

PROCEEDINGS OF SPIE

[SPIDigitalLibrary.org/conference-proceedings-of-spie](https://spiedigitallibrary.org/conference-proceedings-of-spie)

Visual environment recognition for robot path planning using template matched filters

Ulises Orozco-Rosas
Kenia Picos
V́ctor H. D́az-Raḿrez
Oscar Montiel
Roberto Sepúlveda

Visual environment recognition for robot path planning using template matched filters

Ulises Orozco-Rosas^{1,2}, Kenia Picos^{1,2}, Víctor H. Díaz-Ramírez², Oscar Montiel², and Roberto Sepúlveda²

¹CETYS Universidad, Ave. CETYS Universidad No. 4, Fracc. El Lago C.P. 22210, Tijuana, B.C., México

²Instituto Politécnico Nacional - CITEDI, Ave. Instituto Politécnico Nacional No. 1310, Col. Nueva Tijuana C.P. 22435, Tijuana, B.C., México

ABSTRACT

A visual approach in environment recognition for robot navigation is proposed. This work includes a template matching filtering technique to detect obstacles and feasible paths using a single camera to sense a cluttered environment. In this problem statement, a robot can move from the start to the goal by choosing a single path between multiple possible ways. In order to generate an efficient and safe path for mobile robot navigation, the proposal employs a pseudo-bacterial potential field algorithm to derive optimal potential field functions using evolutionary computation. Simulation results are evaluated in synthetic and real scenes in terms of accuracy of environment recognition and efficiency of path planning computation.

Keywords: Environment recognition, robot vision, template matching filters, path planning, pseudo-bacterial potential field.

1. INTRODUCTION

At the present time, robotics is one of the most important technologies since it is a fundamental part in automation and manufacturing process.¹ In particular, there is a demand for autonomous mobile robots in various fields of application, such as material transport, cleaning, monitoring, guiding people, and military applications. These autonomous mobile robots must interact with their environment to accomplish their tasks. Computer vision techniques are widely used to sense and to perform the environment recognition, which is frequently changing and unforeseen. This work addresses the problem of autonomous mobile robot navigation, understanding that the mobile robot will have to interact with the environment and avoid collisions with obstacles, following the path planned to achieve its established mission. All of these tasks, without the assistance of a human operator. There are three tasks that must be carried out by an autonomous mobile robot to enable the execution of the navigation. These activities are: world perception consisting in **obstacle detection** and modeling the environment; **path planning** for obtaining an ordered sequence of objective points and convert this sequence into a path; the **path tracking** in controlling the robot to follow the path.

In this work, an accurate obstacle detection algorithm is implemented in order to model the environment of the robot. We employ a template matching approach which allows to reliably detect obstacles and accurately estimate their location coordinates. To achieve an effective autonomous mobile robot navigation, we propose the integration of a recognition algorithm based on template-matching technique and a path planning algorithm based on pseudo-bacterial potential field² (PBPF). Computer simulation results are presented and discussed in terms of accuracy of the recognition proposal, effectiveness of the path planning, and computational efficiency.

The paper is organized as follows. Section 2 describes the theoretical background of obstacle detection using template matched filtering, and path planning using the PBPF. Section 3 presents the proposed methodology for visual environment recognition for robot path planning. Section 4 presents and discuss the results given by the experiments in real environments for robot navigation. Finally, Section 5 summarizes our conclusions.

Further author information:

U. Orozco-Rosas: E-mail: ulises.orozco@cetys.mx

Optics and Photonics for Information Processing XI, edited by Khan M. Iftikharuddin, Abdul A. S. Awwal, Mireya García Vázquez, Andrés Marquez, Víctor Díaz-Ramírez, Proc. of SPIE Vol. 10395, 103950N · © 2017 SPIE · CCC code: 0277-786X/17/\$18 · doi: 10.1117/12.2273596

2. BACKGROUND

The theoretical background for visual environment recognition for robot path planning is described in this section. The fundamentals for obstacle detection using template matched filters and the path planning background with PBPF are explained below.

2.1 Visual environment recognition

Consider the scenario from Fig. 1 in order to monitoring the environment of a mobile robot. The optical setup consists in an upper camera in which captures the working area of the robot. Let be $f(x, y)$ the input scene which is composed by a feasible workspace for navigation $b(x, y)$, and several obstacles $\tilde{t}(x, y)$ placed at unknown location (x_0, y_0) . Also, the input scene may be degraded with additive noise $n(x, y)$. The signal model of the input scene can be represented by the following expression³

$$f(x, y) = \tilde{t}(x, y) + b(x, y)[1 - \tilde{w}(x, y)] + n(x, y), \quad (1)$$

where $\tilde{t}(x, y)$ is formed by the superposition of several obstacles $\{t^j(x, y) | j = 1, \dots, J\}$, and the term $\tilde{w}(x, y)$ is a binary function which represents the support area of all the obstacles $t^j(x, y)$. So, the mathematical representation of the scene can be denoted by

$$f(x, y) = \sum_{j=1}^J t^j(x - x_0^j, y - y_0^j) + b(x, y) \sum_{j=1}^J (1 - w^j(x - x_0^j, y - y_0^j)) + n(x, y), \quad (2)$$

where J is the number of obstacles presented in the scene, each one placed in an unknown location (x_0^j, y_0^j) .

For obstacle detection, template matched filtering is an accurate technique based on correlation operations in frequency domain. Correlation filtering is a linear system, in which its impulse response is computed in order

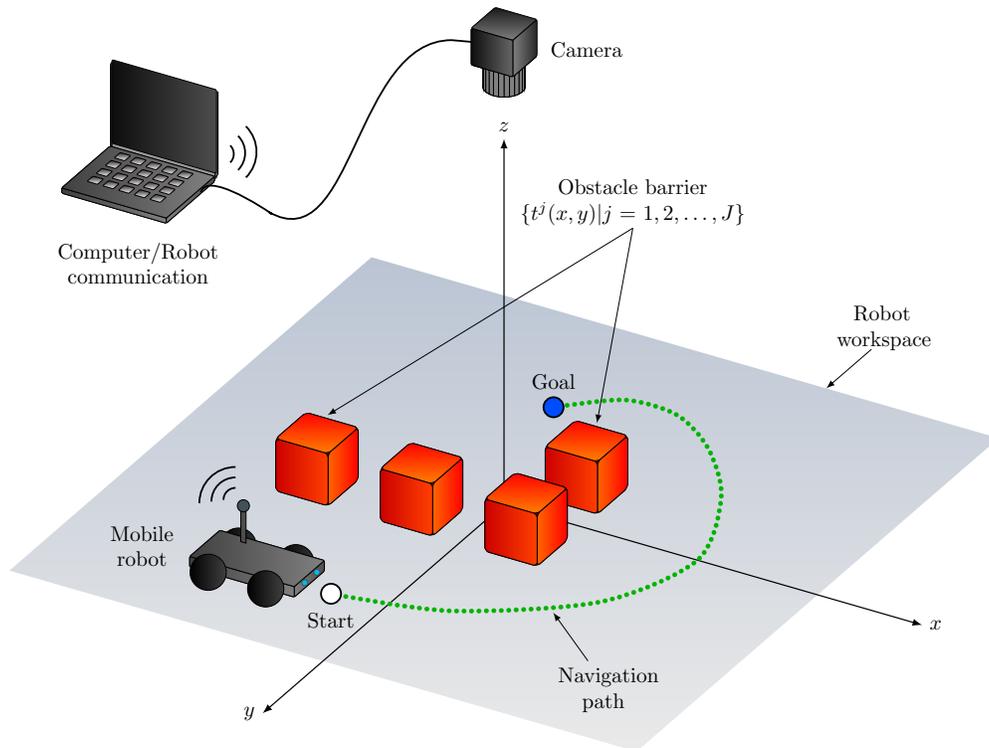


Figure 1. Optical setup for obstacle detection for mobile robot path planning

to produce a high matching value between the input image and an appropriate filter. The designed filter is related to a prior target reference information. Correlation filters can be synthesized by optimization of several performance measures.^{4,5} An optimal filter design for target detection in terms of signal-to-noise ratio (SNR) is the generalized matched filter.⁶ Also the generalized matched filter (GMF) optimized the variance of location errors.^{7,8} Given the input signal of Eq. 1, the frequency response of a GMF is given as follows⁹

$$H(\mu, \nu) = \frac{T(\mu, \nu) + W_b(\mu, \nu) + W_t(\mu, \nu)}{\frac{1}{2\pi}|W_b(\mu, \nu)|^2 * N_b(\mu, \nu) + \frac{1}{2\pi}|W_t(\mu, \nu)|^2 * N_t(\mu, \nu)}. \quad (3)$$

Here, $T(\mu, \nu)$, and $W_b(\mu, \nu)$ are the Fourier transforms of $t(x, y)$ and $w(x, y)$, respectively, where $W_t(\mu, \nu)$ is the Fourier transform of $1 - w(x, y)$. The terms m_b and m_t represents the mean value of the background and target signal, respectively. The terms $N_b(\mu, \nu)$ and $N_t(\mu, \nu)$ are the spectral density functions of the background and target, respectively.

2.2 Path planning

In this work, a PBPF algorithm is employed for path planning. The PBPF algorithm makes use of the artificial potential field¹⁰ (APF) method and mathematical programming, using a metaheuristic based on a pseudo-bacterial genetic algorithm¹¹ (PBGA) as the global optimization method.

The main idea of the APF method is to establish an attractive potential field force around the goal point, as well as to establish a repulsive potential field force around obstacles. The two potential fields acting together (attractive + repulsive) form the total potential function $U_{total}(q)$. The APF method searches the falling of the potential function to find a collision-free path, which is built from the start position to the goal position.²

The total potential function $U_{total}(q)$ can be obtained by the sum of the attractive potential and repulsive potential,

$$U_{total}(q) = \frac{1}{2} \left[k_a (q - q_f)^2 + k_r \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 \right], \quad (4)$$

where q represents the robot position vector in a two-dimensional workspace, $q = [x, y]^T$. The vector q_f represents the goal position and k_a is a positive scalar-constant that represents the attractive proportional gain of the function. The expression $q - q_f$ is related to the linear distance between the robot and the goal position. The repulsive potential function has a limited range of influence; this prevents the movement of the robot from being affected by a distant obstacle, where ρ_0 represents the limit distance of influence of the potential field, and ρ is the shortest distance to the obstacle; the positive scalar-constant k_r is the repulsive proportional gain.

The generalized force $F_{total}(q)$ which is used to drive the robot from the start position to the goal position, see Fig. 1, it is obtained by the negative gradient of the total potential function $U_{total}(q)$,¹⁰ this force is expressed as follows

$$F_{total}(q) = -\nabla U_{total}(q). \quad (5)$$

In Eq. 4, all the parameters are known except for the positive scalar-constants k_a and k_r . Many ways can be used to know the adequate value of this proportional gains, the most common methods are mathematical analysis and approximate methods.¹² In this work for the PBPF algorithm, the APF method is blended with a PBGA to find the optimal (or nearly optimal) values for the proportional gains k_a and k_r .

The PBGA introduced a genetic operator called bacterial mutation¹¹ that has demonstrated to be useful in environments with a weak relationship between the parameters of a system. It is a simple algorithm that presents a fast convergence and improvement in the solutions¹³ (in our case the values for the proportional gains k_a and k_r), without being detrimental in landscape exploration. The core of PBGA contains the bacterium which is able to carry a copy of a gene from a host cell and insert it into an infected cell. By the bacterial mutation, the characteristics of a single bacterium can spread to the rest of the population (i.e., solutions), hence this method mimics the process of microbial evolution.¹⁴ The PBGA can be algorithmically modeled for computer simulation using the difference equation expressed as follows

$$P(\tau + 1) = s(v(P(\tau))), \quad (6)$$

where τ represents the time, the new population $P(\tau + 1)$ is obtained from the current population $P(\tau)$ after it was operated by random variation v , and selection s .¹⁵

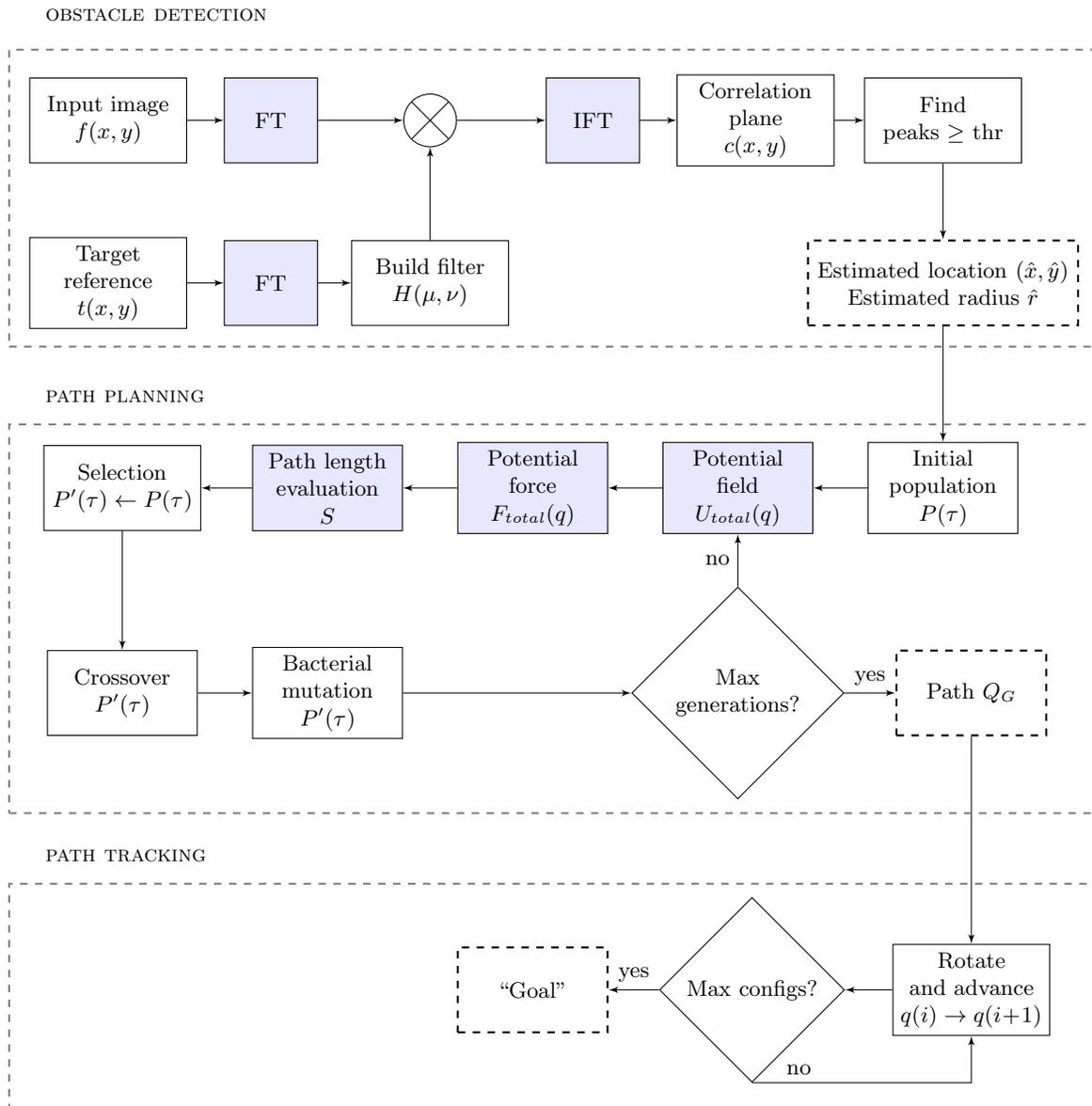


Figure 2. Proposed procedure for autonomous mobile robot navigation

3. PROPOSAL

The proposal to achieve the mobile robot navigation using visual environment recognition is presented in Fig. 2. The proposed methodology consists in the integration of three algorithms: obstacle detection, path planning and path tracking. These algorithms are explained as follows.

3.1 Obstacle detection

In the first section of Fig. 2, the obstacle detection procedure is shown. An input scene $f(x, y)$ is captured by the upper camera in order to monitor the environment, as described in Eq. 1. Assuming that the scene contains a feasible workspace $b(x, y)$ for robot navigation with J number of obstacles $t^j(x, y)$ presented in the environment.

In order to avoid collisions between the mobile robot and the obstacles, a correlation filtering approach is implemented to detect the obstacles. A correlation filter is build using Eq. 3, and the filter is synthesized with a

template reference. The template reference $t(x, y)$ contains the average matrix from all the images of the target centered at the origin. Also, the template contains average of the angle variation of the target with a set of $\{0, 20, \dots, 360\}$ degrees.

The cross correlation procedure between the input scene and the designed filter is computed. A correlation plane is obtained as follows:

$$c(x, y) = f(x, y) \otimes h(x, y), \quad (7)$$

where $h(x, y) = \text{IFT}\{H(\mu, \nu)\}$ represents the inverse Fourier transform of the GMF filter described in Eq. 3. The output correlation plane yields quantity levels of the best match. In this case, several peaks are produced in where the area of each obstacle coincides.

The estimated location coordinates (x_0^j, y_0^j) of each detected obstacle are computed as

$$(\hat{x}_0, \hat{y}_0) = \underset{(x, y)}{\operatorname{argmax}} \{|c(x, y)|^2\}. \quad (8)$$

The computation of the location of each detected obstacle is needed, so to determine the location of the correlation peak a threshold constrain is needed. For the estimated radius of each obstacle detected, we compute the area A of using image binarization. The estimated radius is computed as

$$\hat{r} = \sqrt{A\pi}, \quad (9)$$

where A is the area of the circumference form the detected object, given in pixels, and computed as follows:

$$A = \sum_{x, y}^d w_a(x, y), \quad (10)$$

where d is the number of pixels of the estimated area of the target, and $w_a(x, y)$ is a binary matrix which equals the unity in the target support region, and zero otherwise.

3.2 Path Planning

In the second section of Fig. 2, the path planning procedure is shown. The PBPF algorithm is integrated by the PBGA (composed by the genetic operators: selection, crossover and bacterial mutation), and by the APF method with the potential field functions denoted in colored blocks. The PBPF for path planning starts with the creation of an initial random population $P(\tau)$ of bacteria, where each bacterium contains a pair of the proportional gains, one k_a and one k_r . The resultant path Q_G (solution) given by the PBPF will be composed by the optimal pair of the proportional gains found.

After that the initial population is created, the evaluation process starts. The fitness function evaluation contains three steps. First, the potential field $U_{total}(q)$ is computed using Eq. 4. Then, the potential force $F_{total}(q)$ is computed using Eq. 5. Last, the distance to travel by the robot is measured by the path length function

$$S = \sum_{i=0}^m \|q_{i+1} - q_i\|, \quad (11)$$

where m is the number of robot configurations to reach the goal position.

Once that the evaluation has been completed, we continue with the selection process, $P'(\tau) \leftarrow P(\tau)$. In the selection process, the best bacteria are chosen according to their fitness value. The selection process drives the PBPF to improve the population of bacteria fitness over the successive generations.¹⁶ Next, the crossover process is performed over the $P'(\tau)$. The crossover process roughly mimics biological recombination between two single-chromosomes organisms.¹⁷ Then, the bacterial mutation process is performed over the $P'(\tau)$, where random mutations alter a certain percentage of the bits in the list of bacteria. The bacterial mutation operator is inspired by the biological model of bacterial cells; it makes that the PBGA mimics the phenomenon of microbial evolution.¹³ To find the global optimum; it is necessary to explore different regions in the search space that

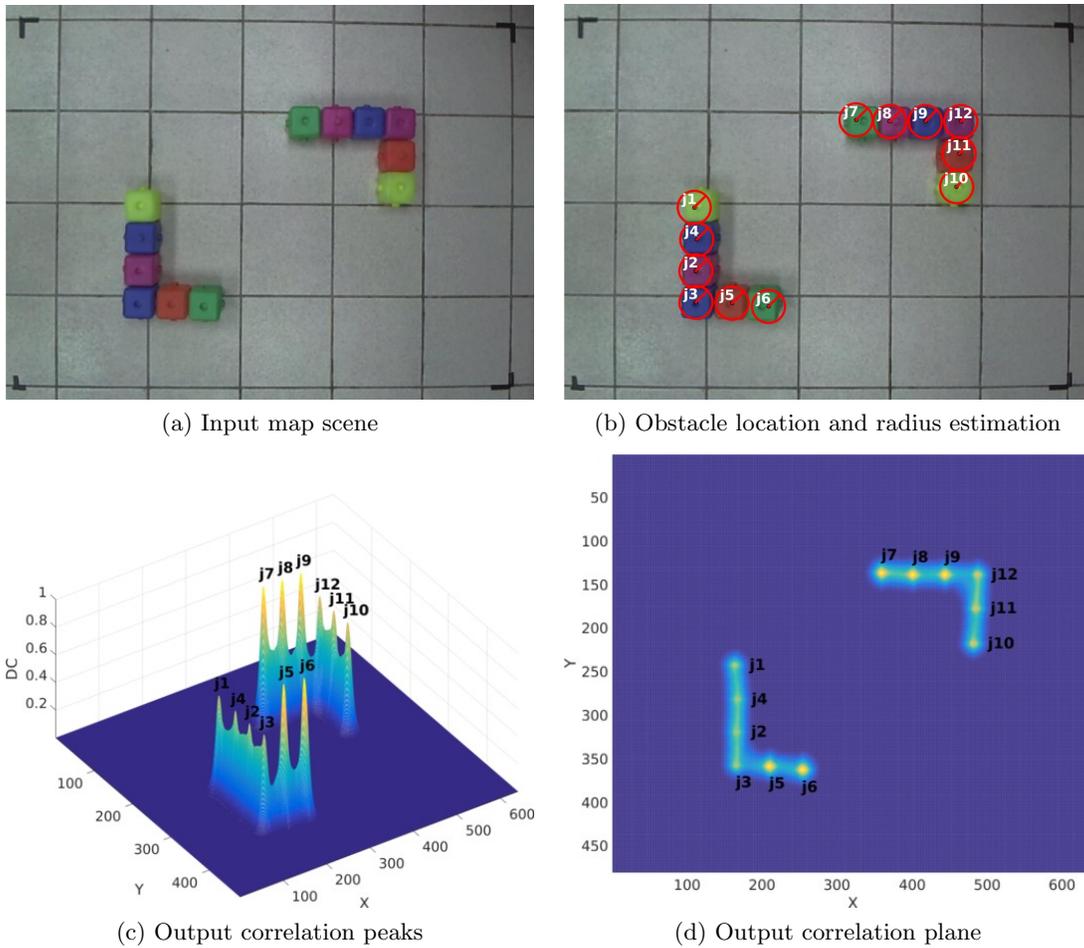


Figure 3. Obstacle detection from an input scene using correlation filtering

have not been covered by the current population of bacteria. This is achieved by adding new information to the bacteria; the information is generated randomly by the bacterial mutation operator applied to all the bacteria, one by one.

The PBPF iterates until the maximum number of generations is reached. Therefore, the PBPF evolves the proportional gains k_a and k_r to obtain the corresponding optimal values to perform the path planning. Finally, the PBPF gives the path Q_G found.

3.3 Path Tracking

In the third section of Fig. 2, the path tracking procedure is shown. The path Q_G obtained by the PBPF is a sequence of ordered objective points from the start to the goal position. This sequence of ordered objective points is converted to rotate and advance motion commands to let the robot moves from $q(i)$ to $q(i + 1)$ until the last configuration is reached, i.e., the goal position.

4. EXPERIMENTS AND RESULTS

The experimental results for visual environment recognition and path planning are presented in this section. The proposed system to achieve the mobile robot navigation using visual environment recognition that is shown in Fig. 2 was implemented on a computer with the Intel Core i7-4770, with a webcam Logitech C920HD for environment recognition. The proposed system was programmed in Matlab 2015a on Ubuntu 14.04.

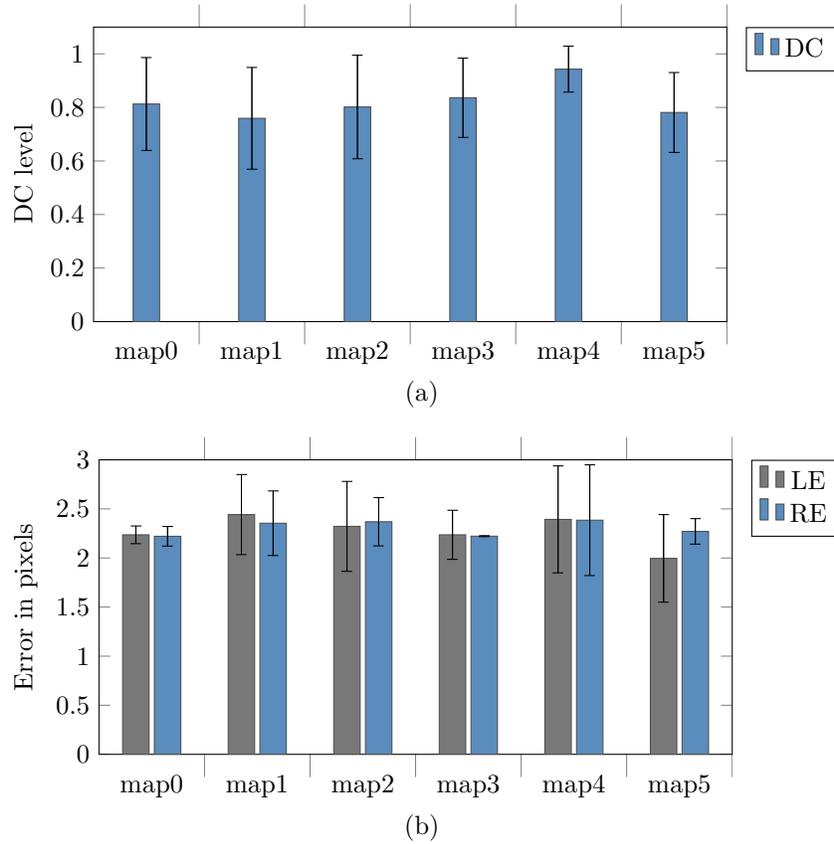


Figure 4. Obstacle detection in terms of (a) DC metric, and (b) location and radius error

The optical setup described by Fig. 1 was built in a real scenario. This scene represents $f(x, y)$ from the signal model depicted in Eq. 2. Here, the camera captures a scene used as visual environment for obstacle detection, see Fig. 3. As it can be observed, the input scene shows the map with a complex setup in order to solve obstacle detection with several challenges. Six different configurations of the environment for mobile robot navigation were tested.

We evaluated the performance of the visual environment recognition algorithm using quantitative metrics. To quantify the accuracy of obstacle detection, we tested the quality of the correlation filter. For this, we employed the discrimination capability (DC) of the correlation procedure. The DC measures the ability of a filter to recognize a target among false artifacts, given by

$$DC = 1 - \frac{|c^b|^2}{|c^t|^2}, \quad (12)$$

where c^b and c^t is the maximum value produced in the area of the background and the target, respectively. The evaluation of the proposed algorithm in terms of DC is shown in Fig. 4(a). The algorithm yields an overall performance of $DC = 0.822 \pm 0.157$ using the entire map set shown Fig. 5. A quantitative evaluation for obstacle detection is established. The accuracy of the detection of each obstacle is measured with the location error and the radius error. The location error (LE) is a measure for comparing the real and estimated values, described by

$$LE = \frac{1}{J} \sum_{j=1}^J \sqrt{(x_0^j - \hat{x}_0^j)^2 + (y_0^j - \hat{y}_0^j)^2}, \quad (13)$$

where (x_0^j, y_0^j) and $(\hat{x}_0^j, \hat{y}_0^j)$ are the real and estimated coordinates of all the obstacles presented in input scene.

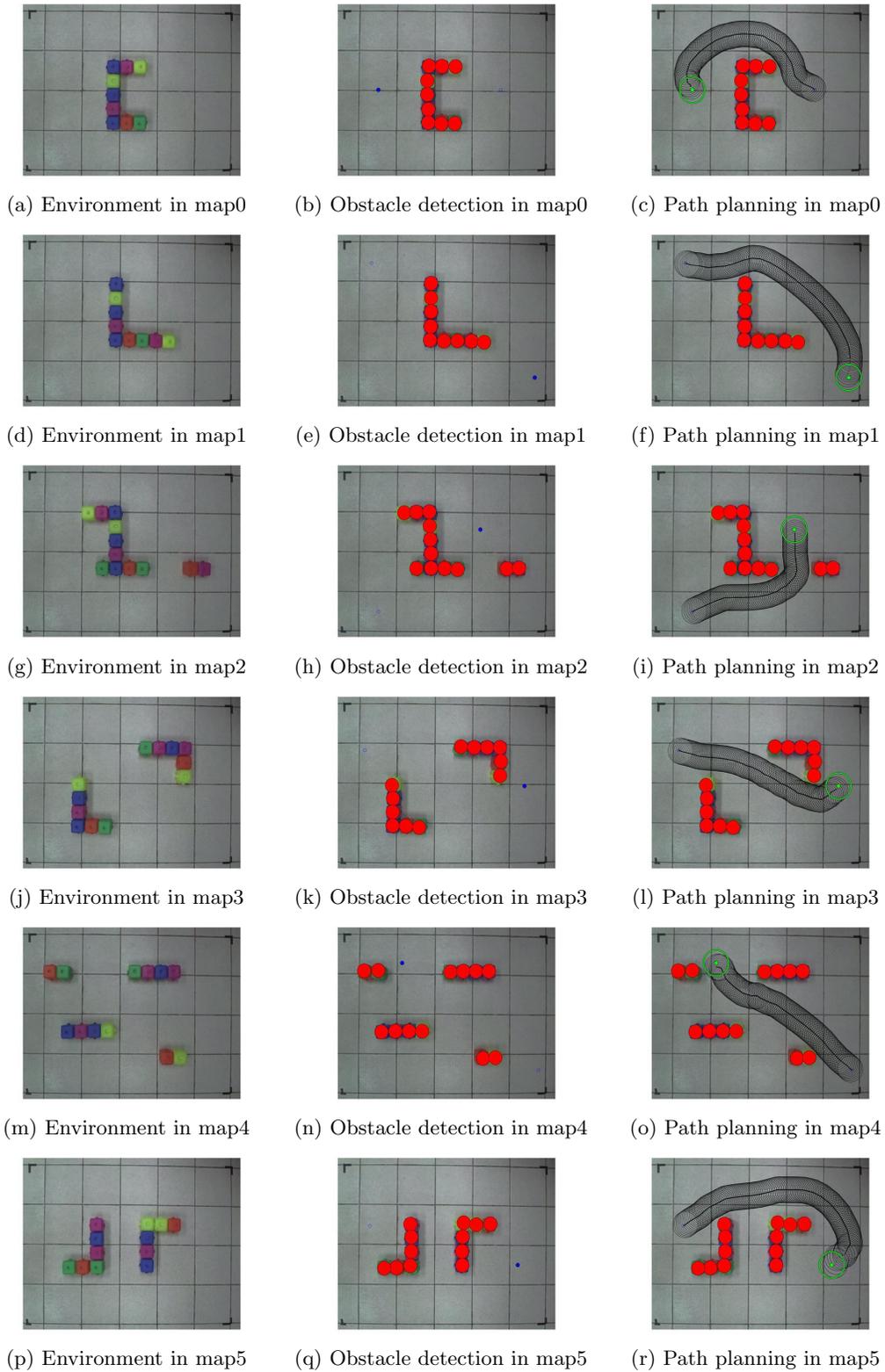


Figure 5. Environments used for the experiments, obstacle detection and path planning for a feasible and safe mobile robot navigation

Table 1. Visual environment recognition, position and radius of the obstacles detected in each environment

Environment	Obstacle $j(x, y, r)$ in pixels	Obstacle $j(x, y, r)$ in meters
map0 num. of obstacles detected: 9	(262, 215, 19.779), (262, 254, 19.779), (263, 174, 19.779), (265, 296, 19.779), (265, 336, 19.779), (305, 175, 19.779), (308, 336, 19.779), (348, 176, 19.779), (352, 335, 19.779).	(0.712, 0.585, 0.054), (0.712, 0.691, 0.054), (0.715, 0.473, 0.054), (0.720, 0.805, 0.054), (0.720, 0.914, 0.054), (0.829, 0.476, 0.054), (0.837, 0.914, 0.054), (0.946, 0.479, 0.054), (0.957, 0.911, 0.054).
map1 num. of obstacles detected: 9	(269, 296, 19.763), (272, 258, 19.779), (273, 137, 19.779), (273, 178, 19.779), (273, 216, 19.779), (311, 299, 19.617), (351, 298, 19.763), (391, 299, 18.780), (429, 302, 19.779).	(0.731, 0.805, 0.054), (0.740, 0.701, 0.054), (0.742, 0.372, 0.054), (0.742, 0.484, 0.054), (0.742, 0.587, 0.054), (0.846, 0.813, 0.053), (0.954, 0.810, 0.054), (1.063, 0.813, 0.051), (1.166, 0.821, 0.054).
map2 num. of obstacles detected: 12	(188, 134, 19.779), (230, 132, 19.779), (230, 290, 19.779), (269, 171, 19.779), (271, 209, 19.779), (273, 131, 19.779), (273, 248, 19.779), (273, 289, 19.779), (317, 290, 19.779), (360, 291, 19.779), (496, 291, 19.779), (538, 289, 19.779).	(0.511, 0.364, 0.054), (0.625, 0.359, 0.054), (0.625, 0.788, 0.054), (0.731, 0.465, 0.054), (0.737, 0.568, 0.054), (0.742, 0.356, 0.054), (0.742, 0.674, 0.054), (0.742, 0.786, 0.054), (0.862, 0.788, 0.054), (0.979, 0.791, 0.054), (1.349, 0.791, 0.054), (1.463, 0.786, 0.054).
map3 num. of obstacles detected: 12	(158, 247, 19.779), (160, 324, 19.779), (160, 362, 19.779), (161, 286, 19.779), (204, 363, 19.779), (248, 367, 19.779), (354, 141, 19.763), (395, 143, 19.779), (438, 143, 19.779), (476, 222, 19.779), (479, 182, 19.779), (481, 143, 19.779).	(0.430, 0.672, 0.054), (0.435, 0.881, 0.054), (0.435, 0.984, 0.054), (0.438, 0.778, 0.054), (0.555, 0.987, 0.054), (0.674, 0.998, 0.054), (0.962, 0.383, 0.054), (1.074, 0.389, 0.054), (1.191, 0.389, 0.054), (1.294, 0.604, 0.054), (1.302, 0.495, 0.054), (1.308, 0.389, 0.054).
map4 num. of obstacles detected: 12	(76, 123, 19.779), (118, 122, 19.779), (127, 294, 19.779), (170, 292, 19.779), (212, 292, 19.779), (254, 292, 19.779), (326, 126, 19.779), (368, 124, 19.763), (407, 124, 17.823), (426, 368, 19.779), (443, 124, 19.779), (468, 366, 19.779).	(0.207, 0.334, 0.054), (0.321, 0.332, 0.054), (0.345, 0.799, 0.054), (0.462, 0.794, 0.054), (0.576, 0.794, 0.054), (0.691, 0.794, 0.054), (0.886, 0.343, 0.054), (1.001, 0.337, 0.054), (1.107, 0.337, 0.048), (1.158, 1.001, 0.054), (1.204, 0.337, 0.054), (1.272, 0.995, 0.054).
map5 num. of obstacles detected: 12	(127, 310, 19.779), (170, 309, 19.348), (212, 306, 19.779), (213, 181, 19.779), (215, 223, 19.779), (215, 265, 19.609), (364, 263, 19.779), (365, 303, 19.779), (366, 222, 19.779), (370, 182, 19.779), (413, 185, 19.779), (456, 186, 19.779).	(0.345, 0.843, 0.054), (0.462, 0.840, 0.053), (0.576, 0.832, 0.054), (0.579, 0.492, 0.054), (0.585, 0.606, 0.054), (0.585, 0.720, 0.053), (0.990, 0.715, 0.054), (0.992, 0.824, 0.054), (0.995, 0.604, 0.054), (1.006, 0.495, 0.054), (1.123, 0.503, 0.054), (1.240, 0.506, 0.054).

Fig. 4(b) presents the evaluation of the proposed algorithm in terms of LE. The proposed algorithm yields an accuracy of $LE=2.27 \pm 0.37$ pixels for the tested map set. Also, the radius error (RE) compares the real and estimated radius in pixels of the obstacle in the scene. The RE is given by

$$RE = \frac{1}{J} \sum_{j=1}^J |r^j - \hat{r}^j|, \quad (14)$$

where r^j and \hat{r}^j are the real and estimated radii. As we can see in Fig. 4(b) the proposed algorithm yields an accuracy of $RE=2.31 \pm 0.23$ pixels.

Table 1 shows the position and radius of the obstacles detected in each test environment. The first column indicates the environment and the number of obstacles detected by the proposed obstacle detection algorithm, described in the first section of Fig. 2. The second column contains the information of each obstacle in pixels, the information is ordered in the coordinates x and y , and its radius r . We are considering a workspace of

Table 2. Robot mission (start and goal position), resultant proportional gains (best), and path planning results for each test environment. All data are expressed in meters except for the proportional gains

	map0	map1	map2	map3	map4	map5
Start	(1.305, 0.653)	(0.272, 0.218)	(0.326, 1.115)	(0.218, 0.408)	(1.604, 1.088)	(0.258, 0.517)
Goal	(0.326, 0.653)	(1.577, 1.088)	(1.142, 0.489)	(1.495, 0.680)	(0.517, 0.272)	(1.441, 0.816)
k_a	1.765	0.353	3.255	2.353	2.745	0.157
k_r	9.804	3.373	7.843	4.314	7.177	1.098
Best	1.465	1.819	1.228	1.365	1.381	1.834
Mean	1.552	1.860	1.247	1.386	1.396	2.012
Worst	1.811	1.928	1.323	1.442	1.466	2.396
Std. Dev.	0.092	0.028	0.019	0.019	0.015	0.171

640 × 480 pixels (images), where the coordinate (0,0) is at the upper left corner. The third column contains the information of the obstacles in meters (real world). It is the same information that the second column but converted to meters. So, we are considering a workspace of 1.74 × 1.29 meters and also the coordinate (0,0) is at the upper left corner.

Table 2 shows the robot mission assigned in each test environment, the mission is composed by a pair of coordinates (x, y) , one pair for the start position and one pair for the goal position. The Table 2 shows the best proportional gains k_a and k_r found by the PBPF algorithm to perform the path planning, described in the second section of Fig. 2. The resultant paths using the best proportional gains can be observed in Fig. 5 (c), (f), (i), (l), (o), and (r). On the figures of the resultant paths it can be observed how all the paths are safe (i.e., free of collisions) and in all cases the PBPF gives a smooth and effective path to drive the mobile robot to its goal. Table 2 also shows the statistical results for thirty independent tests in each environment. The shortest path length (best) found in meters, the average path length (mean), the worst path length found for each test environment (worst), and the standard deviation of the tests (std. dev.).

5. CONCLUSIONS

A proposal for visual environment recognition for robot path planning is presented. The proposal employs an upper camera for the recognition of the obstacles in order to establish a feasible workspace for the mobile robot. In order to solve obstacle detection, we propose a correlation filtering approach. The proposed algorithm for obstacle detection yields good estimation of the location and radii of several obstacles. Then a feasible workspace is built in order to plan an efficient and safe path for mobile robot navigation. We have seen through the results how the PBPF algorithm solves the path planning problem for the different environments exposed. The PBPF algorithm takes the information of the environment given by the obstacle detection algorithm, then the PBPF is able to perform the path planning. The resultant path is employed to perform the mobile robot navigation. The overall results prove the performance of the proposal in terms of obstacle detection accuracy and path planning efficiency.

ACKNOWLEDGMENTS

This research was supported in part by CETYS Universidad, *Consejo Nacional de Ciencia y Tecnología* (CONACYT), and *Secretaría de Investigación y Posgrado* from Instituto Politécnico Nacional projects SIP-20171387 and SIP-20170645.

REFERENCES

- [1] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D., [*Introduction to Autonomous Mobile Robots. Second Edition*], The MIT Press, Cambridge, Mass. (2011).
- [2] Orozco-Rosas, U., Montiel, O., and Sepúlveda, R., “Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation,” *International Journal of Advanced Robotic Systems* **12**(7), 1–14 (2015).
- [3] Diaz-Ramirez, V. H., Contreras, V., Kober, V., and Picos, K., “Real-time tracking of multiple objects using adaptive correlation filters with complex constraints,” *Opt. Commun.* **309**, 265–278 (2013).
- [4] Kumar, B. V. K. V. and Hassebrook, L., “Performance measures for correlation filters,” *Appl. Opt.* **29**(20), 2997–3006 (1990).
- [5] Kerekes, R. and Vijaya-Kumar, B., “Correlation filters with controlled scale response,” *IEEE Trans. Image Process.* **15**(7), 1794–1802 (2006).
- [6] Javidi, B. and Wang, J., “Design of filters to detect a noisy target in nonoverlapping background noise,” *J. Opt. Soc. Am. A* **11**, 2604–2612 (1994).
- [7] Diaz-Ramirez, V. H., Picos, K., and Kober, V., “Target tracking in nonuniform illumination conditions using locally adaptive correlation filters,” *Opt. Commun.* **323**, 32–43 (2014).
- [8] Kober, V. and Campos, J., “Accuracy of location measurement of a noisy target in a nonoverlapping background,” *J. Opt. Soc. Am. A* **13**(8), 1653–1666 (1996).
- [9] Picos, K., Diaz-Ramirez, V. H., Kober, V., Montemayor, A. S., and Pantrigo, J. J., “Accurate three-dimensional pose recognition from monocular images using template matched filtering,” *Opt. Eng.* **55**(6), 1–11 (2016).
- [10] Khatib, O., “Real-time obstacle avoidance for manipulators and mobile robots,” in [*Proceedings of the IEEE International Conference on Robotics and Automation*], **2**, 500–505 (1985).
- [11] Nawa, N. E., Hashiyama, T., Furuhashi, T., and Uchikawa, Y., “A study on fuzzy rules discovery using pseudo-bacterial genetic algorithm with adaptive operator,” in [*IEEE International Conference on Evolutionary Computation*], 589–593 (1997).
- [12] Orozco-Rosas, U., Montiel, O., and Sepúlveda, R., [*An Optimized GPU Implementation for a Path Planning Algorithm Based on Parallel Pseudo-bacterial Potential Field*], vol. 667 of *Studies in Computational Intelligence*, ch. 31, 477–492, Springer International Publishing (2017).
- [13] Botzheim, J., Gál, L., and Kóczy, L. T., [*Fuzzy Rule Base Model Identification by Bacterial Memetic Algorithms*], 21–43, Springer Berlin Heidelberg (2009).
- [14] Gál, L., Kóczy, L. T., and Lovassy, R., “Three step bacterial memetic algorithm,” in [*2010 IEEE 14th International Conference on Intelligent Engineering Systems*], 31–36 (2010).
- [15] Fogel, D. B., [*Evolutionary Computation: The Fossil Record*], ch. An Introduction to Evolutionary Computation, 1–28, Wiley-IEEE Press (1998).
- [16] Sivanandam, S. N. and Deepa, S. N., [*Introduction to Genetic Algorithms*], Springer, Heidelberg (2008).
- [17] Mitchell, M., [*An introduction to Genetic Algorithms*], Bradford, Cambridge, Massachusetts, USA (2001).