

Research Article

A Method for Estimating View Transformations from Image Correspondences Based on the Harmony Search Algorithm

Erik Cuevas¹ and Margarita Díaz²

¹*Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI, Avenida Revolución 1500, 44430 Guadalajara, JAL, Mexico*

²*División de Ciencia y Tecnología, Universidad de Guadalajara, CU-Norte, Carretera Federal No. 23, Km. 191, 46200 Colotlán, JAL, Mexico*

Correspondence should be addressed to Erik Cuevas; erik.cuevas@cucei.udg.mx

Received 30 September 2014; Accepted 12 December 2014

Academic Editor: Rahib H. Abiyev

Copyright © 2015 E. Cuevas and M. Díaz. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a new method for robustly estimating multiple view relations from point correspondences is presented. The approach combines the popular random sampling consensus (RANSAC) algorithm and the evolutionary method harmony search (HS). With this combination, the proposed method adopts a different sampling strategy than RANSAC to generate putative solutions. Under the new mechanism, at each iteration, new candidate solutions are built taking into account the quality of the models generated by previous candidate solutions, rather than purely random as it is the case of RANSAC. The rules for the generation of candidate solutions (samples) are motivated by the improvisation process that occurs when a musician searches for a better state of harmony. As a result, the proposed approach can substantially reduce the number of iterations still preserving the robust capabilities of RANSAC. The method is generic and its use is illustrated by the estimation of homographies, considering synthetic and real images. Additionally, in order to demonstrate the performance of the proposed approach within a real engineering application, it is employed to solve the problem of position estimation in a humanoid robot. Experimental results validate the efficiency of the proposed method in terms of accuracy, speed, and robustness.

1. Introduction

The goal of estimating geometric relations in images is to find an appropriate global transformation to overlay images of the same scene taken at different viewpoints. It can be applied in image processing when an object moves in front of a static camera and when a static scene is captured by a moving camera or multiple cameras from different viewpoints. This methodology has been widely adopted in many applications, for instance, when series of images can be stitched together to generate a panorama image [1–3]. Also, multiple image superresolution approaches can be applied in the overlapped region calculated according to the estimated geometry [4–6]. The motion of a moving object can also be estimated using its geometric relations [7] and a distributed camera network can be calibrated, where each camera's position, orientation, and focal length can be calculated based on

their correspondences [8–10]. Another example is the robot position that can be controlled or estimated through the estimation of the fundamental matrix/homography [11–13].

In a modelling problem, those data that can be explained by the hypothetical model are known as *inliers* of this model. Other points, for example, those generated by matching errors, are called *outliers*. The outliers are caused by external effects not related to the investigated model. Based on different criteria, several robust techniques have been proposed to identify points as inliers or outliers, being the random sampling consensus (RANSAC) algorithm [14] the most well known [15–17].

RANSAC adopts a simple hypothesize-and-evaluation process. Under such approach, a minimal subset of elements (correspondences) is sampled randomly, and a candidate model is hypothesized using this subset. Then, the candidate model is evaluated on the entire dataset separating all

elements from the dataset into inliers and outliers, according to their degree of matching (error scale) to the candidate model. These steps are iterated until there is a high probability that an accurate model could be found during iterations. The model with the largest number of inliers is considered as the estimation result.

Although RANSAC algorithm is simple and powerful, it presents two main problems [18, 19]: the high consumption of iterations and the inflexible definition of its objective function. In the RANSAC algorithm, candidate models are generated by selecting data samples. Since such a strategy is completely random, a large number of iterations are required to explore a representative subset of noisy data and to find a reliable model that could contain the maximum number of inliers. In general terms, the number of iterations is strongly affected by the contamination level of the dataset. The other crucial issue is the objective function to evaluate the correctness of a candidate model from contaminated data. In the RANSAC methodology, the best estimation result is the model that maximizes the number of inliers. Therefore, the objective function involves the count, one by one, of the number of inliers associated with a candidate model. Such an objective function is fixed and prone to obtain suboptimal models under different circumstances [19].

Several variants have been proposed in order to enhance the performance of the RANSAC method. One example constitutes the approach MLESAC [20] which searches the best hypothesis by maximizing the likelihood via the RANSAC process by assuming that the inlier data would distribute as a Gaussian function and outliers are distributed randomly. Alternatively, instead of giving the error scale (i.e., the threshold to separate inliers from outliers) a priori, the SIMFIT method [21] proposes its prediction based on an iterative procedure. Other representative works, such as the projection-pursuit method [22] and TSSE (two-step scale estimator) [23], employ the mean shift technique to model the inlier distribution and obtain an inlier scale. Such approaches enables RANSAC to be data-driven; however, the whole process becomes quite time consuming.

Although all the proposed variants allow solving one of the two main RANSAC problems, the other challenge still remains. Such situation comes from the fact that the estimation process is approached as an optimization problem where the search strategy is a random walking algorithm while the objective function is fixed to the number of inliers associated with the candidate model. In order to overcome the typical RANSAC problems, we propose to visualize the RANSAC operation as a generic optimization procedure. Under this point of view, a new efficient search strategy can be added for reducing the number of consumed iterations. Likewise, it can be defined as a new objective function which incorporates other elements that allow an accurate evaluation of the quality of a candidate model.

Two important difficulties in selecting a search strategy for RANSAC are the high multimodality and the complex characteristics of the estimation process produced by the elevated contamination of the dataset. Under such circumstances, classical methods present a bad performance [24, 25], making way for recent new approaches that have

been proposed to solve complex and ill-posed engineering problems. These methods include the application of modern optimization techniques such as evolutionary algorithms and metaheuristic techniques [26, 27] which have delivered better solutions over those obtained by classical methods.

The harmony search algorithm (HS) introduced by Geem et al. [28] is one example of these approaches. HS is an optimization algorithm based on the metaphor of the improvisation process that occurs when a musician searches for a better state of harmony. The HS produces a new candidate solution from all existing solutions. In HS, the solution vector is analogous to the harmony in music, and its generation schemes are analogous to musician's improvisations. With regard to other metaheuristics in the literature, HS imposes fewer mathematical prerequisites; therefore, it can be easily modified for solving several sorts of engineering optimization challenges [29, 30]. Numerical comparisons have established that the convergence of HS is faster than GA [29, 31, 32]. Such a fact has attracted the attention of the evolutionary computation community. It has been effectively applied to solve a wide range of practical optimization problems such as structural optimization [33], parameter estimation of the nonlinear Muskingum model [34], design optimization of water distribution networks [35], vehicle routing [36], image segmentation [37], and circle detection in images [38].

Although HS allows identifying promising regions at the solution space within a reasonable time interval, it underperforms in local searching, in particular for parameter identification applications [39–42]. In order to enhance the fine-tuning (accuracy) properties of HS, the local search parameter (BW) is dynamically adjusted to improve the balance between exploration and exploitation during the search process (see [29]). However, considering that the adjustment follows an exponential function, longer exploitation periods are allowed, affecting the exploring capacity of HS particularly when it is applied to complex objective functions. A better adjustment alternative, which employs the use of a linear model, has been recently proposed in [43]. It presents better searching capacities than the approaches based on exponential functions. For this reason, such an approach is used in our method.

In this paper, a new method is presented for the robust estimation of multiple view relations from point correspondences. The approach combines the RANSAC method with the HS. Upon such combination, the proposed method adopts a different sampling strategy in comparison to RANSAC to generate putative solutions. Under the new mechanism, new candidate solutions are built iteratively by considering the quality of models generated by previous candidate solutions, rather than relying over a pure random selection as it is the case of RANSAC. Likewise, a more accurate objective function is incorporated to accurately evaluate the quality of a candidate model. As a result, the proposed approach can substantially reduce the number of iterations still preserving the robust capabilities of RANSAC. The method is generic and its use is illustrated by the estimation of homographies, considering synthetic and real images. Additionally, in order to demonstrate the performance of the proposed approach in a real engineering application, it is

employed to solve the problem of position estimation of a humanoid robot. Experimental results validate the efficiency of the proposed method in terms of accuracy, speed, and robustness.

The paper is organized as follows. Section 2 explains the problem of image matching considering multiple views. Section 3 introduces the fundamentals of the RANSAC method. Section 4 explains the harmony search algorithm while Section 5 presents the proposed approach. Section 6 exhibits the experimental set and its performance results. Section 7 exposes a robotic application of the proposed approach. Finally, Section 8 establishes final conclusions.

2. View Relations from Point Correspondences

The problem of image matching consists in finding a geometric transformation that maps one image of a scene to another image taken from a different point of view. To determine the correspondence among points, it is necessary to find corresponding points on both images. Such point pairs can be obtained as a result of applying an automatic algorithm of detection and matching [44, 45]. The detected points are described by vectors of parameters (descriptors), and frequently these parameters do not allow discriminating one point from another with complete certainty. As a result, an erroneous matching about the correspondence of points located on different parts of different images may emerge.

In this section the geometric relations of points between two views are discussed, considering the case of homography.

Assume that there is a collection of pairs of the corresponding points that are found on two images

$$\mathbf{U} = \{(\mathbf{x}_1, \mathbf{x}'_1), (\mathbf{x}_2, \mathbf{x}'_2), \dots, (\mathbf{x}_M, \mathbf{x}'_M)\}, \quad (1)$$

where $\mathbf{x}_i = (x_i, y_i, 1)^T$ and $\mathbf{x}'_i = (x'_i, y'_i, 1)^T$ are the positions of points in the first and second images, respectively.

Two perspective images can geometrically be linked through a plane Q of the scene by a homography $\mathbf{H} \in \mathcal{R}^{3 \times 3}$ (see Figure 1). This projective transformation \mathbf{H} relates corresponding points of the plane projected into two images by $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$ or $\mathbf{x}_i = \mathbf{H}^{-1}\mathbf{x}'_i$. The homography across two views can be computed by solving a linear system from a set of four point matches [46]. The quality of the estimated homography \mathbf{H} is evaluated by considering the distance between the position of the point calculated with the help of the matrix \mathbf{H} and the actually observed position. Therefore, the mismatch error EH_i^2 produced by the i -correspondence $(\mathbf{x}_i, \mathbf{x}'_i)$ is defined as the sum of squared distances from the points to their estimated positions:

$$EH_i^2 = [d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)]^2 + [d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)]^2, \quad (2)$$

where $\eta = d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)$ and $\eta' = d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)$ correspond to the errors produced in the first and second images, respectively.

Figure 2 shows the error evaluation process of \mathbf{H} for a particular example which involves five correspondences $\mathbf{U} = \{(\mathbf{x}_1, \mathbf{x}'_1), \dots, (\mathbf{x}_5, \mathbf{x}'_5)\}$ distributed in both views. In the example, the correspondence $(\mathbf{x}_3, \mathbf{x}'_3)$ presents a considerable

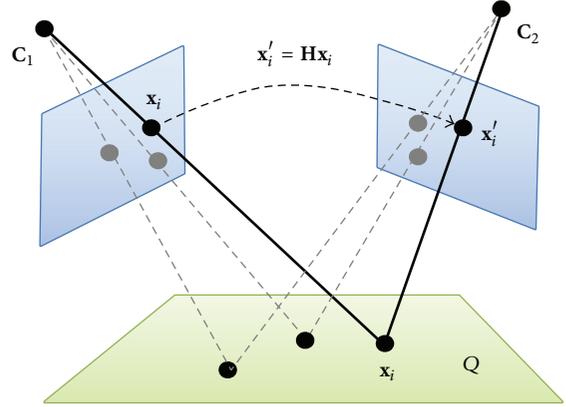


FIGURE 1: Homography from a plane between two views.

error evaluated as the distances (η, η') between the points $(\mathbf{x}_3, \mathbf{x}'_3)$ and their positions calculated with the use of the matrix \mathbf{H} ($\mathbf{H}\mathbf{x}_3, \mathbf{H}^{-1}\mathbf{x}'_3$).

3. Random Sampling Consensus (RANSAC) Algorithm

The goal of RANSAC is to estimate the geometric transformation (the homography \mathbf{H}) from image correspondences over two views. Potentially there are a significant number of mismatches amongst the correspondences. Correct matches will obey the homography transformation. Therefore, the aim is to obtain a set of *inliers* consistent with the homography transformation by using a robust technique. In this case *outliers* are points inconsistent with the homography transformation. In order to solve such a problem, the RANSAC algorithm has proven to be the most successful [15–17].

RANSAC solves the problem of model parameters estimation by finding the best hypothesis h^B among the set of all possible hypotheses H generated by the source data. Such source data are typically contaminated by noise. In order to build the hypothesis h_i about the unknown parameters, a sample S_i of the minimum size (s) required for model estimation is obtained (e.g., a sample of only two points is sufficient to calculate a straight line, $s = 2$, and of four to obtain a homography, $s = 4$). Under this consideration, the probability of finding an outlier is reduced. Considering that the number of elements contained in a sample is small, the amount of possible samples that can be generated from the complete source data \mathbf{U} is enormous. Under such circumstances, the exhausting testing of all samples for a reasonable time is impossible. RANSAC faces such problem because it only considers G samples which are randomly selected and evaluated. Algorithms of the RANSAC family consist of G iterations of the following cycle.

- (1) Construct a sample $S_i \subset \mathbf{U}$ consisting of s different elements.
- (2) Build the hypothesis h_i based on the sample S_i .
- (3) Evaluate the degree of agreement A_i of the hypothesis h_i with the set of all source data \mathbf{U} .

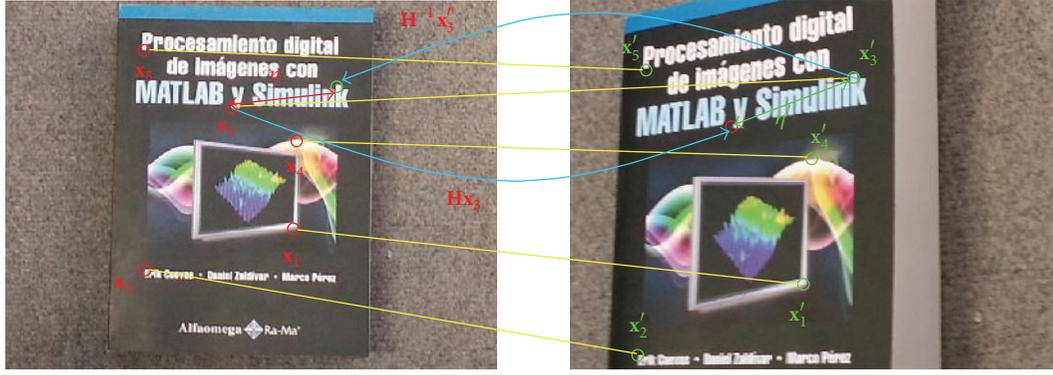


FIGURE 2: Example of evaluation process for a particular homography H .

After the construction and evaluation of all G hypotheses, the hypothesis h^B with the best degree of agreement is chosen among them. It is considered as a robust estimate of the model parameters. Such operation can be described as follows:

$$h^B = \arg \max_{i=1, \dots, G} A_i(\mathbf{U}, h_i). \quad (3)$$

The maximization of the degree of agreement (number of inliers) is equivalent to the minimization of the penalty function whose value depends on the number of outliers. Therefore, the degree of agreement $A_i(\mathbf{U}, h_i)$ is computed as follows:

$$A_i(\mathbf{U}, h_i) = \sum_{j=1}^M \theta(e_j^2(h_i)), \quad j = 1, \dots, M, \quad (4)$$

$$\theta(e_j^2(h_i)) = \begin{cases} 0 & e_j^2(h_i) > \text{Th} \\ 1 & e_j^2(h_i) \leq \text{Th}, \end{cases}$$

where Th is a permissible error, M is the number of elements contained in the source data \mathbf{U} , and $e_j^2(h_i)$ is the quadratic error produced by the j th data considering the hypothesis h_i . In the context of this paper, $e_j^2(h_i)$ corresponds to EH_i^2 which represents the error produced by the i th correspondence.

The hypothesis with a minimum penalty (i.e., with the maximum degree of agreement) is chosen as the best matching criterion. In the original scheme of RANSAC, the quality of a hypothesis is defined as the number of inliers. For a given value of the permissible error Th , the point j that produces the error $e_j^2(h_i)$ is regarded to be an inlier of h_i if its value does not exceed the threshold Th ; otherwise the point is regarded as an outlier.

In the RANSAC algorithm, the optimal hypothesis h^B is found and the penalty is minimized by using a search strategy of random walking; therefore many attempts are necessary to investigate in sufficient detail the space of possible samples and to find the sample for which the hypothesis has the greatest degree of agreement on the source data. The number of iterations and thus the time spent for the search can be reduced by choosing points according to some directed rules, rather than randomly. Optimization algorithms can be

considered as a robust scheme in contrast to the random search [47]. In an optimization algorithm, new candidate solutions are generated in accordance to the information obtained from past candidate solutions.

In this paper, we propose a different approach based on the HS as optimization algorithm. The goal is to demonstrate that the new method, by combining the idea of testing minimum-sized samples with the directed search inspired by the improvisation process that occurs when a musician searches for a better state of harmony, allows performing an efficient search among the correspondences to generate models of higher quality. It is also shown that the number of inliers found by the new method with the use of a fixed number of samples is significantly greater than the number of inliers determined by the family of algorithms based on RANSAC.

4. Harmony Search Algorithm

In the basic HS, each solution is called a “harmony” and is represented by an n -dimension real vector. An initial population of harmony vectors are randomly generated and stored within a harmony memory (HM). A new candidate harmony is thus generated from the elements in the HM by using a memory consideration operation either by a random reinitialization or a pitch adjustment operation. Finally, the HM is updated by comparing the new candidate harmony and the worst harmony vector in the HM. The worst harmony vector is replaced by the new candidate vector in case it is better than the worst harmony vector in the HM. The above process is repeated until a certain termination criterion is met. The basic HS algorithm consists of three basic phases: HM initialization, improvisation of new harmony vectors, and updating of the HM. The following discussion addresses details about each stage.

4.1. Initializing the Problem and Algorithm Parameters. In general, the global optimization problem can be summarized as follows: $\min f(\mathbf{p}) : p(j) \in [l(j), u(j)], j = 1, 2, \dots, n$, where $f(\mathbf{p})$ is the objective function, $\mathbf{p} = (p(1), p(2), \dots, p(n))$ is the set of design variables, n is the number of design variables, and $l(j)$ and $u(j)$ are the lower and upper bounds for the

design variable $p(j)$, respectively. The parameters for HS are the harmony memory size, that is, the number of solution vectors lying on the harmony memory (HM), the harmony-memory consideration rate (HMCR), the pitch adjusting rate (PAR), the distance bandwidth (BW), and the number of improvisations (NI) which represents the total number of iterations. It is obvious that an adequate selection for HS parameters would enhance the algorithm's ability to search for the global optimum under a high convergence rate.

4.2. Harmony Memory Initialization. In this stage, initial vector components at HM, that is, HMS vectors, are configured. Let $\mathbf{p}_i = \{p_i(1), p_i(2), \dots, p_i(n)\}$ represent the i th randomly generated harmony vector: $p_i(j) = l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1)$ for $j = 1, 2, \dots, n$ and $i = 1, 2, \dots, \text{HMS}$, where $\text{rand}(0, 1)$ is a uniform random number between 0 and 1. Then, the HM matrix is filled with the HMS harmony vectors as follows:

$$\text{HM} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{\text{HMS}} \end{bmatrix}. \quad (5)$$

4.3. Improvisation of New Harmony Vectors. In this phase, a new harmony vector \mathbf{p}_{new} is built by applying the following three operators: memory consideration, random reinitialization, and pitch adjustment. Generating a new harmony is known as "improvisation." In the memory consideration step, the value of the first decision variable $p_{\text{new}}(1)$ for the new vector is chosen randomly from any of the values already existing in the current HM, that is, from the set $\{p_1(1), p_2(1), \dots, p_{\text{HMS}}(1)\}$. For this operation, a uniform random number r_1 is generated within the range $[0, 1]$. If r_1 is less than HMCR, the decision variable $p_{\text{new}}(1)$ is generated through memory considerations; otherwise, $p_{\text{new}}(1)$ is obtained from a random reinitialization between the search bounds $[l(1), u(1)]$. Values of the other decision variables $p_{\text{new}}(2), p_{\text{new}}(3), \dots, p_{\text{new}}(n)$ are also chosen accordingly. Therefore, both operations, memory consideration and random reinitialization, can be modelled as follows:

$$p_{\text{new}}(j) = \begin{cases} p_i(j) \in \{p_1(j), p_2(j), \dots, p_{\text{HMS}}(j)\} \\ \quad \text{with probability HMCR} \\ l(j) + (u(j) - l(j)) \cdot \text{rand}(0, 1) \\ \quad \text{with probability } 1 - \text{HMCR}. \end{cases} \quad (6)$$

Every component obtained by memory consideration is further examined to determine whether it should be pitch-adjusted. For this operation, the pitch adjusting rate (PAR) is defined as to assign the frequency of the adjustment and the bandwidth factor (BW) to control the local search around the selected elements of the HM. Hence, the pitch adjusting decision is calculated as follows:

$$p_{\text{new}}(j) = \begin{cases} p_{\text{new}}(j) = p_{\text{new}}(j) \pm \text{rand}(0, 1) \cdot \text{BW} \\ \quad \text{with probability PAR} \\ p_{\text{new}}(j) \quad \text{with probability } (1 - \text{PAR}). \end{cases} \quad (7)$$

Pitch adjusting is responsible for generating new potential harmonies by slightly modifying original variable positions. Such operation can be considered similar to the mutation process in evolutionary algorithms. Therefore, the decision variable is either perturbed by a random number between $-\text{BW}$ and BW or left unaltered. In order to protect the pitch adjusting operation, it is important to assure that points lying outside the feasible range $[l, u]$ must be reassigned, that is, truncated to the maximum or minimum value of the interval.

4.4. Updating the Harmony Memory. After a new harmony vector \mathbf{p}_{new} is generated, the harmony memory is updated by the survival of the fit competition between \mathbf{p}_{new} and the worst harmony vector \mathbf{p}_w , according to its fitness value, in the HM. Therefore \mathbf{p}_{new} will replace \mathbf{p}_w and become a new member of the HM in case the fitness value of \mathbf{p}_{new} is better than the fitness value of \mathbf{p}_w .

4.5. Computational Procedure. The computational procedure of the basic HS can be summarized as shown in Procedure 1 [18].

4.6. Dynamical Linear Adjustment of BW. Every metaheuristic algorithm needs to address the issue of exploration-exploitation of the search space. Exploration is the process of visiting entirely new points of a search space whilst exploitation is the process of refining those points within the neighborhood of previously visited locations in order to improve their solution quality.

In HS, the BW parameter controls the local search around HM elements. A large BW value eases the algorithm's searching at a larger scope, while a small BW value is appropriate for fine-tuning of best solution vectors.

In the standard HS, the BW value is considered as a constant number. However, in this work, the BW value is dynamically adjusted as to favor exploration at early stages while exploitation is reinforced during final stages of the searching process. The adjustment uses a linear model defined as follows:

$$\text{BW}(k) = \begin{cases} \text{BW}_{\max} - \left(\frac{\text{BW}_{\max} - \text{BW}_{\min}}{2 \cdot \text{NI}} \right) \cdot 3k & \text{if } k < \left(\frac{2}{3} \right) \text{NI} \\ \text{BW}_{\min} & \text{if } k \geq \left(\frac{2}{3} \right) \text{NI}, \end{cases} \quad (8)$$

where k is the iteration index, while BW_{\max} and BW_{\min} are the maximum and minimum BW values, respectively. In contrast to exponential adjustment [26], linear models, as the one used in this paper, allow a better balance between exploration and exploitation (fine-tuning) of the search process [40].

Since all candidate solutions are generated by using the HS operators, there is a low probability to be trapped into local minima [48]. HS can effectively handle challenging multimodal optimization problems [49, 50]. Such fact contrasts to well-known genetic algorithms (GA) [51] and particle swarm optimization (PSO) [52] which usually tends

```

Step 1. Set the parameters HMS, HMCR, PAR, BW and NI.
Step 2. Initialize the HM and calculate the objective function value of each harmony vector.
Step 3. Improve a new harmony  $\mathbf{p}_{\text{new}}$  as follows:
    for ( $j = 1$  to  $n$ ) do
        if ( $r_1 < \text{HMCR}$ ) then
            Select randomly a number  $a$  where  $a \in (1, 2, \dots, \text{HMS})$ 
             $p_{\text{new}}(j) = p_a(j)$ 
            if ( $r_2 < \text{PAR}$ ) then
                 $p_{\text{new}}(j) = p_{\text{new}}(j) \pm r_3 \cdot \text{BW}$  where  $r_1, r_2, r_3 \in \text{rand}(0, 1)$ 
            end if
            if  $p_{\text{new}}(j) < l(j)$ 
                 $p_{\text{new}}(j) = l(j)$ 
            end if
            if  $p_{\text{new}}(j) > u(j)$ 
                 $p_{\text{new}}(j) = u(j)$ 
            end if
        else
             $p_{\text{new}}(j) = l(j) + r \cdot (u(j) - l(j))$ , where  $r \in \text{rand}(0, 1)$ 
        end if
    end for
Step 4. Update the HM as  $\mathbf{p}_w = \mathbf{p}_{\text{new}}$  if  $f(\mathbf{p}_{\text{new}}) > f(\mathbf{p}_w)$ 
Step 5. If NI is completed, the best harmony vector  $\mathbf{p}_b$  according to its fitness value in the HM is
    returned; otherwise go back to Step 3.

```

PROCEDURE 1

to conduct the whole population towards the best candidate solution [53] producing premature convergence.

5. Method for Geometric Estimation Using HS

The estimation of model parameters in algorithms of the RANSAC family is implied to find an optimal sample of length s from a set consisting of M elements. In the standard scheme, RANSAC uses a random walking algorithm as a search strategy. The idea of the proposed method considers the use of HS to generate samples based on information about their quality, rather than randomness. The quality of a sample, that is, the fitness of a harmony $f(\mathbf{p}_i)$, is defined as the matching degree of the hypothesis h_i that is constructed based on the correspondence numbers coded within \mathbf{p}_i .

Considering that the problem consists in estimating the parameters of \mathbf{H} through a set $\mathbf{U} = \{(\mathbf{x}_1, \mathbf{x}'_1), (\mathbf{x}_2, \mathbf{x}'_2), \dots, (\mathbf{x}_M, \mathbf{x}'_M)\}$ of M different correspondences, the proposed approach can be described as shown in Algorithm 1.

The proposed approach combines the RANSAC method with the HS adopting a different sampling strategy in comparison to RANSAC to generate putative solutions. Under the new mechanism, at each iteration, new candidate solutions are built taking into account the quality of the models that have been generated by previous candidate solutions, rather than purely random as it is the case in RANSAC.

Since the approach visualizes the RANSAC operation as a generic optimization procedure, different objective functions can be incorporated to accurately evaluate the quality of a candidate model. Although several objective functions can be tested, this work employs the expression in Equation (A).

In contrast to the traditional RANSAC algorithm, the objective function considers two different aims: the number of inliers and the approximation error. The idea is to find the candidate homography that maximizes the number of inliers and simultaneously minimizes the approximation error. Under such circumstances, the obtained estimation represents the solution that presents the best trade-off between both objectives. As a result, the proposed approach can substantially reduce the number of iterations, still preserving the robust capabilities of RANSAC method.

6. Experimental Results

In this section, a comprehensive set of experiments have been conducted to test the performance of the proposed approach. The results are divided into two different categories: (1) effect of the main HS parameters in the estimation results and (2) comparison results over synthetic and real homographies.

In the experiments, three performance indexes are considered: the number of inliers (NofI), the error (E_s, E_r), and the number of function evaluations (NFE). The first two indexes assess the accuracy of the solution whereas the last one measures the computational cost.

The number of inliers (NofI) expresses the amount of elements contained in the set \mathbf{I} of detected inliers. The error (E_s, E_r) provides a quality measure of the estimated relation. In case of synthetic data, the error is calculated as

$$E_s = \left(\sum_{ij} \frac{d^2(\mathbf{x}_i^j, \hat{\mathbf{x}}_i^j)}{\text{NofI}} \right)^{1/2}, \quad i \in \mathbf{I}, j \in \{1, 2\}, \quad (9)$$

(1) Configuration
 (a) Set the parameters HMS, HMCR, PAR, BW_{\min} , BW_{\max} and NI.

(2) Initial population.
 (a) Build the harmony memory (HM) $HM = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{HMS}\}$ where each individual \mathbf{p}_i consists 4 random non-repeating indices from 1 to M .
 (b) Compute homography \mathbf{H}_i (hypothesis h_i) by using the indices from \mathbf{p}_i .
 (c) Calculate the fitness value $f(\mathbf{p}_i)$ as the matching quality of the constructed homography \mathbf{H}_i considering the whole available data \mathbf{U} . Such fitness value is calculated by using a new objective function defined as:

$$F(\mathbf{a}_i(k)) = \sum_{j=1}^M \theta(e_j^2(h_i)) - \lambda \cdot e_j^2(h_i) \quad \theta(e_j^2(h_i)) = \begin{cases} 0 & e_j^2(h_i) > Th \\ 1 & e_j^2(h_i) \leq Th \end{cases}, \quad (A)$$

where $e_j^2(h_i)$ represents the quadratic error produced by the j th correspondence considering the hypothesis h_i whereas λ is the penalty associated with the mismatch magnitude. Such error corresponds to the mismatch EH_j^2 generated by the evaluation of \mathbf{H}_i .

(3) Iterations $k = 1, \dots, NI$.
 (a) Generate a new harmony \mathbf{p}_{new} (candidate solution) as follows:

$$BW(k) = \begin{cases} BW_{\max} - \left(\frac{BW_{\max} - BW_{\min}}{2 \cdot NI} \right) \cdot 3k & \text{if } k < \left(\frac{2}{3} \right) NI \\ BW_{\min} & \text{if } k \geq \left(\frac{2}{3} \right) NI \end{cases}$$
 for ($j = 1$ to n) do
 if ($r_1 < HMCR$) then
 Select randomly a number a where $a \in (1, 2, \dots, HMS)$
 $p_{\text{new}}(j) = p_a(j)$
 if ($r_2 < PAR$) then
 $p_{\text{new}}(j) = p_{\text{new}}(j) \pm r_3 \cdot BW(k)$ where $r_1, r_2, r_3 \in \text{rand}(0, 1)$
 end if
 if $p_{\text{new}}(j) < l(j)$
 $p_{\text{new}}(j) = l(j)$
 end if
 if $p_{\text{new}}(j) > u(j)$
 $p_{\text{new}}(j) = u(j)$
 end if
 else
 $p_{\text{new}}(j) = l(j) + r \cdot (u(j) - l(j))$, where $r \in \text{rand}(0, 1)$
 end if
 end for
 (b) Compute homography \mathbf{H}_i by using the indices from \mathbf{p}_{new} .
 (d) Calculate the fitness value $f(\mathbf{p}_{\text{new}})$ as the matching quality of the constructed homography \mathbf{H}_{new} considering the whole available data \mathbf{U} . Such fitness value is calculated by using the objective function described in (A).
 (e) Update the HM as $\mathbf{p}_w = \mathbf{p}_{\text{new}}$ if $f(\mathbf{p}_{\text{new}}) > f(\mathbf{p}_w)$

(4) Estimation result
 (a) The best estimation \mathbf{H}^B consists of the parameters computed by using the indices from the best element \mathbf{p}^B of HMin terms of its affinity, so that $\mathbf{p}^B = \text{argmax}_{i=1, \dots, HMS} f(\mathbf{p}_i)$.

ALGORITHM 1

where \mathbf{x}_i^j is the inlier point calculated by the estimated relation in the j -view, $\tilde{\mathbf{x}}_i^j$ is the inlier ground true point, and $d(\cdot)$ is the Euclidian distance between the points. Therefore, E_s evaluates the fit of the estimated relation, computed from the noisy data, against the known ground truth points.

In the case of real data the error is assessed from the standard deviation of the inliers. Thus, E_r is computed as follows:

$$E_r = \left(\sum_i \frac{e_i^2}{\text{Nofl}} \right)^{1/2}, \quad i \in \mathbf{I}, \quad (10)$$

TABLE 1: Effect of the HS parameters in the estimation process.

	(NofI, E_s)			
	HMCR = 0.5	HMCR = 0.6	HMCR = 0.7	HMCR = 0.8
PAR = 0.1	(21, 4.2147)	(26, 3.8124)	(29, 3.4721)	(27, 4.0112)
PAR = 0.2	(22, 4.1457)	(35, 2.0974)	(36, 2.1474)	(31, 3.3784)
PAR = 0.3	(21, 4.5714)	(38, 1.1124)	(40, 0.8514)	(35, 2.0053)
PAR = 0.3	(23, 4.0781)	(34, 2.0078)	(37, 2.0012)	(31, 3.4079)

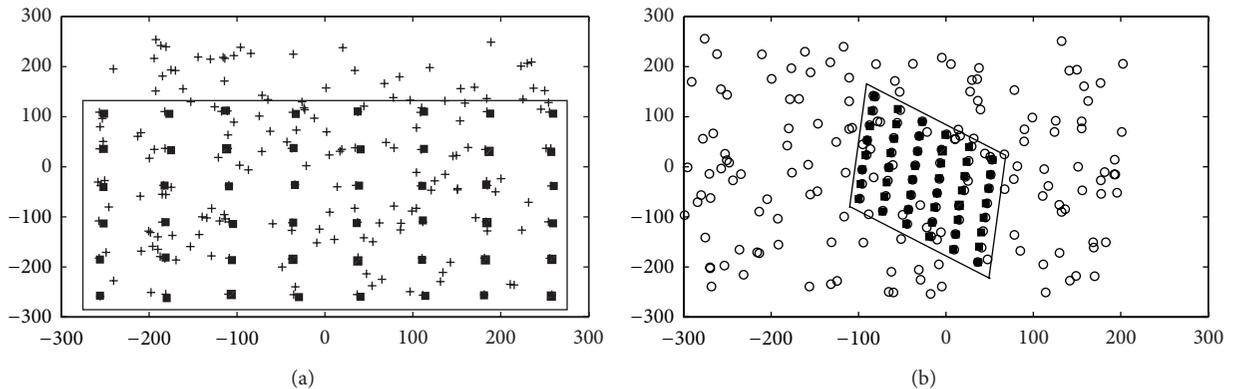


FIGURE 3: A test example where the HS-RANSAC has been applied to estimate a random transformation \mathbf{H} considering only the 75% of additional outliers. (a) The first view and (b) the second view, with black squares representing the detected inliers.

where e_i^2 is the quadratic error produced by the i th inlier. In the context of this paper, e_i^2 corresponds to EH_i^2 which represents the error produced by the i th inlier.

The number of function evaluations (NFE) specifies the total number of transformations that have been evaluated by the algorithm until the best estimation has been reached.

6.1. Effect of the HS Parameters. Several parameters define the performance of HS. However, from all of them, the harmony-memory consideration rate (HMCR) and pitch adjusting rate (PAR) are the most important [55]. To study the impact of these parameters, over the performance of HS in the estimation procedure, different values have been tested on the computation of a synthetic homography. Such a homography was generated, in its first view, by using a rectangular pattern of 8×6 elements within a 2-dimensional space of $[-300, 300]$. Then, such points were transformed by a random homography \mathbf{H} and contaminated by normally distributed noise for constructing their correspondences in the second view. A set of outliers was added by selecting randomly data points within the space limits. In the test, the fraction of outliers is of 75%. In order to illustrate the experimental setup, Figures 3(a) and 3(b) exhibit the first and second views, respectively. Considering the correspondence points, the HS-RANSAC algorithm generates the estimation of \mathbf{H} . In Figure 3(a), the black squares indicate the position in the first view of a point from the second view as a result of the \mathbf{H} transformation. Likewise, the black squares in Figure 3(b) exhibit the position in the second view of a point from the first view as a result of the \mathbf{H} transformation.

TABLE 2: HS-RANSAC estimator parameters.

HMS	HMCR	PAR	BW _{max}	BW _{min}	NI	λ	Th
50	0.7	0.3	10	1	950	0.001	5

In the experiment, the maximum number of iterations is set to 950. HMS, BW_{max}, BW_{min}, λ , and Th are fixed to 50, 10, 1, 0.001, and 5, respectively. The results report the number of inliers (NofI) and the produced estimation error (E_s) of HS-RANSAC, averaged over 30 runs, for the different values of HMCR and PAR. In the experiment, the parameter values are modified considering specific interval. HMCR varies from 0.5 to 0.8 whereas PAR changes from 0.1 to 0.4. The results, shown in Table 1, suggest that a proper combination of different parameter values can improve the performance of HS-RANSAC and the quality of the estimations. The best parameter configuration in the experiment is highlighted in Table 1.

After considering the analysis of Table 1, the parameter values for the proposed estimator are defined in Table 2. Once defined, such values have been kept in all experiments reported in this paper.

6.2. Comparison Results over Synthetic and Real Homographies. We have applied the proposed method to estimate homographies on real and synthetic data in order to compare its performance against other estimation algorithms such as the standard RANSAC [14], the MLESAC [20], the SIMFIT method [21], the projection-pursuit algorithm [22], the TSSE [23], and the PSO algorithm (PSO-RANSAC) [54]. The first

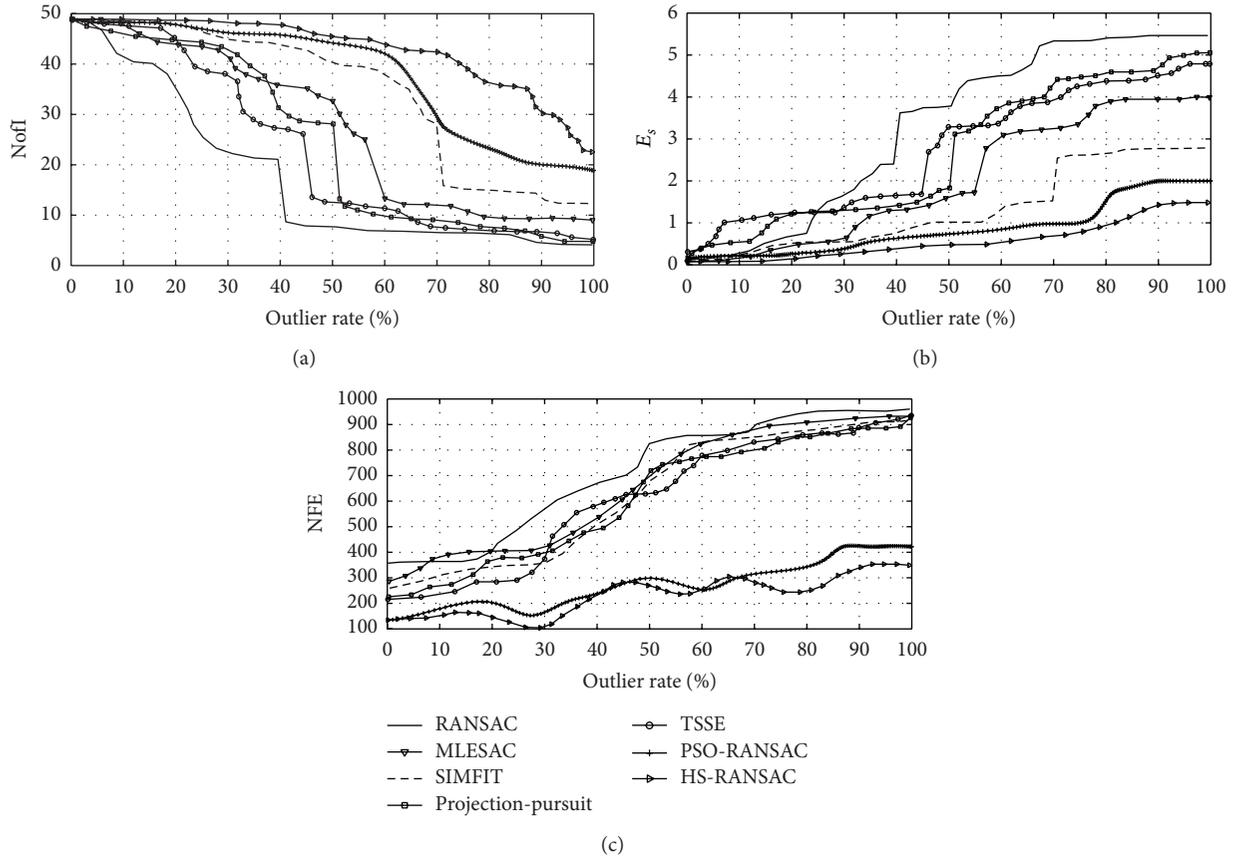


FIGURE 4: Experimental results corresponding to the estimation of \mathbf{H} considering synthetic data.

five approaches are RANSAC-based estimators whose results are broadly known. In all cases, the algorithms are tuned according to the value set which is originally proposed by their own references. However, the PSO method has been included as a reference, only to validate the performance of the HS as an optimization approach.

In order to conduct a fair comparison between the HS version used in this work and PSO, an enhanced version of PSO has been also chosen with similar characteristics. Therefore, it is used in the comparisons, the PSO version reported in [54]. Such an approach is proposed to mitigate the premature convergence problem of the original PSO method. It incorporates two new elements: (1) a weight factor w and (2) a constriction factor V_{\max} . Similar to BW in the HS method, the weight factor w is linearly decreased during the algorithm execution to regulate the attraction force towards the best particle seen so far. On the other hand, the constriction factor V_{\max} permits limiting the particle velocities in order to control their trajectories. Under such circumstances, the enhanced PSO version is used in combination with RANSAC considering the following configuration: $P = 10$, $c_1 = 2$, $c_2 = 2$, and $\text{Th} = 5$ whereas the weight factor w decreases linearly from 0.9 to 0.2. Additionally, the constriction factor V_{\max} is fixed to 2. Such a configuration presents the best possible performance according to [54].

6.2.1. Homography Estimation with Synthetic Data. This section reports the experimental results corresponding to the estimation of homography matrix considering synthetic data. In the experiments, the same synthetic homography produced in Section 4.1 has been used (see Figure 3). The only difference is that the fraction of the incorporated outliers varies from 0 to 100%.

In the experiment, each algorithm's execution requires 1000 iterations. Since the proposed HS-RANSAC involves 50 initial evaluations (size of the harmony memory), it requires the execution of only 950 iterations to reach the 1000 evaluations. On the other hand, PSO-RANSAC possesses 10 particles; for this reason 100 generations need to be evolved in order to fulfill the 1000 iterations.

Figure 4 presents the performance for each algorithm. The results present the averaged outcomes obtained throughout 50 different executions. In order to appropriately analyze these results, it is necessary to define the concept of a breakdown point [18]. The breakdown point is identified as the highest outlier ratio from which the algorithm degrades its capacity to find inliers. It can be seen from Figure 4(a) that standard RANSAC has a breakdown point at 40%, the MLESAC at 55%, the SIMFIT method at 70%, the projection-pursuit algorithm at 50%, the TSSE at 45%, and the PSO-RANSAC at 80%. In contrast to such methods, the proposed approach, HS-RANSAC, does not seem to have a prominent

TABLE 3: Inlier detection comparison in terms of the detection rate (DR), the error (E_r), and the number of function evaluations (NFE) for standard RANSAC [14], the MLESAC [20], the SIMFIT method [21], the projection-pursuit algorithm [22], the TSSE [23], the PSO algorithm (PSO-RANSAC) [54], and the proposed HS-RANSAC approach, considering the four test images shown in Figures 5, 6, 7, and 8.

Image	Method	Detected inliers (NofI)	Missing	False alarms	DR (%)	E_r	NFE
(A) Total number of inliers (86)	Standard RANSAC	41	45	21	47.7	4.75	876
	MLESAC	55	31	14	63.9	3.11	852
	SIMFIT	62	24	11	72.0	2.98	842
	Projection-pursuit	58	28	12	67.4	3.53	798
	TSSE	48	38	14	55.8	3.42	815
	PSO-RANSAC	75	11	8	87.2	1.68	491
	HS-RANSAC	82	4	5	95.3	0.88	396
(B) Total number of inliers (72)	Standard RANSAC	32	40	18	44.4	3.98	765
	MLESAC	40	32	14	55.5	3.43	825
	SIMFIT	58	14	8	80.5	2.87	891
	Projection-pursuit	47	25	12	65.2	3.12	759
	TSSE	43	29	16	59.7	3.47	786
	PSO-RANSAC	63	9	5	87.5	1.51	374
	HS-RANSAC	70	2	3	97.2	0.79	328
(C) Total number of inliers (56)	Standard RANSAC	24	32	15	42.8	2.96	689
	MLESAC	27	29	11	48.2	2.41	628
	SIMFIT	42	14	9	75.0	1.98	724
	Projection-pursuit	37	19	13	66.0	2.85	754
	TSSE	32	24	14	57.1	2.74	776
	PSO-RANSAC	48	8	9	85.7	0.94	349
	HS-RANSAC	53	3	5	94.6	0.25	272
(D) Total number of inliers (122)	Standard RANSAC	62	60	22	50.8	4.02	832
	MLESAC	77	45	18	63.1	3.41	924
	SIMFIT	90	32	13	73.7	2.86	845
	Projection-pursuit	75	47	19	61.4	3.52	914
	TSSE	76	46	21	62.2	3.73	887
	PSO-RANSAC	110	12	10	90.1	1.41	427
	HS-RANSAC	115	7	5	94.2	0.51	338

breakdown point, since its capacity to detect inliers smoothly degrades. It is also observed that the HS-RANSAC algorithm presents the best performance in terms of the number of inliers (NofI), as it is able to detect most of them. For the estimated \mathbf{H} , the error E_s (Figure 4(b)) is fairly comparable for all methods until they reach their breakdown points. Nonetheless, the proposed algorithm performed better, being the only algorithm that consistently found the minimum error at all outlier ratios.

In terms of number of function evaluations (NFE), Figure 4(c) shows that the standard RANSAC, the MLESAC, the projection-pursuit algorithm, and the TSSE invest approximately the same number of iterations for reaching their best estimation of \mathbf{H} . Since such methods use a random walking algorithm as a search strategy, the NFE significantly grows as the number of outliers increases. On the other hand, the PSO-RANSAC and the HS-RANSAC (that use an optimization algorithm as search strategy) maintain a considerably low NFE value with independence of the number of outliers.

From the experiment, it is evident that the use of an optimization approach can considerably reduce the NFE value. However, there is no optimization algorithm suitable to find a good enough estimation considering the high

multimodality and complex characteristics of the estimation process which is produced by the elevated contamination of the dataset. Therefore, although the PSO-RANSAC finds its best estimated fundamental matrix \mathbf{H} investing approximately the same number of evaluations as the HS-RANSAC, such estimated matrix represents only a suboptimal solution. This fact can be observed in Figure 4(b) where it is clear that the PSO-RANSAC algorithm presents higher E_s values in comparison to the HS-RANSAC approach. The reason for this problem points to those operators used by PSO for modifying the individual positions. In PSO, during their evolution, the position of each agent in the next iteration is updated yielding an attraction towards the position of the best particle seen so far. Such behavior shows that the entire population, as the algorithm evolves, concentrates around the best particle, favoring the premature convergence (reaching suboptimal solutions) [53].

6.2.2. Homography Estimation with Real Images. In this section, the experimental results of the estimation of homographies \mathbf{H} considering real images are reported. To evaluate the estimation performance of the proposed method, Table 3 tabulates the comparative inlier detection performance of the standard RANSAC [14], the MLESAC [20], the SIMFIT

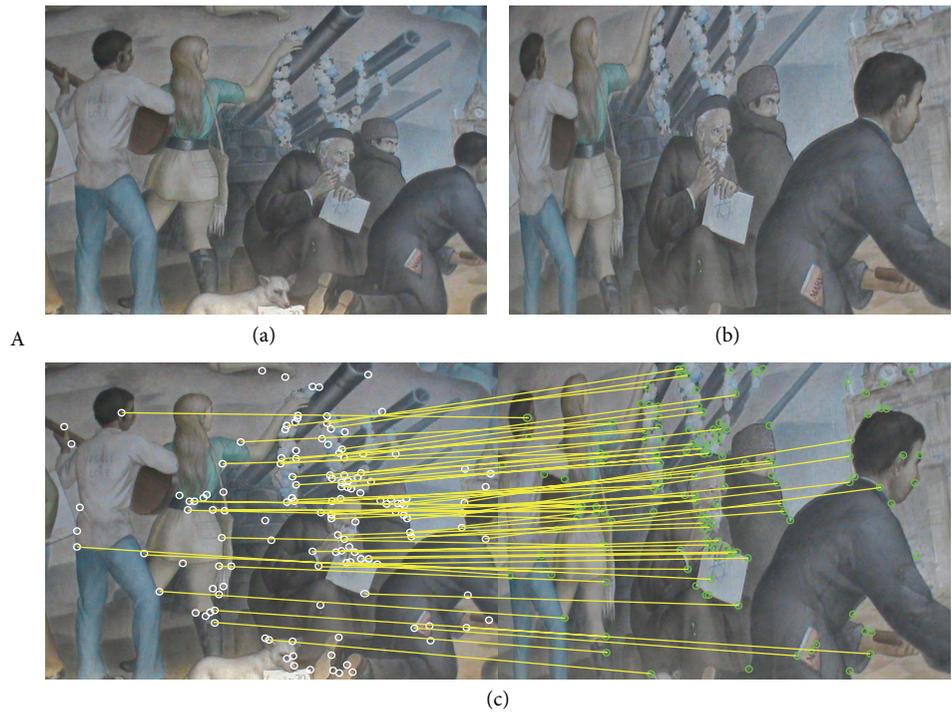


FIGURE 5: Test image “A”: (a) first view, (b) second view, and (c) correspondence points and inliers produced by HS-RANSAC.

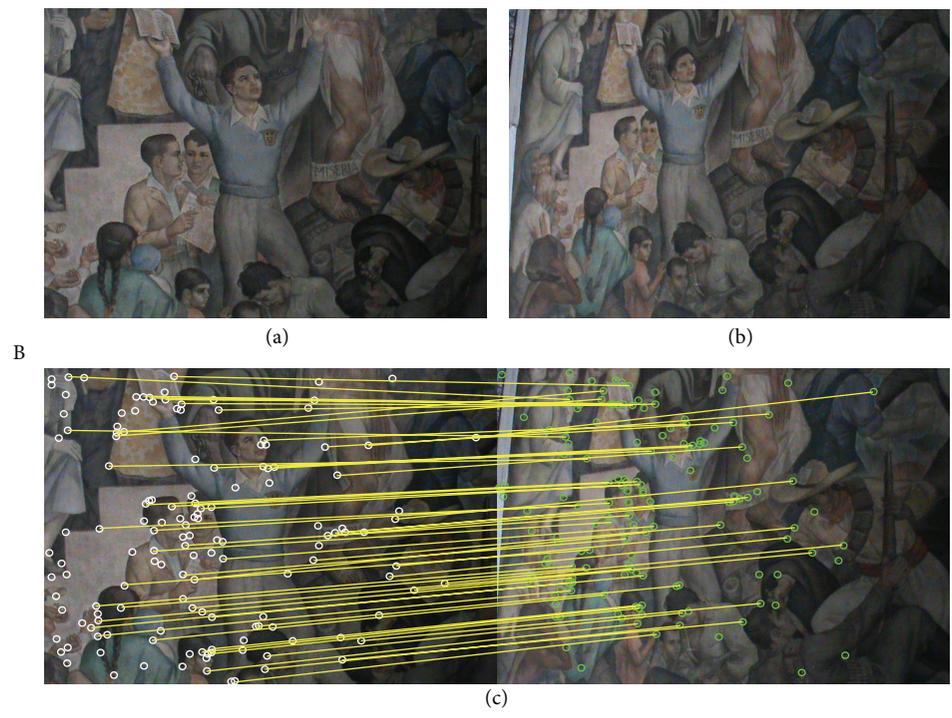


FIGURE 6: Test image “B”: (a) first view, (b) second view, and (c) correspondence points and inliers produced by HS-RANSAC.

method [21], the projection-pursuit algorithm [22], the TSSE [23], the PSO algorithm (PSO-RANSAC) [54], and the proposed HS-RANSAC approach, in terms of the detection rate (DR), the error (E_r), and the number of function evaluations (NFE). The experimental dataset includes 4 images (images

A, B, C, and D) which are shown in Figures 5, 6, 7, and 8. Such images contain a determined number of inliers which have been detected and counted by a human expert ($A = 86$, $B = 72$, $C = 56$, and $D = 122$). Such values act as ground truth for all the experiments. For the comparison, the detection rate



FIGURE 7: Test image “C”: (a) first view, (b) second view, and (c) correspondence points and inliers produced by HS-RANSAC.

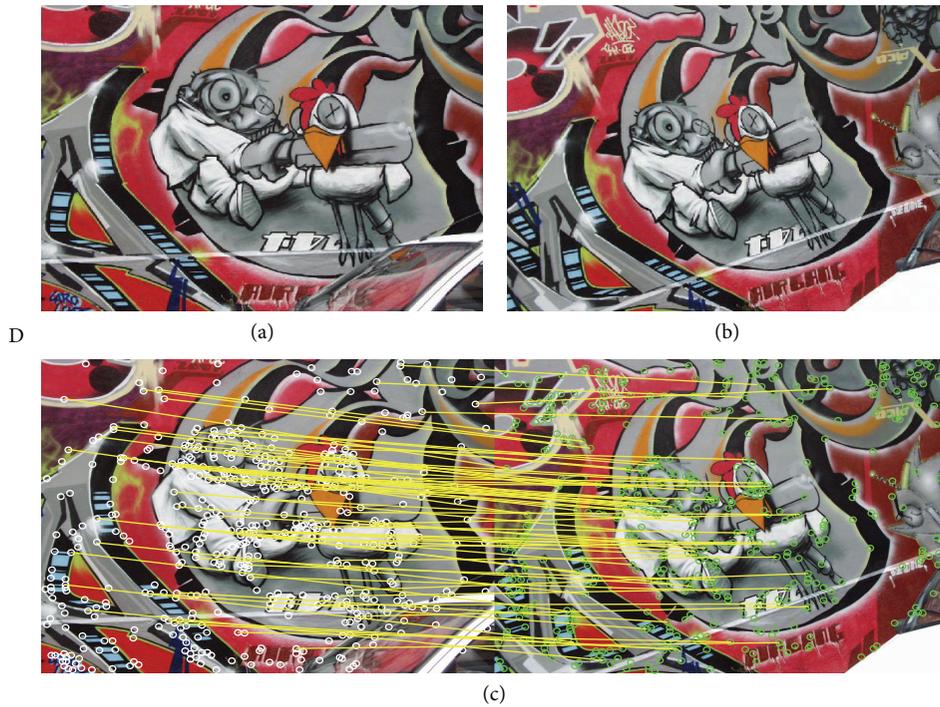


FIGURE 8: Test image “D”: (a) first view, (b) second view, and (c) correspondence points and inliers produced by HS-RANSAC.

(DR) is defined as the ratio between the number of inliers correctly detected by the algorithm (Nofl value) and the total number of inliers determined by the expert. The results consider 50 different executions for each algorithm over the four images. Experimental results show that the proposed

HS method accomplishes at least a 94.2% of inlier detection accuracy. A close inspection of Table 3 also reveals that the proposed approach is able to achieve the smallest error (E_r), yet requiring a few number of function evaluations (NFE) for most cases.

Figures 5, 6, 7, and 8 also exhibit the results after applying the HS-RANSAC estimator. Such results present the median case obtained throughout 50 runs.

7. Engineering Application: Position Estimation in a Humanoid Robot

Additionally, in order to demonstrate the performance of the proposed approach in a real engineering application, the paper also reports the application of the HS-RANSAC to solve the problem of position estimation of a humanoid robot.

In the last decades, much work has already been accomplished in the area of humanoid robotics [56, 57]. Position determination for humanoid robots is a critical problem, since it is used to control their balance and locomotion. Recently, a notable research [58] has been devoted to achieving better performance in system position for humanoid robots by using sensor fusion methods. In general, integrating information from different sensors increases not only the versatility of the system, but also its cost and complexity. Vision is one of the most studied sensory modalities for position and navigation purposes since it provides rich information of the environment.

The framework of the approach presented in this section, as an application, is a vision system consisting of a fixed camera mounted on a Bioloid© humanoid robot. In the approach, the position (x, y) of the robot is computed considering the homography estimated by the HS-RANSAC. Therefore, the idea is to calculate the planar motion of the humanoid robot through the estimated homographies. Figure 9 illustrates the process of planar motion calculation.

The homography can be related to camera motion and plane location as follows:

$$\mathbf{H} = \mathbf{R} + \frac{1}{d} \mathbf{t} \mathbf{n}^T, \quad (11)$$

where d is the distance from the camera to the plane (the height of the humanoid approximately). \mathbf{R} describes a rotation γ about the Z axis and can be expressed as

$$\mathbf{R} = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

And \mathbf{t} is a translation vector with the form

$$\mathbf{t} = (t_x, t_y, 0). \quad (13)$$

As the unit normal \mathbf{n} is $(0, 0, 1)$, considering the point \mathbf{p} , the rotation matrix \mathbf{R} , and the vector \mathbf{t} (where \mathbf{R} and \mathbf{t} are calculated from the homography \mathbf{H}), the new planar position \mathbf{p}_{new} can be computed as

$$\mathbf{p}_{\text{new}} = \mathbf{R}\mathbf{p} + \mathbf{t}. \quad (14)$$

More details about planar motion based on homography can be found in [59]. The HS-RANSAC algorithm and (11)–(14) were implemented in a Raspberry Pi. Since the computation must be verified in real time, the number of iterations is

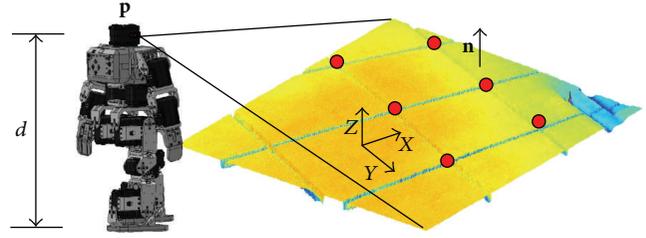


FIGURE 9: Process of planar motion calculation based on homographies.

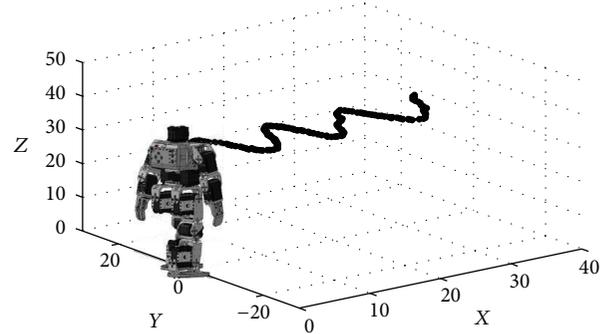


FIGURE 10: Position calculated from the homography.

fixed to only 150. Figure 10 shows the calculated positions from the homographies estimated during the humanoid locomotion. Such a figure demonstrates that the information of the estimated position adequately reflects the humanoid movement in spite of the reduced number of iterations.

8. Conclusions

In this paper, a new method for robustly estimating homographies from point correspondences based on the evolutionary algorithm has been presented. The approach combines the RANSAC method and the harmony search (HS) algorithm. With the combination, the proposed method adopts an alternative sampling strategy in comparison with RANSAC to build putative solutions. Under the new mechanism, new candidate solutions are generated iteratively by taking into consideration the quality of models produced by previous candidate solutions, instead of relying over a pure random selection as it is the case of RANSAC. On the other hand, a more accurate objective function was incorporated to adequately assess the quality of a candidate model. As a result, the proposed approach can substantially reduce the number of iterations still preserving the robust capabilities of RANSAC.

The proposed approach has been compared to other similar techniques proposed in the literature such as standard RANSAC [14], the MLESAC [17], the SIMFIT method [18], the projection-pursuit algorithm [19], the TSSE [20], and the PSO algorithm (PSO-RANSAC) [52]. The efficiency of the algorithm has been evaluated in terms of the detection rate (DR, Nof1), accuracy (E_s, E_r), and computational cost

(NFE). Experimental results that consider real and synthetic data provide evidence on the remarkable performance of the proposed algorithm in comparison to such methods. Additionally, in order to demonstrate the performance of the proposed approach in a real engineering application, it has been employed to solve the problem of position estimation in a humanoid robot.

Although the experimental results indicate that the proposed method can yield better results on estimating homographies, it should be noticed that the aim of our paper is not intended to beat all the RANSAC methods which have been proposed earlier but to show that the use of evolutionary approaches can effectively serve as an attractive alternative to solve complex optimization problems, yet demanding fewer function evaluations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

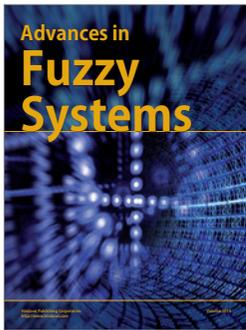
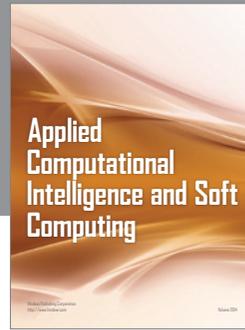
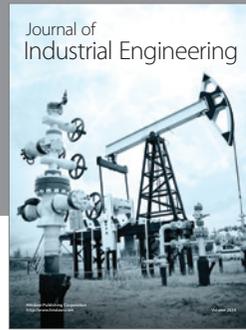
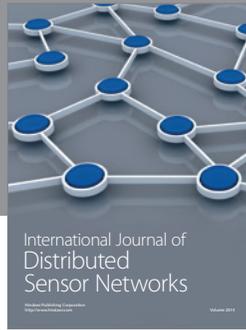
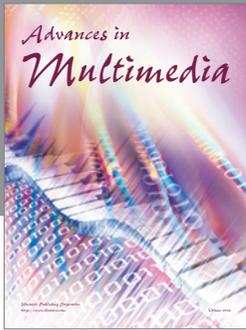
Acknowledgment

The present research has been supported under the Grant CONACYT CB 181053. The second author acknowledges the SEP of Mexico for its support (2014-2015).

References

- [1] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Proceedings of the Conference on Computer Graphics (SIGGRAPH '97)*, pp. 251–258, August 1997.
- [2] Y. He and R. Chung, "Image mosaicking for polyhedral scene and in particular singly visible surfaces," *Pattern Recognition*, vol. 41, no. 3, pp. 1200–1213, 2008.
- [3] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [4] A. Akyol and M. Gökmen, "Super-resolution reconstruction of faces by enhanced global models of shape and texture," *Pattern Recognition*, vol. 45, no. 12, pp. 4103–4116, 2012.
- [5] H. Huang, H. He, X. Fan, and J. Zhang, "Super-resolution of human face image using canonical correlation analysis," *Pattern Recognition*, vol. 43, no. 7, pp. 2532–2543, 2010.
- [6] K. Jia and S. Gong, "Hallucinating multiple occluded face images of different resolutions," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1768–1775, 2006.
- [7] O. Deniz, G. Bueno, E. Bermejo, and R. Sukthankar, "Fast and accurate global motion compensation," *Pattern Recognition*, vol. 44, no. 12, pp. 2887–2901, 2011.
- [8] E. Montijano and C. Sagues, "Distributed multi-camera visual mapping using topological maps of planar regions," *Pattern Recognition*, vol. 44, no. 7, pp. 1528–1539, 2011.
- [9] J. Su, R. Chung, and L. Jin, "Homography-based partitioning of curved surface for stereo correspondence establishment," *Pattern Recognition Letters*, vol. 28, no. 12, pp. 1459–1471, 2007.
- [10] T. T. Santos and C. H. Morimoto, "Multiple camera people detection and tracking using support integration," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 47–55, 2011.
- [11] H. Zhang and J. P. Ostrowski, "Visual motion planning for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 199–208, 2002.
- [12] C. Sagüés and J. J. Guerrero, "Visual correction for mobile robot homing," *Robotics and Autonomous Systems*, vol. 50, no. 1, pp. 41–49, 2005.
- [13] G. López-Nicolás, J. J. Guerrero, and C. Sagüés, "Visual control of vehicles using two-view geometry," *Mechatronics*, vol. 20, no. 2, pp. 315–325, 2010.
- [14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the Association for Computing Machinery*, vol. 24, no. 6, pp. 381–395, 1981.
- [15] E. İmre and A. Hilton, "Order statistics of RANSAC and their practical application," *International Journal of Computer Vision*, 2014.
- [16] X. Wu, Q. Zhao, and W. Bu, "A SIFT-based contactless palmprint verification approach using iterative RANSAC and local palmprint descriptors," *Pattern Recognition*, vol. 47, no. 10, pp. 3314–3326, 2014.
- [17] F. Zhou, Y. Cui, Y. Wang, L. Liu, and H. Gao, "Accurate and robust estimation of camera parameters using RANSAC," *Optics and Lasers in Engineering*, vol. 51, no. 3, pp. 197–212, 2013.
- [18] J. Matas and O. Chum, "Randomized RANSAC with $T_{d,d}$ test," *Image and Vision Computing*, vol. 22, no. 10, pp. 837–842, 2004.
- [19] C.-M. Cheng and S.-H. Lai, "A consensus sampling technique for fast and robust model fitting," *Pattern Recognition*, vol. 42, no. 7, pp. 1318–1329, 2009.
- [20] P. H. S. Torr and A. Zisserman, "MLESAC: a new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.
- [21] S. B. Heinrich, "Efficient and robust model fitting with unknown noise scale," *Image and Vision Computing*, vol. 31, no. 10, pp. 735–747, 2013.
- [22] R. Subbarao and P. Meer, "Beyond RANSAC: user independent robust regression," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '06)*, vol. no. 1, p. 101, June 2006.
- [23] H. Wang and D. Suter, "Robust adaptive-scale parametric model estimation for computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1459–1474, 2004.
- [24] L. Jun-Hua and L. Ming, "An analysis on convergence and convergence rate estimate of elitist genetic algorithms in noisy environments," *Optik*, vol. 124, no. 24, pp. 6780–6785, 2013.
- [25] E. Mendel, R. A. Krohling, and M. Campos, "Swarm algorithms with chaotic jumps applied to noisy optimization problems," *Information Sciences*, vol. 181, no. 20, pp. 4494–4514, 2011.
- [26] H. Pan, L. Wang, and B. Liu, "Particle swarm optimization for function optimization in noisy environment," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 908–919, 2006.
- [27] H.-G. Beyer, "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 239–267, 2000.
- [28] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [29] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems,"

- Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.
- [30] M. G. Omran and M. Mahdavi, “Global-best harmony search,” *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.
- [31] K. S. Lee and Z. W. Geem, “A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.
- [32] K. S. Lee, Z. W. Geem, S.-H. Lee, and K.-W. Bae, “The harmony search heuristic algorithm for discrete structural optimization,” *Engineering Optimization*, vol. 37, no. 7, pp. 663–684, 2005.
- [33] K. S. Lee and Z. W. Geem, “A new structural optimization method based on the harmony search algorithm,” *Computers and Structures*, vol. 82, no. 9–10, pp. 781–798, 2004.
- [34] J. H. Kim, Z. W. Geem, and E. S. Kim, “Parameter estimation of the nonlinear Muskingum model using harmony search,” *Journal of the American Water Resources Association*, vol. 37, no. 5, pp. 1131–1138, 2001.
- [35] Z. W. Geem, “Optimal cost design of water distribution networks using harmony search,” *Engineering Optimization*, vol. 38, no. 3, pp. 259–280, 2006.
- [36] Z. W. Geem, K. S. Lee, and Y. Park, “Application of harmony search to vehicle routing,” *American Journal of Applied Sciences*, vol. 2, no. 12, pp. 1552–1557, 2005.
- [37] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar, and M. Perez-Cisneros, “Multilevel thresholding segmentation based on harmony search optimization,” *Journal of Applied Mathematics*, vol. 2013, Article ID 575414, 24 pages, 2013.
- [38] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez-Cisneros, “Circle detection by harmony search optimization,” *Journal of Intelligent & Robotic Systems: Theory and Applications*, vol. 66, no. 3, pp. 359–376, 2012.
- [39] A. Kaveh and S. Talatahari, “Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures,” *Computers and Structures*, vol. 87, no. 5–6, pp. 267–283, 2009.
- [40] S. Mun and Z. W. Geem, “Determination of individual sound power levels of noise sources using a harmony search algorithm,” *International Journal of Industrial Ergonomics*, vol. 39, no. 2, pp. 366–370, 2009.
- [41] S. Mun and Z. W. Geem, “Determination of viscoelastic and damage properties of hot mix asphalt concrete using a harmony search algorithm,” *Mechanics of Materials*, vol. 41, no. 3, pp. 339–353, 2009.
- [42] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, “A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem,” *Expert Systems with Applications*, vol. 38, no. 4, pp. 3252–3259, 2011.
- [43] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, “A self-adaptive global best harmony search algorithm for continuous optimization problems,” *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830–848, 2010.
- [44] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “SURF: speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [45] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.
- [46] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2004.
- [47] P. Meer, “Robust techniques in computer vision,” in *Emerging Topics in Computer Vision*, G. Medioni and S. B. Kang, Eds., pp. 107–190, Prentice Hall, Boston, Mass, USA, 2004.
- [48] E. Valian, S. Tavakoli, and S. Mohanna, “An intelligent global harmony search approach to continuous optimization problems,” *Applied Mathematics and Computation*, vol. 232, pp. 670–684, 2014.
- [49] V. Kumar, J. K. Chhabra, and D. Kumar, “Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems,” *Journal of Computational Science*, vol. 5, no. 2, pp. 144–155, 2014.
- [50] S. M. Ashrafi and A. B. Dariane, “Performance evaluation of an improved harmony search algorithm for numerical optimization: Melody Search (MS),” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1301–1321, 2013.
- [51] C. Hamzaçebi, “Improving genetic algorithms’ performance by local search for continuous function optimization,” *Applied Mathematics and Computation*, vol. 196, no. 1, pp. 309–317, 2008.
- [52] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.
- [53] B. O. Arani, P. Mirzabeygi, and M. S. Panahi, “An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration-exploitation balance,” *Swarm and Evolutionary Computation*, vol. 11, pp. 1–15, 2013.
- [54] M. Clerc and J. Kennedy, “The particle swarm—explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [55] J. Chen, Q.-k. Pan, and J.-q. Li, “Harmony search algorithm with dynamic control parameters,” *Applied Mathematics and Computation*, vol. 219, no. 2, pp. 592–604, 2012.
- [56] A.-L. Lee and J.-H. Kim, “3-Dimensional pose sensor algorithm for humanoid robot,” *Control Engineering Practice*, vol. 18, no. 10, pp. 1173–1182, 2010.
- [57] H. Shahbazi, K. Jamshidi, A. H. Monadjemi, and H. Eslami, “Biologically inspired layered learning in humanoid robots,” *Knowledge-Based Systems*, vol. 57, pp. 8–27, 2014.
- [58] J. F. Seara and G. Schmidt, “Intelligent gaze control for vision-guided humanoid walking: methodological aspects,” *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 231–248, 2004.
- [59] B. Liang and N. Pears, “Visual navigation using planar homographies,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’02)*, vol. 1, pp. 205–210, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

