Y, and θ error for each step of the visual odometry estimation process using the appearance-based method and using 64 Fourier components for each row of the Fourier signature.

Table 1: Relevant parameters of the image sets used in the experiments.

Environment	Name	Number images	Image resolution	Distance traversed	Environment size
Laboratory 1	Data set 1	858	720 × 720 pixels	85.8 m	1.8 × 1.7 m
Laboratory 2	Data set 2	267	720 × 720 pixels	26.7 m	3.7 × 5.4 m
Corridor	Data set 3	416	720 × 720 pixels	41.6 m	1.4 × 7.1 m

equal to 64 for the three cases. The visual odometry obtained outperforms the robot odometry for all scenarios, for any number of Fourier components chosen. Figures 9(b), 10(b), and 11(b) show the average error of the entire experiment comparing to the real path when using the first set of images and the computational time needed to add a new pose to the visual odometry. As we can see, for the three scenarios, this figure gives us information about the number of Fourier components that allows to outperform the robot odometry and to work in a realtime. It is worthy to note that generally it exists an optimum number of Fourier components in terms of RMS error. The increasing in the number of Fourier

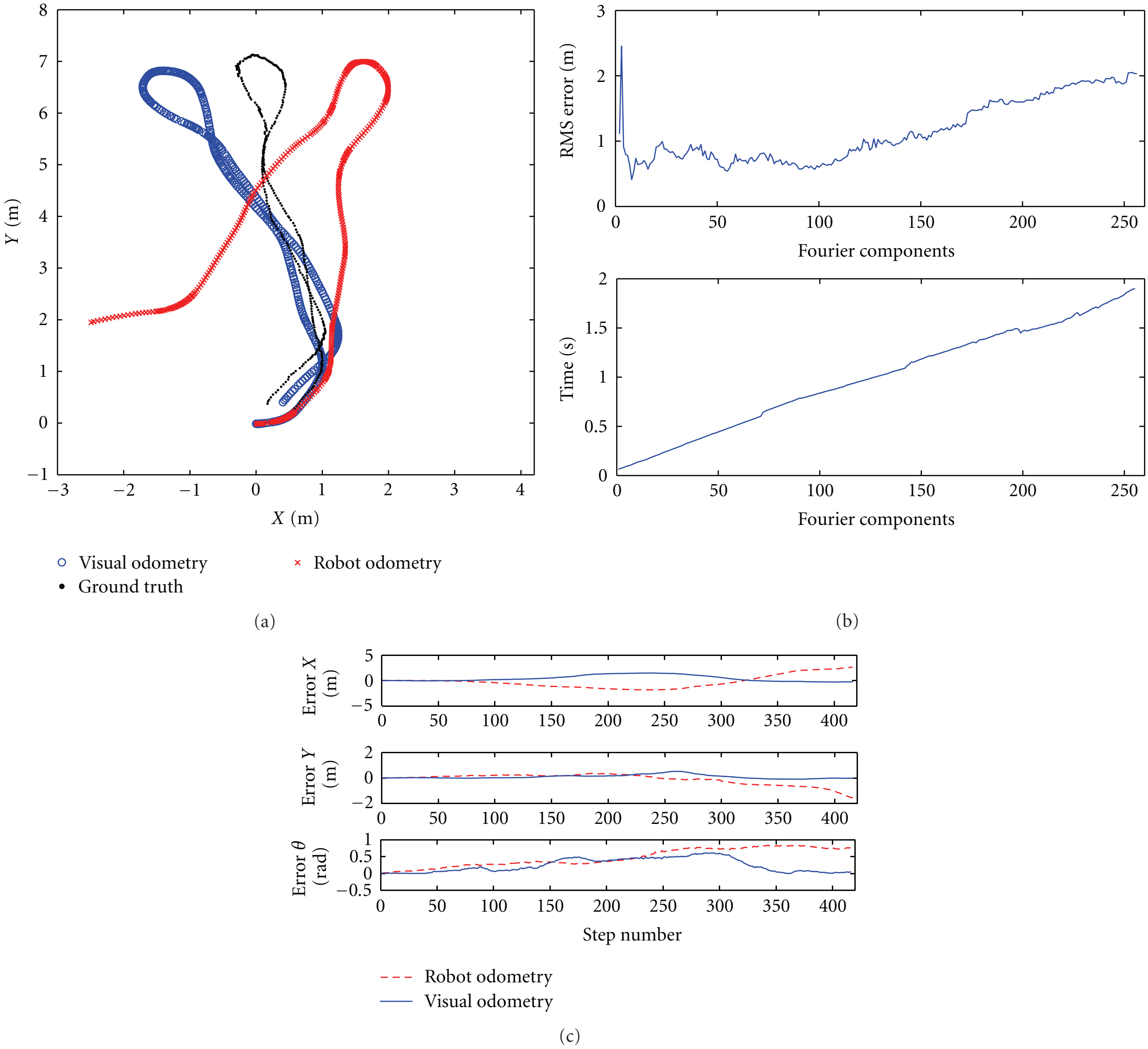


Figure 11: (a) Example of an experiment using the data set 3 and the appearance-based method, using 64 Fourier components for each row of the Fourier signature. (b) RMS error and step time versus the number of Fourier components for each row of the Fourier signature using the appearance-based method. (c) X, Y, and θ error for each step of the visual odometry estimation process using the appearance-based method and using 64 Fourier components for each row of the Fourier signature.

components does not involve better RMS error results since high-frequency components of the Fourier descriptor may introduce noise, as it can be observed in Figures 10(b) and 11(b). However, in Figure 9(b) we obtain a different result due to the circular movement of the robot along the environment, so that the robot traverses around close poses, fact that makes the RMS error not to increase in a dramatic manner. In Figures 9(c), 10(c), and 11(c), we plot the X, Y, and θ error both for robot odometry (red dashed line) and visual odometry (blue line) for each step of the procedure in one of the experiments carried out. In general, the figures show that our method outperforms the robot odometry at each step for the three data sets. With this method it is possible to obtain an average error of the visual odometry data comparing with the ground truth of 0.485 m for the *data set* 1, 0.575 m for the *data set* 2, and 0.705 m for the *data set* 3 with a computational time that allows us to work in real time (0.594 s for the *data set* 1, 0.533 s for the *data set* 2, and 0.549 s for the *data set* 3) when we use 64 Fourier components.

On the other hand, we have carried out the same experiments, but in this case the visual odometry has been built using the feature-based method. Figures 12(a), 13(a), and 14(a) show an example of the visual odometry (black dots) computed, the ground truth (blue circles) that represent the real path and the robot odometry (red crosses), for the three

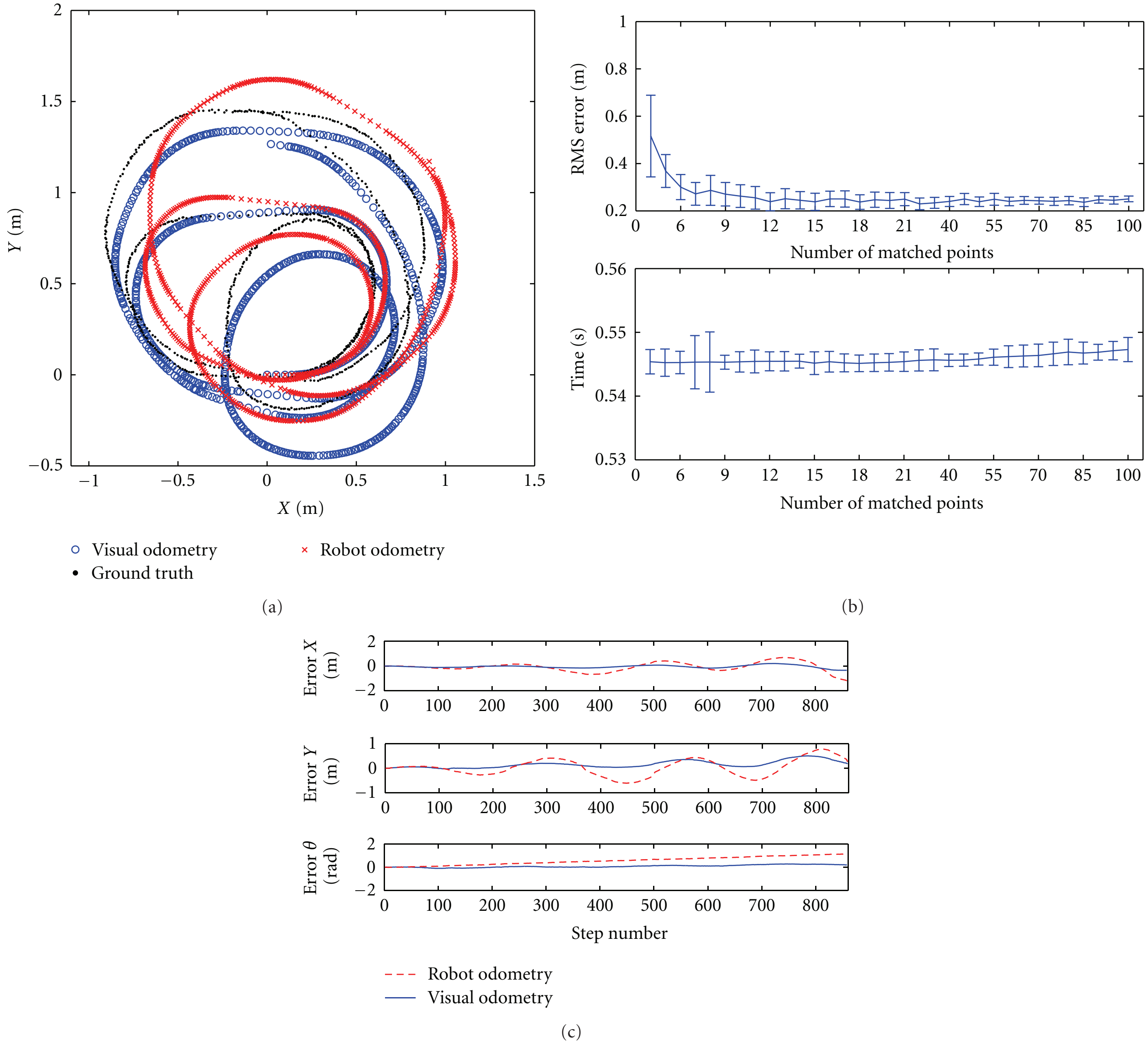
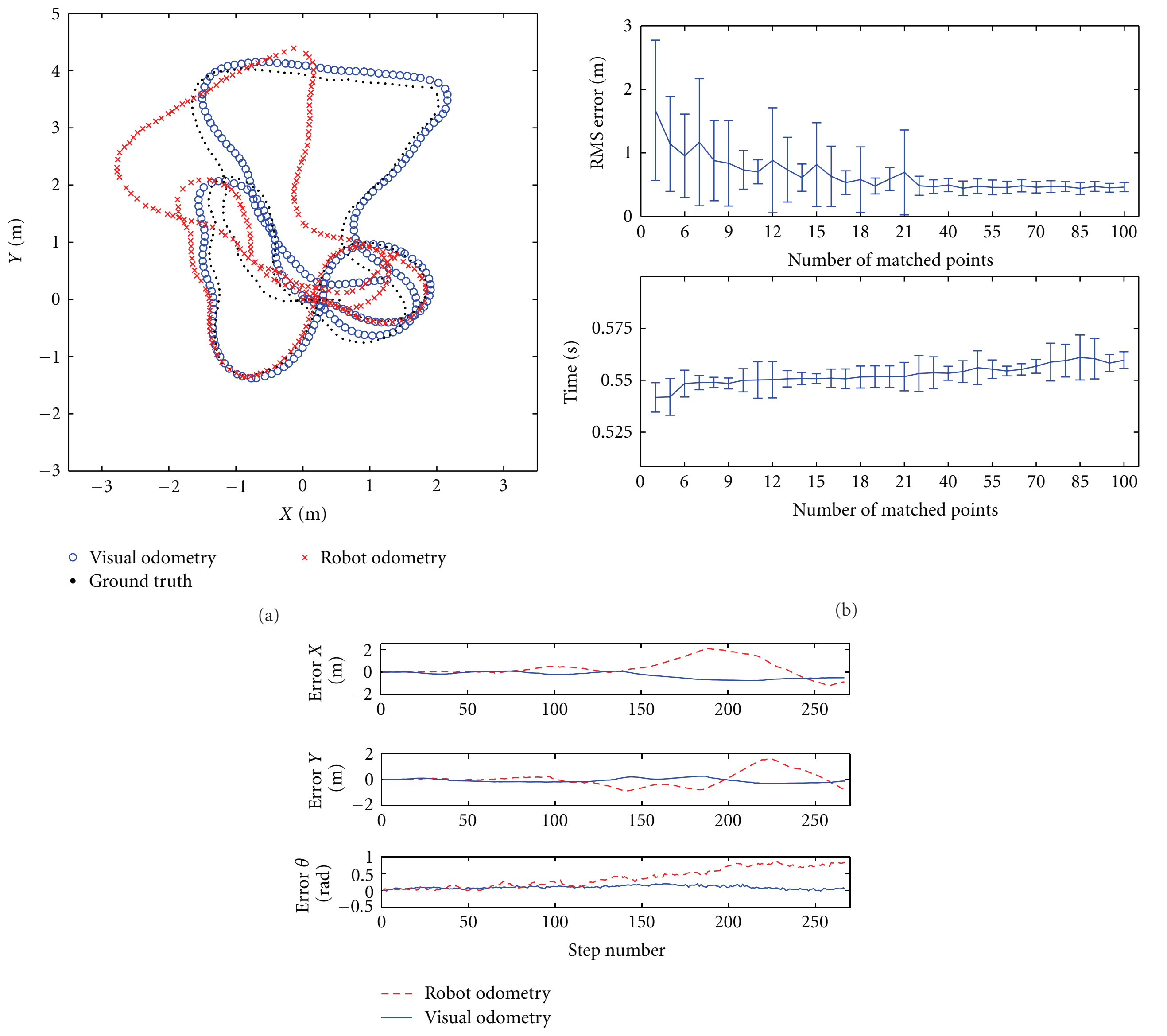


Figure 12: (a) Example of an experiment using the data set 1 and the feature-based method, using 20 matched points. (b) RMS error and step time versus the number of matched points using the feature-based method. (c) X, Y, and θ error for each step of the visual odometry estimation process using the feature-based method using 40 matched points.

sets of images, consecutively. This visual odometry has been built using a number of matched points equal to 40 for the three cases. The figures show that, for the number of matched points chosen, the visual odometry obtained outperforms the robot odometry for all scenarios. Figures 12(b), 13(b), and 14(b) show the average error of the entire experiment comparing to the real path when using the first set of images and the computational time needed to add a new pose to the visual odometry. We can see as for the three scenarios, we can used a number of matched points that allow us to clearly outperform the robot odometry and to work in a real time. Particularly, looking at Figures 12(b), 13(b), and 14(b) we can observe that the increase of the number of matched points considered is not a dramatic fact for the real-time operation. It is worthy to note that the main computational load is caused by the feature point detection, which always has to be carried out between two images. Once we have extracted the entire set of feature points, both matching and the SVD-solving processes are optimized in terms of computational cost, so that it is shown a low dependency on the number of feature points versus the time increase. In Figures 12(c), 13(c), and 14(c), we show an example of an experiment where we plot the X, Y, and θ error both for robot odometry (red dashed line) and visual odometry (blue line) for each step of the procedure. In general, the figures show that our method outperforms the robot odometry at



(a)

(b)

(c)

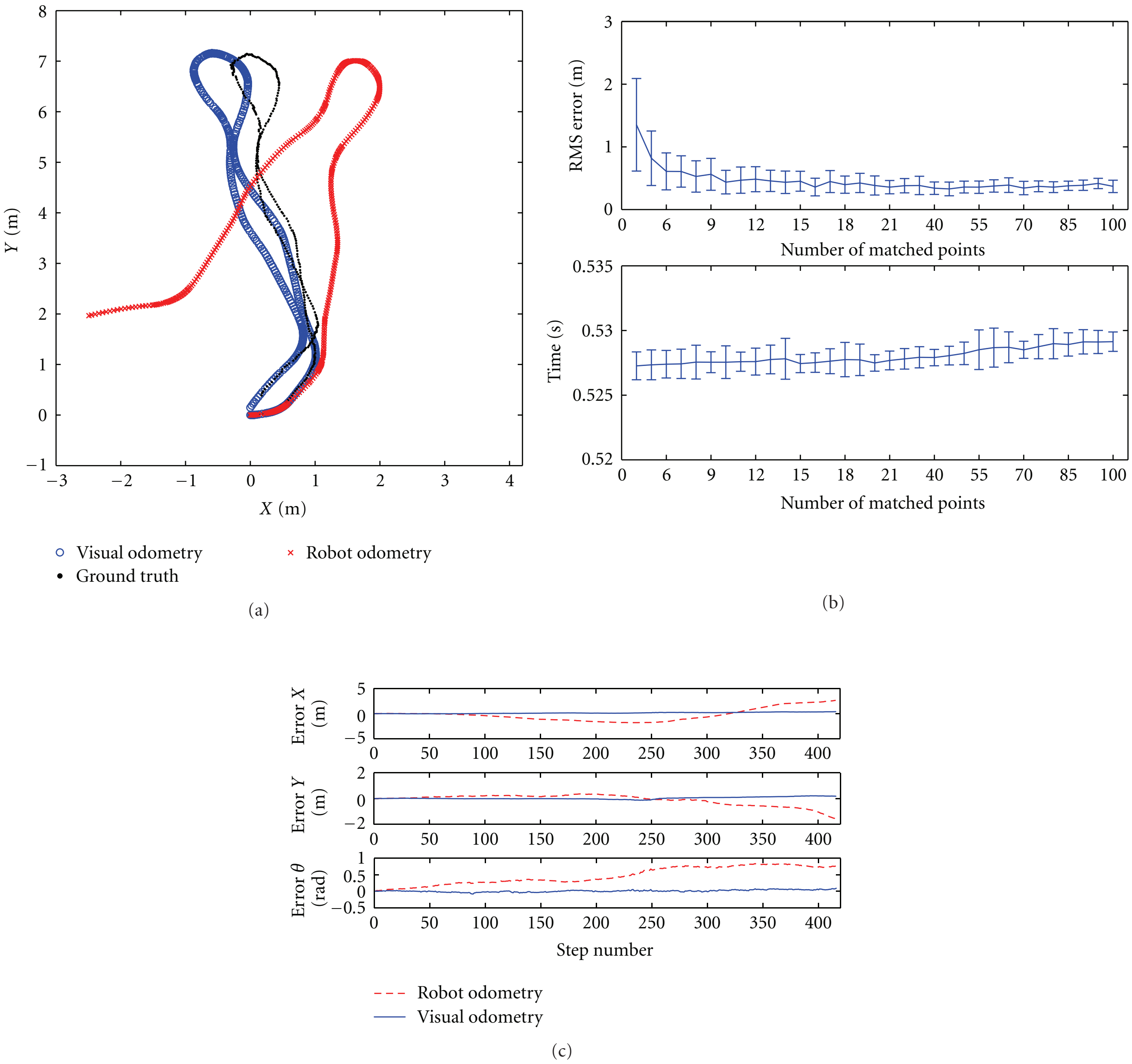
FIGURE 13: (a) Example of an experiment using the data set 2 and the feature-based method, using 20 matched points. (b) RMS error and step time versus the number of matched points using the feature-based method. (c) X, Y, and θ error for each step of the visual odometry estimation process using the feature-based method using 40 matched points.

each step for the three data sets. With this method it is possible to obtain an average error of the visual odometry data comparing with the ground truth of 0.239 m for the *data set* 1, 0.494 m for the *data set* 2, and 0.340 m for the *data set* 3 with a computational time that allow us to work in a real time (0.512 s for the *data set* 1, 0.551 s for the *data set* 2, and 0.527 s for the *data set* 3) when we use a number of matched points equal to 40.

Finally, we must highlight the fact that in cases where we work with appearance-based methods, we only represent the average error for the experiments and not the standard deviation. This is due to the fact that it is a deterministic method. In contrast, when working with the method based on features, it is necessary to represent the standard deviation in error, because it is nondeterministic.

6. Conclusions

In this paper we have presented two visual odometry approaches with omnidirectional images, which may be intended for Localization and SLAM applications in the context of mobile autonomous robotics. We suggest taking advantage of the benefits provided by omnidirectional images, since large information about the scene is encapsulated in a single image. Consequently we extract the estimation of the relative motion transformation between



(a)

(b)

(c)

Figure 14: (a) Example of an experiment using the data set 3 and the feature-based method, using 20 matched points. (b) RMS error and step time versus the number of matched points using the feature-based method. (c) X, Y, and θ error for each step of the visual odometry estimation process using the feature-based method using 40 matched points.

consecutive poses, which is computed from the processing of two consecutive omnidirectional images through both appearance-based and feature-based methods, respectively. This estimation is completely obtained in an incremental manner. A set of experiments carried out with real data acquired with an indoor robot have been presented in order to verify the reliability of both methods. The results gathered from the three different scenarios show that accurate estimations for the motion of the robot can be achieved. Further analysis of these results comes up with the determination of a specific setting parameters for both appearance-based and feature-based methods that generate an optimal solution in terms of accuracy. Besides, the solution accomplishes with real-time requirements. This fact is highly recommended for many applications in mobile robotics.

References

[1] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, “Robust stereo ego-motion for long distance navigation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’00)*, pp. 453–458, June 2000.

[2] L. Matthies, *Dynamic stereo vision [Ph.D. thesis]*, Carnegie Mellon University, 1989.

[3] H. Moravec, *Obstacle avoidance and navigation in the real world by a seeing robot rover [Ph.D. thesis]*, Stanford University, 1980.

[4] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’04)*, pp. I652–I659, July 2004.

[5] D. Nistér, “Preemptive RANSAC for live structure and motion estimation,” *Machine Vision and Applications*, vol. 16, no. 5, pp. 321–329, 2005.

[6] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[7] J. P. Tardif, Y. Pavlidis, and K. Daniilidis, “Monocular visual odometry in urban environments using an omnidirectional camera,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’08)*, pp. 2531–2538, September 2008.

[8] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, “Real-time monocular visual odometry for on-road vehicles with 1-Point RANSAC,” in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA ’09)*, 2009.

[9] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, “Visual vehicle egomotion estimation using the Fourier-Mellin transform,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’07)*, pp. 450–455, June 2007.

[10] L. Fernández, L. Payá, O. Reinoso, and F. Amorós, “Appearance-based visual odometry with omnidirectional images. A practical application to topological mapping,” in *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO ’11)*, 2011.

[11] E. Menegatti, T. Maeda, and H. Ishiguro, “Image-based memory for robot navigation using properties of omnidirectional images,” *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.

[12] D. Scaramuzza and R. Siegwart, “Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1015–1026, 2008.

[13] P. Corke, D. Strelow, and S. Singh, “Omnidirectional visual odometry for a planetary rover,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’04)*, pp. 4007–4012, October 2004.

[14] D. Scaramuzza, “Performance evaluation of 1-point RANSAC visual odometry,” *Journal of Field Robotics*, vol. 28, no. 5, pp. 792–811, 2011.

[15] A. Gil, D. Valiente, O. Reinoso, L. Fernández, and J. M. Marín, “Building visual maps with a single omnidirectional camera,” in *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO ’11)*, 2011.

[16] D. Nistér, “An efficient solution to the five-point relative pose problem,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’03)*, pp. 195–202, June 2003.

[17] L. Payá, L. Fernández, O. Reinoso, A. Gil, and D. Úbeda, “Appearance-based dense maps creation: comparison of compression techniques with panoramic images,” in *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics (ICINCO ’09)*, pp. 250–255, July 2009.

[18] L. F. Rojo, L. Payá, O. Reinoso, A. Gil, and M. Juliá, “Robust methods for robot localization under changing illumination conditions: comparison of different filtering techniques,” in *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART ’10)*, vol. 1, pp. 223–228, Valencia, Spain, January 2010.

[19] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley Longman, Boston, Mass, USA, 2nd edition, 1992.

[20] C. G. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of the Alvey Vision Conference*, 1988.

[21] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “SURF: speeded up robust features,” in *Proceedings of the 9th European Conference on Computer Vision (ECCV ’06)*, 2006.

[23] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, “A comparative evaluation of interest point detectors and local descriptors for visual SLAM,” *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.

[24] A. C. Murillo, J. J. Guerrero, and C. Sagüés, “SURF features for efficient robot localization with omnidirectional images,” in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA’07)*, pp. 3901–3907, April 2007.

[25] R. Kawanishi, A. Yamashita, and T. Kaneko, “Construction of 3D environment model from an omni-directional image sequence,” in *Proceedings of the Asia International Symposium on Mechatronics*, 2008.

[26] H. Stewénius, C. Engels, and D. Nistér, “Recent developments on direct relative orientation,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

[27] R. Bunschoten and B. Kröse, “Visual odometry from an omnidirectional vision system,” in *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 577–583, September 2003.

[28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[29] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proceedings of the International Conference on Computer Vision (ICCV ’03)*, pp. 1403–1410, October 2003.

[30] C. Stachniss, G. Grisetti, D. Hahnel, and W. Burgard, “Improved rao-blackwellized mapping by adaptive sampling and active loop-closure,” in *Proceedings of the Workshop on Self-Organization of Adaptive behavior (SOAVE ’04)*, pp. 1–15, 2004.

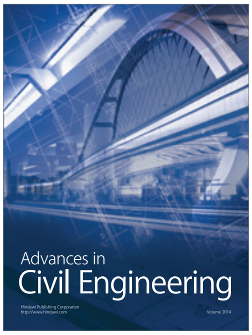
Journal of
Engineering

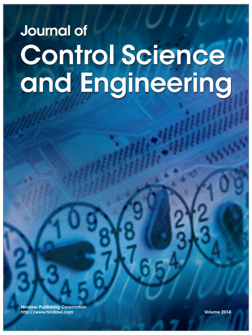
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

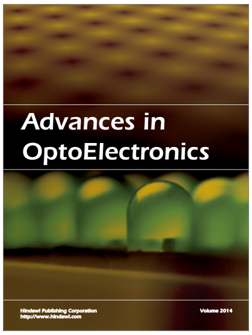
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

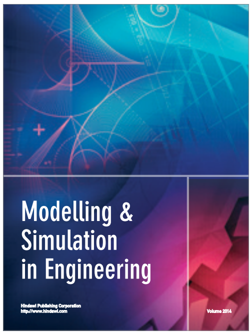
Journal of
Robotics

Journal of
Electrical and Computer Engineering

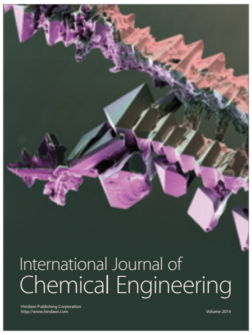

Advances in
OptoElectronics

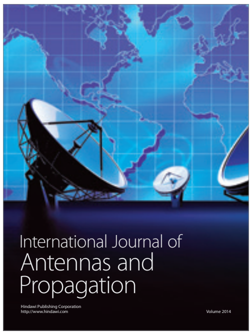
VLSI Design

International Journal of
Navigation and Observation

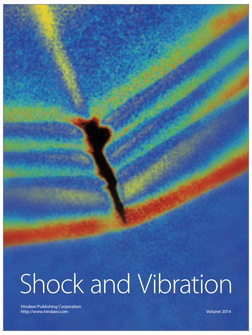

Modelling &
Simulation
in Engineering

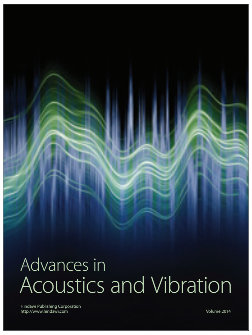
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration