

# Scatter Search for the 3D Point Matching Problem in Image Registration

OSCAR CORDÓN

Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada. Daniel Saucedo Aranda s/n, 18071 Granada, Spain. [ocordon@decsai.ugr.es](mailto:ocordon@decsai.ugr.es)

SERGIO DAMAS, JOSE SANTAMARÍA

Departamento de Lenguajes y Sistemas Informáticos. Universidad de Granada. Daniel Saucedo Aranda s/n, 18071 Granada, Spain. [sdamas@ugr.es](mailto:sdamas@ugr.es), [jsantam@ugr.es](mailto:jsantam@ugr.es)

RAFAEL MARTÍ

Departamento de Estadística e Investigación Operativa. Universidad de Valencia. Dr. Moliner 50, 46100 Burjassot (Valencia), Spain. [rafael.marti@uv.es](mailto:rafael.marti@uv.es)

## Abstract

Scatter search is a population-based method that has recently been shown to yield promising outcomes for solving combinatorial and nonlinear optimization problems. Based on formulations originally proposed in the 1960s for combining decision rules and problem constraints, such as the surrogate constraint method, scatter search uses strategies for combining solution vectors that have proved effective in a variety of problem settings.

In this paper, we present a scatter search implementation designed to find high quality solutions for the 3D image registration problem, which has a significant number of applications in practice. This problem arises in computer vision applications when finding a correspondence or transformation between two computer images taken under different conditions. Our implementation goes beyond a simple exercise on applying scatter search, by incorporating innovative mechanisms to combine and improve solutions and to create a balance between intensification and diversification in the reference set. Besides, heuristic information taken from a preprocessing of the images is incorporated into the algorithm in order to improve its performance. Our computational experimentation in a real-world medical registration application establishes the effectiveness of the scatter search procedure in relation to different approaches usually applied to solve the problem.

## KeyWords

Metaheuristics, Evolutionary Computation, Scatter Search, Image Registration, Point Matching.

## 1. Introduction

The purpose of this paper is to develop a heuristic method for solving an important combinatorial optimization problem. Specifically we tackle the 3D Image Registration (IR) problem in the context of computer vision systems (Brown 1992). Practical applications of IR are numerous. They include 3D model construction (Eisert et al. 2000), autoradiograph alignment in Neuroscience (Rangarajan et al. 1997) or statistical physics (Yuille and Kosowsky 1994). The main contribution of our work is the development of a procedure based on the Scatter Search methodology (Glover 1977, 1998) that is used for searching the solution space of the optimization problem that appears in the IR process. Moreover, the proposed implementation incorporates innovative mechanisms to exploit the knowledge on the problem and to create a trade-off between intensification and diversification for an efficient search.

IR can be simply stated as finding a mapping between two images:  $I_1$  named *scene*, and  $I_2$  named *model*. The objective is to find the mathematical transformation  $f$  that applied to  $I_1$  obtains  $I_2$ . Generally speaking, an image is stored in a huge amount of pixels, therefore most IR methods usually apply a preprocessing to extract the most relevant geometric primitives (point, lines, etc) that, in a certain way, define the objects contained in the image. Therefore, in these *feature-based* methods, the problem is reduced to find the transformation between two sets of geometric primitives. In this paper, we restrict our attention to the case of two sets of primitives  $P_1$  and  $P_2$ , consisting uniquely of points ( $P_1 \subseteq I_1, P_2 \subseteq I_2$ ). Hence, this IR problem can be defined in two different search spaces (both with the same final goal of achieving the best alignment between the scene and model images): the space of parameters that define  $f$ , or the space of permutations of  $P_1$  (to match  $P_2$ ). While the former approach to solve the problem is based on directly searching for the best parameters defining the transformation  $f$  (see for example, Yamany et al. 1999), the latter receives the name of *point matching* and is probably the most classical method in feature-based registration.

Point matching can be described in mathematical terms as follows. Given two set of points  $P_1 = \{x_1, x_2, \dots, x_n\}$  and  $P_2 = \{y_1, y_2, \dots, y_m\}$ , the problem is to find a transformation  $f$  such that  $y_i = f(x_{\sigma(i)})$  for  $i=1, \dots, r$ , where  $r = \min(n, m)$  and  $\sigma$  is a permutation of size  $l$  (with  $l$  being the maximum between  $n$  and  $m$ ). The IR problem is then naturally divided into two phases. In the first one, a permutation  $\sigma$  of  $l$  elements defines the matching between the points in  $P_1$  and  $P_2$ . In the second phase, from this matching of points and using a numerical optimization method (usually least squares estimation), the parameters defining the transformation  $f_\sigma$  are computed. The objective is to find the transformation minimizing the distances between the model points and the corresponding transformed scene points. Therefore, in optimization terms, the value associated with permutation  $\sigma$  is given by the expression:

$$g(\sigma) = \frac{\sum_{i=1}^r \|f_\sigma(x_{\sigma(i)}) - y_i\|^2}{r}.$$

The point matching problem can be simply stated as minimizing  $g(\sigma)$  for any permutation  $\sigma$  of  $l$  elements and its corresponding transformation  $f_\sigma$ . In this paper we face the IR problem within this point matching approach, as it is the case of previous methods like the well known Iterative Closest Point algorithm (Besl 1992, Feldmar and Ayache 1996, Liu 2004), the technique usually applied in the computer vision field. We propose a scatter search implementation to find high quality solutions to this combinatorial optimization problem.

Our solution method presents contributions in both optimization and IR fields. Evolutionary methods have been widely applied to solve IR problems and typically use genetic operators for combination (Yamany et al. 1999, He and Narayana 2002, Chow et al. 2004). In this paper, we test the effectiveness of other combination mechanisms that do not rely on randomization in the context of point matching. Moreover, we design problem dependent search mechanisms based on image specific information, which have been proved to return good quality solutions (Cordón and Damas 2005). This information is *a priori* extracted from the shapes of the objects existing in the images and results in the association of characteristic values. This information is used in a two-fold way: on the one hand, the differences between the characteristic values of the matched points in the current solution are incorporated to the solution evaluation to better guide the search from a global perspective. On the other hand, they are taken into account in the neighborhood operator of the local search mechanism to properly intensify the search,

as well as in the diversification generation method to create an initial set of high quality solutions with a large degree of diversity among them. This way, we implement candidate list strategies in which permutations assigning feature points with similar characteristic values are ranked first since they seem more promising than those with relatively different values. The consideration of this additional information in the point matching process allows the scatter search algorithm to obtain high quality solutions faster than other previous approaches.

In the remaining of the paper, we first briefly describe the scatter search methodology and then, in Section 3, present our implementation to solve the point matching problem. The paper ends with the computational experiments and associated conclusions.

## 2. Scatter Search

Scatter search (SS) was first introduced in Glover (1977) as a heuristic for integer programming. SS orients its explorations systematically relative to a set of reference points that typically consist of good solutions obtained by prior problem solving efforts. The *scatter search template* (Glover 1998) has served as the main reference for most of the SS implementations to date. The SS methodology is very flexible, since each of its elements can be implemented in a variety of ways and degrees of sophistication. In this section we give a basic design to implement SS based on the well-known “five-method template” (Laguna and Marti, 2003). The advanced features of SS are related to the way these five methods are implemented. That is, the sophistication comes from the implementation of the SS methods instead of the decision to include or exclude some elements (as in the case of tabu search or other metaheuristics).

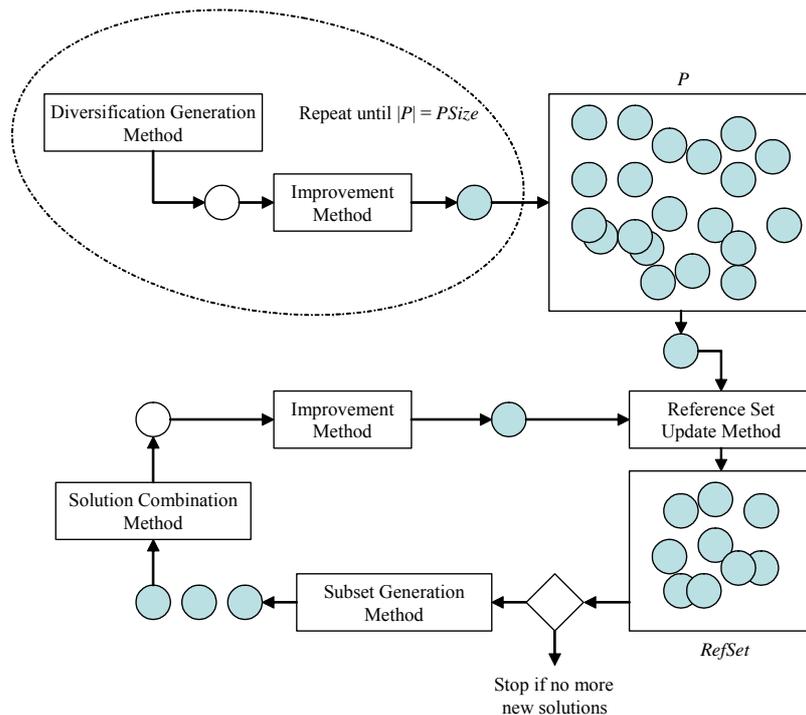


Figure 1: Schematic representation of a basic SS design

The fact that the mechanisms within SS are not restricted to a single uniform design allows the exploration of strategic possibilities that may prove effective in a particular implementation. These observations and principles lead to the following “five-method template” for implementing SS:

1. A *Diversification Generation Method* to generate a collection of diverse trial solutions, using an arbitrary trial solution (or seed solution) as an input.
2. An *Improvement Method* to transform a trial solution into one or more enhanced trial solutions. Neither the input nor the output solutions are required to be feasible, though the output solutions will more usually be expected to be so. If no improvement of the input trial solution results, the “enhanced” solution is considered to be the same as the input solution.

3. A *Reference Set Update Method* to build and maintain a *reference set* consisting of the  $b$  “best” solutions found (where the value of  $b$  is typically small, e.g., no more than 20), organized to provide efficient accessing by other parts of the method. Solutions gain membership to the reference set according to their quality or their diversity.
4. A *Subset Generation Method* to operate on the reference set, to produce several subsets of its solutions as a basis for creating combined solutions.
5. A *Solution Combination Method* to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solution vectors.

Figure 1 shows the interaction among these five methods and highlights the central role of the reference set. This basic design starts with the creation of an initial set of solutions  $P$ , and then extracts from it the reference set (*RefSet*) of solutions. The darker circles represent improved solutions resulting from the application of the Improvement Method.

The Diversification Generation Method is used to build a large set  $P$  of diverse solutions. The size of  $P$  ( $PSize$ ) is typically at least 10 times the size of *RefSet*. The initial reference set is built according to the Reference Set Update Method, which can take the  $b$  better solutions (as regards their quality in the problem solving) from  $P$  to compose the *RefSet*. However, diversity can be considered instead of or additionally to quality for the updating. For example, the Reference Set Update Method could consist of selecting  $b$  distinct and maximally diverse solutions from  $P$ . Regardless of the rules used to select the reference solutions, the solutions in *RefSet* are ordered according to quality, where the best solution is the first one in the list. The search is then initiated by applying the Subset Generation Method, which, in its simplest form, involves generating all pairs of reference solutions. The pairs of solutions in *RefSet* are selected one at a time and the Solution Combination Method is applied to generate one or more trial solutions. These trial solutions are subjected to the Improvement Method. The Reference Set Update Method is applied once again to build the new *RefSet* with the best solutions, according to the objective function value, from the current *RefSet* and the set of trial solutions. The basic procedure terminates after all the generated subsets are subjected to the Combination Method and none of the improved trial solutions are admitted to *RefSet* under the rules of the Reference Set Update Method.

The reference set, *RefSet*, is a collection of both high quality solutions and diverse solutions that are used to generate new solutions by way of applying the Combination Method. We can use a simple mechanism to construct an initial reference set and then update it during the search. The size of the reference set is denoted by  $b = b_1 + b_2 = |RefSet|$ . The construction of the initial reference set starts with the selection of the best  $b_1$  solutions from  $P$ . These solutions are added to *RefSet* and deleted from  $P$ . For each solution in  $P-RefSet$ , the minimum of the distances to the solutions in *RefSet* is computed. Then, the solution with the maximum of these minimum distances is selected. This solution is added to *RefSet* and deleted from  $P$ , and the minimum distances are updated. The process is repeated  $b_2$  times, where  $b_2 = b - b_1$ . The resulting reference set has  $b_1$  high quality solutions and  $b_2$  diverse solutions.

Of the five methods in the SS methodology, only four are strictly required. The Improvement Method is usually needed if high quality outcomes are desired, but a SS procedure can be implemented without it. On the other hand, hybrid SS designs could incorporate a short term tabu search or other complex metaheuristic as the improvement method (usually demanding more running time).

### 3. The Point Matching Search Method

As described in the previous section, the SS methodology basically consists of five elements (and their associated strategies). Three of them, the Diversification Generation, the Improvement and the Combination Methods, are problem dependent, and should be designed specifically for the problem at hand (although it is possible to design “generic” procedures, it is more effective to base the design on specific characteristics of the problem setting). The other two, the Reference Set Update and the Subset Generation Methods are context independent, and usually have a standard implementation.

In this work, we have implemented an advanced design of the reference set that complements the *RefSet* creation mechanism introduced in the previous Section by means of an updating process that proactively injects diversification into the search. This strategy is called *2-tier* design (Laguna and Martí, 2003) and is based on partitioning the *RefSet* in two tiers. The first tier *RefSet*<sub>1</sub> (*Quality RefSet*) consists of  $b_1$  high quality solutions  $\{S^1, \dots, S^{b_1}\}$ , while the second tier *RefSet*<sub>2</sub> (*Diversity RefSet*) consists of  $b_2 = b - b_1$  diverse

solutions  $\{S^{b_1+1}, \dots, S^{b_2}\}$ . The solutions in  $RefSet_1$  are ordered according to their objective function value and a new solution  $S$  replaces the worst solution  $S^{b_1}$  if the quality of the former is better than that of the latter.  $RefSet_2$  is ordered according to their diversity value, so, a new solution  $S$  replaces the worst solution  $S^{b_2}$  if  $d(S) > d(S^{b_2})$ , where the diversity value  $d$  is computed with the distance PMD defined in Section 3.2.

Following the guidelines given in Laguna and Martí (2003), we implement the Combination and the Subset Generation Methods with all the pairs (2-element subsets) in the  $RefSet$  with a static updating. As our reference set is composed of a quality and a diversity part, solution subsets of three different kinds are generated. On the one hand, subsets with the  $b_1*(b_1-1)$  possible pairs of solutions in the quality  $RefSet$  are created in order to intensify the search by combining high quality solutions. On the other hand, each of the  $b_2*(b_2-1)$  pairs of solutions in the diversity part are also considered to generate combined solutions for diversification purposes. Finally, a third group of  $b_1*b_2$  subsets is created by pairing each solution of the quality part with every one in the diversity part, thus getting combined solutions with an intermediate search behavior. Every new trial solution generated in the combination and improvement steps is inserted in a pool of solutions,  $Pool$ , and increasingly ordered according to their objective function. Those worst solutions in  $RefSet_1$  will be replaced with the corresponding better solutions in  $Pool$ . Subsequently, the remaining solutions in  $Pool$  (all of them with a lower quality than those in  $RefSet_1$ ) will be considered to update  $RefSet_2$ .

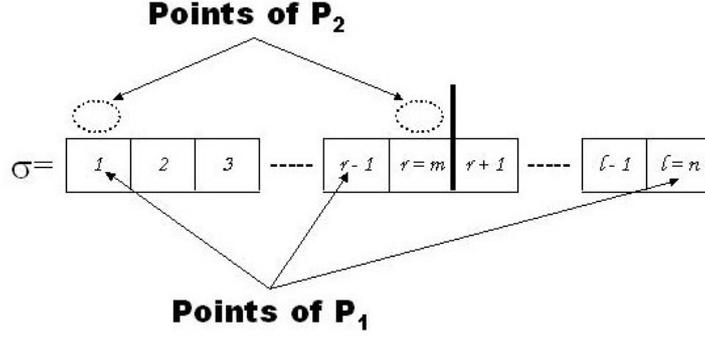
The next four subsections are respectively devoted to describe the coding scheme and the use of image curvature information in our search method, and the three specific SS elements mentioned above: the Diversification Generation Method, the Improvement Method and the Solution Combination Method.

### 3.1 Image Curvature Information and Coding Scheme

Our proposal is based on solving the IR problem by searching in the feature-based matching space. So, a coding scheme specifying the matching between model and scene image primitives (points, in our case) has to be defined. First, a pre-processing step (a 3D crest lines edge detector (Monga et al. 1991)) is applied to extract the most relevant feature points for each image,  $P_1 = \{x_1, x_2, \dots, x_n\}$  for the scene and  $P_2 = \{y_1, y_2, \dots, y_m\}$  for the model.

We first compute the iso-surface of the 3D image (i.e. the surface that separates regions of the space when considering a given intensity value known as iso-value). The goal is to obtain the boundary of the object under study (brain, liver, skull, etc.), from an image that typically stores different shapes. This surface defines, for any point  $x$  in the image, a set of curvatures  $C(x)$  reflecting the variation in each direction from  $x$  with respect to the tangent plane in this point. Hence, iso-surfaces allow us to reduce the huge amount of data we are dealing with. If we focus our attention on the zero-crossings of the curvature function  $C(x)$ , such points (known as *crest-lines* points) correspond to ridges and valleys of the iso-surface and represent its most important features. Thanks to this preprocessing, instead of facing the point matching problem from a million-size permutation, we take advantage of curvature information to extract the most relevant points in the image and face a hundred-size permutation problem.

The point matching between both images is represented as a permutation  $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_l)$  of size  $l = \max(n, m)$ , which associates the  $r$  points ( $r = \min(n, m)$ ) of the smaller size point set to the first  $r$  points of the permutation, selected from the larger one. Without loss of generality and to simplify the notation, we consider that  $P_1$  is larger or equal than  $P_2$  ( $n \geq m$ ). We have implemented the permutation in such a way that the first  $r$  elements ( $r = m$  in our case) of  $\sigma$  are the  $P_1$  points associated to each of the  $m$  points in  $P_2$ . Figure 2 illustrates these implementation details:

Figure 2: Implementation details of the point matching permutation  $\sigma$ 

We are able to infer the parameters of the implicit registration transformation  $f$  existing between the two 3D images,  $f_\sigma$ , from the point matching  $\sigma$  by means of simple numerical methods such as the Closed-form solution based on unit quaternion (Horn 1987) solving a least-squares problem. In this contribution, we consider  $f$  to be a similarity transformation, thus being composed of a rotation  $R=(\lambda, \langle \phi_x, \phi_y, \phi_z \rangle)$ , a translation  $t=(t_x, t_y, t_z)$ , and a uniform scaling  $s$ . Such a transformation can be used to register aerial and satellite images, bony structures in medical images, and brain multimodal images (Goshtasby 2005).

Once we know the expression of  $f_\sigma$ , we can estimate the registration error existing between the scene image points  $x_i$  and the model image points  $y_i$ , measured by the  $g()$  function as proposed by Arun et al. (1987):

$$g(\sigma) = \frac{\sum_{i=1}^r \|f_\sigma(x_{\sigma(i)}) - y_i\|^2}{r}, \quad \text{where } f_\sigma(x_{\sigma(i)}) = s \cdot R(x_{\sigma(i)}) + t$$

Note that  $g(\sigma)$  only computes the geometric information of both scene and model feature points. Some authors (Yamany et al. 1999, Luck et al. 2000, Robertson and Fisher 2002) have proposed several metaheuristic approaches for the IR problem which is only aimed at minimizing the previous  $g(\sigma)$  error function.

However, by only considering this objective function evaluation, search algorithms suffer from several problems such as their inability to handle with initial large misalignments between the two images and with those situations where the images have rotational or translational symmetries, both due to the fact of only dealing with the object geometry (Gagnon et al. 1994, Weik 1997). The latter aspects usually cause the given IR algorithm to more likely get trapped in local optima. A good explanation of such undesirable behaviour is found in Luck et al. (2000), where the authors use a simulated annealing method to address these problems.

This way, to overcome these problems in our SS-based IR procedure, we make use of problem dependent (context) information in our search method. To do so, we take again into account the curvature information  $C(x)$  extracted by the 3D crest lines edge detector. For each point  $x$ , we consider the two values of the first and second principal curvatures,  $k_1(x)$  and  $k_2(x)$  in  $C(x)$ , associated with the two principal orthogonal directions (which locally characterize the iso-surface). An interesting quality of this feature is that curvature values represent an invariant source of information respect to the similarity transformation  $f_\sigma$  we are dealing with, i.e., for each point  $x$ , it holds that  $k_1(x)=k_1(f_\sigma(x))$  and  $k_2(x)=k_2(f_\sigma(x))$ . The curvature attributes remain unchanged although a different  $f_\sigma$  is applied.

Therefore, given a scene point  $x_i$  and a model point  $y_j$  (each of them described by means of two curvature values), the closer every pair of curvature values, the higher the probability of a good matching between  $x_i$  and  $y_j$ . Therefore, we introduce the matrix  $D=(d_{ij})_{n \times m}$  to store all the Euclidean distances between the curvature values of each scene and model point. In mathematical terms:

$$d_{ij} = \sqrt{(k_1(x_i) - k_1(y_j))^2 + (k_2(x_i) - k_2(y_j))^2}, \quad \forall x_i \in P_1, y_j \in P_2$$

We will use these distances between curvature values in both the Diversification Generation Method and the Improvement Method of our SS procedure. In the former, they will be considered for an alternative solution evaluation, while in the latter they restrict the size of the neighbourhood of a given solution for an efficient search. In both cases, this curvature information prevents the problems mentioned above as will be shown in Section 4.

### 3.2 Diversification Generation and Reference Set Construction

As seen, the image heuristic information described in the previous section can be used to establish a preference for good assignments between the scene image points and the model image ones. Hence, a point  $x_i$  from the scene image is more likely to be assigned to those model points  $y_j$  presenting the same or similar curvature values  $k_1$  and  $k_2$ , i.e., having the lower distances  $d_{ij}$ .

We can make use of this information in order to generate the initial set  $P$  of diverse solutions for our SS procedure, thus obtaining solutions with both good quality and high diversity. Specifically, instead of fixing a selection order for the scene points  $x_i$  and then assigning the closest model point  $y_j$  (as regards the curvature values) not yet considered to each of them (which would result in a deterministic, greedy heuristic), we introduce randomness in both processes allowing each decision to be randomly taken among the best candidates. This way, our diversification generation method behaves similarly to a GRASP construction phase (Resende and Ribeiro 2001). The most important element in this kind of construction is that the selection in each step must be guided by a greedy function that adapts according to the pseudo-random selections made in the previous steps.

Our method starts by creating two candidate lists of unassigned points ( $CL_1$  and  $CL_2$ ), which at the beginning consist of all the points in the scene and the model (i.e., initially  $CL_1 = P_1$  and  $CL_2 = P_2$ ). For each element  $x_i$  in  $CL_1$ , we compute its potential distance  $d_i$  to  $CL_2$  as the minimum value of the distances from  $x_i$  to all the elements in  $CL_2$ . Then, we construct the restricted candidate list  $RCL_1$  with a percentage  $\alpha$  of the elements in  $CL_1$  with the lowest  $d_i$ -values, and we randomly select one element (say  $x_k$ ) from  $RCL_1$  for matching assignment. In order to find an appropriate point in the model to match with  $x_k$ , we construct the restricted candidate list  $RCL_2$  with a percentage  $\alpha$  of the elements in  $CL_2$  whose curvature values are closer to those of  $x_k$ , i.e., those elements presenting the lowest distance values to  $x_k$ . Finally, we randomly select a point (say  $y_k$ ) in  $RCL_2$  and match it with  $x_k$ . We update  $CL_1$  and  $CL_2$  ( $CL_1 = CL_1 - \{x_k\}$ ,  $CL_2 = CL_2 - \{y_k\}$ ) and perform a new iteration. The algorithm finishes when  $r = \min(n, m)$  points have been matched, i.e., when either  $CL_1$  or  $CL_2$  (that corresponding to the image with less points associated) becomes empty, and the remaining  $l-r$  points of the permutation are taken from the points remaining in the non-empty candidate list in a random order.

We repeat the application of this pseudo-random construction algorithm until we obtain  $|P|$  different solutions. We then apply the Improvement Method below to the generated solutions. Since two different solutions can produce the same improved solution, we apply, if necessary, the construction step a number of extra times until  $|P|$  different improved solutions are obtained. Let  $P$  be the set of these improved solutions.

As mentioned above, the reference set,  $RefSet$ , is a collection of  $b$  solutions (reference points) that are used to generate new solutions. The construction of the initial reference set starts with the selection of the best  $b_1 < b$  improved solutions from  $P$ . These solutions are added to  $RefSet$  and deleted from  $P$ . The remaining  $b_2 = b - b_1$   $RefSet$  solutions are selected from  $P$  taking into account the diversity. To do so, there is a need to define a distance metric between the solution vectors, i.e., between permutations. In this contribution, we consider the distance between two permutations  $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_r)$  and  $\rho = (\rho_1, \rho_2, \dots, \rho_r)$  records the number of times  $\sigma_i$  differs from  $\rho_i$  for  $i = 1, \dots, r$ . Besides, in order to favour the inclusion of quality solutions, as measured by the objective function, we bias the distance measure and divide this quantity by the sum of the evaluations of both solutions modified according to the curvature values. We call this metric *Point Matching Distance (PMD)* in order to differentiate it from the point curvature distance  $d$  and its definition in mathematical terms follows:

$$PMD(\sigma, \rho) = \frac{\sum_{i=1}^r \min(1, |\sigma_i - \rho_i|)}{F(\sigma) + F(\rho)}$$

In this expression, we are using an alternative solution evaluation  $F(\sigma)$  which incorporates the distance between curvature values to overcome the limitations of the objective function evaluation shown in the previous section. Specifically, the value of  $F(\sigma)$  is given by the expression:

$$F(\sigma) = w_g \cdot g(\sigma) + w_{n_{error}} \cdot m_{error}(\sigma) \quad \text{with} \quad m_{error}(\sigma) = \sum_{i=1}^r d_{i\sigma(i)}^2,$$

where the error function  $m_{error}(\sigma)$  measures the goodness of the matching  $\sigma$  by using the extra curvature information attributes associated to each feature point, and the weights  $w_g$  and  $w_{n_{error}}$  define the relative importance of each term. With such a function, we will have a more suitable similarity measure to make a better trek in the solution space, addressing the previous IR methods drawbacks. Furthermore, this definition of the  $m_{error}(\sigma)$  function is a specific case based on just two curvature values. Depending on the nature of the images considered, different attributes extracted in the IR pre-processing step can be considered for an easy redefinition of the  $m_{error}(\sigma)$  function as a reusability mechanism for other IR environments.

Finally, the minimum PMD from each improved solution in  $P$ - $RefSet$  to the current solutions in  $RefSet$  is computed. Then, the solution with the maximum of these minimum distances is selected. This solution is added to  $RefSet$  and deleted from  $P$ , and the minimum distances are updated. This process is repeated  $b_2$  times. As a result of the previous procedure, the obtained reference set has  $b_1$  high-quality solutions and  $b_2$  diverse solutions.

### 3.3 Improvement Method

Swaps are used as the primary mechanism to move from one solution to another in our Improvement Method. We define  $move(\sigma_i, \sigma_j)$ ,  $i \in \{1, \dots, k=\min(n,m)\}$ ,  $j \in \{1, \dots, l=\max(n,m)\}$ ,  $j \neq i$ , to consist of exchanging  $\sigma_i$  and  $\sigma_j$  in the current solution  $\sigma$ . This operation results in the ordering  $\sigma' = (\sigma_1, \dots, \sigma_{i-1}, \sigma_j, \sigma_{i+1}, \dots, \sigma_{j-1}, \sigma_i, \sigma_{j+1}, \dots, \sigma_l)$  when  $i < j$  (and symmetrically when  $j < i$ ).

An important difference with other combinatorial optimization problems is that here we cannot efficiently compute the move value associated with a trial move. In other words, to evaluate the quality of a move, we need to evaluate the final solution  $\sigma'$  when the move is applied, and compare its value with that of the initial solution  $\sigma$  ( $move\ value = g(\sigma) - g(\sigma')$ ). Note that a modification in the solution (permutation  $\sigma$ ) means a change in the matching and it implies a new estimated transformation  $f$ . Unfortunately, the simple modification performed by the swapping of the matching of two points can result in a completely different registration transformation  $f_{\sigma'}$ . Therefore, all the terms in the expression  $g(\sigma)$  can change and there is no way to calculate  $g(\sigma')$  without computing the new transformation  $f_{\sigma'}$  and the corresponding transformed scene points.

A solution  $\sigma$  represents the matching  $(x_{\sigma(i)}, y_i)$ , for  $i=1, \dots, k$ . Then, it is expected that, in a good matching, points  $x_{\sigma(i)}$  and  $y_i$  have similar curvature characteristics. In mathematical terms,  $d_{\sigma(i) i}$  should be relatively low for  $i=1, \dots, k$ . Since the move evaluation is a relatively time consuming operation, we reduce the neighborhood of a solution to include only promising moves. Specifically, the neighborhood of a solution  $\sigma$ ,  $N(\sigma)$ , is restricted to those moves  $move(\sigma_i, \sigma_j)$ , in which this difference of curvatures decreases for  $x_{\sigma(i)}$  or  $x_{\sigma(j)}$ :

$$N(\sigma) = \{move(\sigma_i, \sigma_j) \mid d_{\sigma(j) i} \leq d_{\sigma(i) i} \text{ or } d_{\sigma(i) j} \leq d_{\sigma(j) j}, 1 \leq i \leq k, 1 \leq j \leq l, j \neq i\}$$

Given a solution  $\sigma$  and its associated transformation  $f_{\sigma}$ , each element  $\sigma_i$  in the solution contributes to the solution evaluation  $g(\sigma)$  in  $\delta_i$ , where:

$$\delta_i = \left\| f_{\sigma}(x_{\sigma(i)}) - y_i \right\|^2$$

This measure shows that points should not be treated equally by a procedure that selects an index for a local search (i.e., for search intensification). We consider that  $\delta$  is a measure of influence and can be used to guide an efficient search of  $N(\sigma)$ . Specifically, we order the elements in a solution according to their  $\delta$ -value and select the element  $\sigma_{i^*}$  with the largest value for a swapping. Then, we scan  $N(\sigma)$  (in the order given by the curvature distance  $d_{\sigma(i^*j)}$ ) in search for the first element  $\sigma_j$  whose swapping  $move(\sigma_{i^*}, \sigma_j)$  results in an strictly positive move value (i.e., a move such that  $g(\sigma') \leq g(\sigma)$ ). We apply this first improving strategy, since previous studies (see for example Laguna et al. 1999) indicate that an effective search strategy results from searching for the first move in the neighborhood, as opposed to searching for the best move overall. If we do not find any improvement move associated to element  $\sigma_{i^*}$ , we resort to the next one in the ordered list and proceed in the same way. The local search method terminates either when  $N(\sigma)$  does not contain any improvement move or when a maximum iteration number is reached.

Computing the  $\delta$ -value, ordering the elements and selecting the most influential one is a computationally expensive calculation. In order to speed up the operation of our neighborhood operator, these  $\delta$ -values are however not updated after the execution of a move at each local search iteration, but on the contrary, we keep the order as it stands and select the next element in the list for the next iteration (and proceed in the same way for a certain number of subsequent iterations). The notion of not updating key values (e.g., move values) after every iteration is based on the *elite candidate list* suggested by Glover and Laguna (1997). The design considers that it is not absolutely necessary to update the value of the moves in a candidate list after an iteration is completed (i.e., the selected move is executed) because most of these move values either remain the same or their relative merit remains almost unchanged. The application of this strategy is particularly useful when the updating of the move values is computationally expensive, as in our context. After  $k$  local search iterations, we update the  $\delta$ -values and compute the new order. The parameter  $k$  reflects the trade-off point between information accuracy and computational effort in the implementation, and will be set after experimentation.

Finally, a selective application of the local optimizer is also considered in order to speed up the run time of the whole SS procedure. As mentioned in Section 2, a SS algorithm can be implemented without this component, although its use allows it to obtain high quality outcomes. Hence, in order to obtain an appropriate balance between the solution quality upgrade resulting from the Improvement Method use and the time consumed by it, we decided not to run the local search over each solution generated by the Combination Method but only on some of them. Previous works have demonstrated that a selective application of the local optimizer, with a random choice based on a given, low probability, has resulted in good performance in different memetic algorithms and, specifically, in some SS implementations (Hart 1994, Lozano et al. 2004, Herrera et al. 2005). In our case, this decision is deterministically taken, as the combined solution is optimized only when its evaluation  $F$  is better than that of at least one of the two original solutions used to generate it by the Solution Combination Method.

### 3.4 Solution Combination Method

We have considered two types of combination methods, both of which generate a single combined solution from a subset composed of a pair of original solutions. The first one, named Partially Mapped Crossover (PMX), is based on random elements and is widely used in the context of genetic algorithms. The second one, named Voting Method (VM), is based on deterministic elements and is widely used in the context of adaptive memory programming algorithms. We will compare both types of combinations in our computational experiments section.

#### Partially Mapped Crossover

This is an implementation of the classical recombination operator for order-based representations named partially mapped crossover (PMX) (Goldberg and Lingle 1985). It is designed to preserve the absolute position of some elements in the first solution. The method randomly chooses two crossover points in one reference solution and copies the partial permutation between them into the new trial solution. Both crossover points also define a mapping between the elements in both reference solutions. The remaining elements are copied in the positions they appear in the second reference solutions. If one position is already occupied by an element copied from the first parent, the element provided by the mapping is copied. This process is iterated until the conflict is solved. In order to limit the randomness of the

method and to assure the contribution of both reference solutions to the new trial solution, we randomly generate the first crossover point  $cp_1$  in  $\{1, 0.5 * l\}$  (assuming that  $0.5 * l < r$ ) and set the second crossover point  $cp_2$  to  $cp_2 = cp_1 + 0.25 * l$ .

As stated by Cotta and Troya (1998), this is a *respectful* operator since it transmits to the combined solution a relevant number of characteristics from the original ones. In genetic terms, we say that PMX transmits a block forma (an equivalence class induced by the relations identified as relevant). These authors compare eight genetic operators in the context of flowshop problems (based on a permutation representation) and conclude that the PMX is the best one for them.

### Voting Method

The method scans (from left to right) both reference permutations, and uses the rule that each reference permutation votes for its first element that is still not included in the combined permutation (referred to as the “incipient element”). The voting determines the next element to enter the first still unassigned position of the combined permutation. This is a min-max rule in the sense that if any element of the reference permutation is chosen other than the incipient element, then it would increase the deviation between the reference and the combined permutations. Similarly, if the incipient element were placed later in the combined permutation than its next available position, this deviation would also increase. So the rule attempts to minimize the maximum deviation of the combined solution from the reference solution under consideration, subject to the fact that the other reference solution is also competing to contribute. A bias factor that gives more weight to the vote of the reference permutation with higher quality is also implemented for tie breaking. This rule is used when more than one element receives the same votes. Then, the element with highest weighted vote is selected, where the weight of a vote is directly proportional to the objective function value of the corresponding reference solution. Additional details about this combination method can be found in Campos et al. (2001).

## 4. Computational Experiments

In this section we present a number of experiments in order to study the performance of our proposal. As we will explain below, these tests have been carried out under the same conditions since we wanted to extend our conclusions to other possible situations.

The results obtained by our SS algorithm for the 3D feature-based IR problem will be compared against IR techniques belonging to the two existing approaches mentioned in the Introduction, those searching in the point matching space, and those directly searching in the registration transformation parameter space. From the former group, we will consider the recent improvement of the classical ICP algorithm by Liu (2004) (I-ICP); and the hybrid proposal by J. Luck et al. (2000), combining the ICP algorithm with a simulated annealing technique in an iterative framework (ICP+SA), with the aim of overcoming the ICP problem of likely falling in local optima<sup>1</sup>. Besides, the results obtained by the simple greedy algorithm (Greedy) described in Section 3.2 are also reported as a lower quality threshold for our SS procedure. On the other hand, an evolutionary approach, the fast real-coded dynamic genetic algorithm (Dyn-GA) introduced by Chow et al. (2004), will be used from the latter group. Every algorithm maintains their original form and just the fitness function of the genetic algorithm has been adapted in order to deal with the uniform scaling factor, not considered in its original proposal (it only considered a rigid transformation  $f$ , i.e., only rotations and translations were involved in the IR problem).

### 4.1 Experimental Setup

This subsection describes the experiments developed to estimate several registration transformations in four different 3D images. For the sake of clarity, it has been divided into three different parts. First, the 3D images used to design the four different IR scenarios considered are presented. Then, the sixteen IR problem instances to be solved are introduced by describing the pair of images to be registered in each scenario and the four registration transformations applied on each of them. Finally, we deal with the parameter settings for the different IR algorithms considered.

---

<sup>1</sup> We should notice that, although the two variants of ICP considered in our experimentation solve the IR problem working in the point matching space (as our SS-based proposal does), they are based on assigning each transformed scene point with the closest model image point. This way, *different scene points can be matched with the same model point*, thus making the point matching not be a permutation.

### 3D Images Considered

Our results correspond to a number of registration problems with four different 3D images. These images have been obtained from the BrainWeb database at McGill University (Collins et al. 1998, Kwan et al. 1999). The purpose of this simulator is to provide researchers with ground truth data for image analysis techniques and algorithms. BrainWeb has been widely used by the IR research community (Rogelj et al. 2002, Held et al. 2004, Wachowiak et al. 2004).

To consider a more realistic scenario, every image we will study corresponds to a magnetic resonance (MRI). Moreover, we have added different levels of noise to three of the four images used. The reason is to model noisy conditions related to the images acquired by some devices. Likewise, we cannot avoid one of the most important goals of IR: supporting critical decisions concerning the evolution of a patient's lesion. To do so, two of our images will include a multiple sclerosis lesion. The influence of these two factors (the noise intensity and the presence or absence of lesion) will allow us to design a set of experiments with different complexity levels.

A preprocessing step has been carried out to all these 3D images in order to obtain problem dependent information to guide the IR process as well as to reduce the huge amount of data stored in the initial instances of the images (see Section 3.1). Therefore, from every original image, we extract the isosurface and select crest-lines points with relevant curvature information.

The first image ("I<sub>1</sub>", see Figure 3) corresponds to an MRI of a healthy person obtained with an ideal scanner, i.e., no lesion is present and it is a noise free scenario. After the isosurface extraction to identify the brain and the crest line points study to choose those features with relevant curvature information, 583 points have been selected.

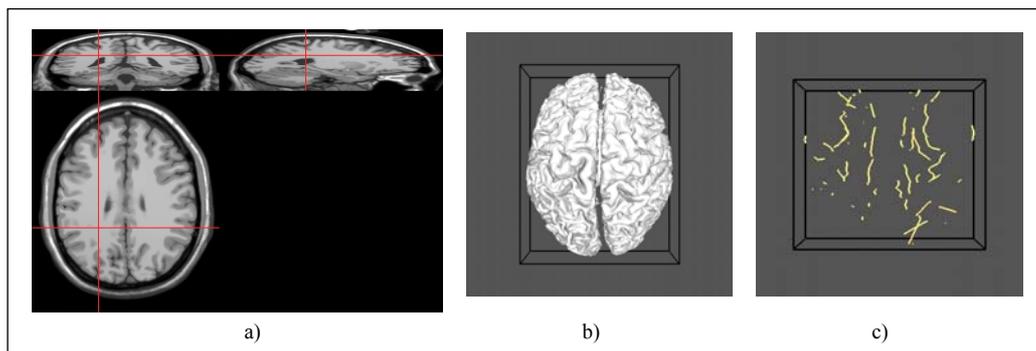


Figure 3: Image I<sub>1</sub>. a) Original MRI with three views (transverse, sagittal, and coronal). Different organs (skull, brain, eyes, etc.) can be clearly identified. b) Isosurface corresponding to the brain from the MRI. c) Crest line points with relevant curvature information

Image "I<sub>2</sub>" (Figure 4) corresponds to a low level of noise scenario (1% of Gaussian noise) of a healthy person. After the isosurface extraction, 393 crest line points have been chosen.

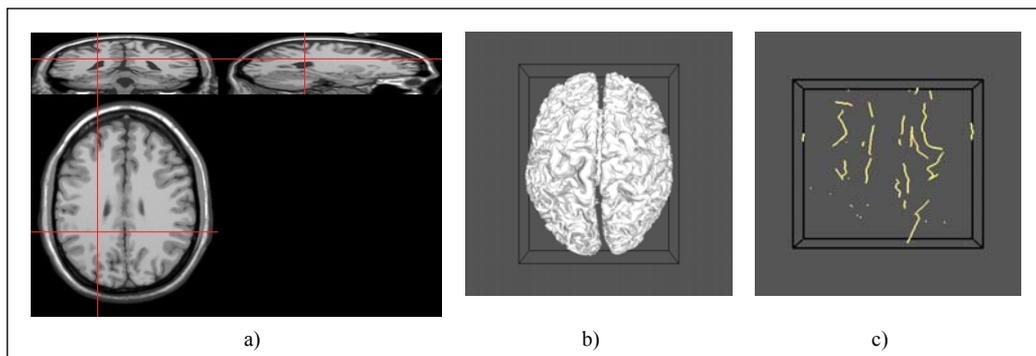


Figure 4: Image I<sub>2</sub>. a) MRI with a 1% of Gaussian noise. b) Isosurface corresponding to the brain from the MRI.

## c) Crest line points with relevant curvature information

The third image (“ $I_3$ ”, see Figure 5) includes a sclerosis multiple lesion and the same level of noise of  $I_2$ . After the isosurface extraction to identify the brain and the crest line points study to choose those features with relevant curvature information, 348 points have been selected.

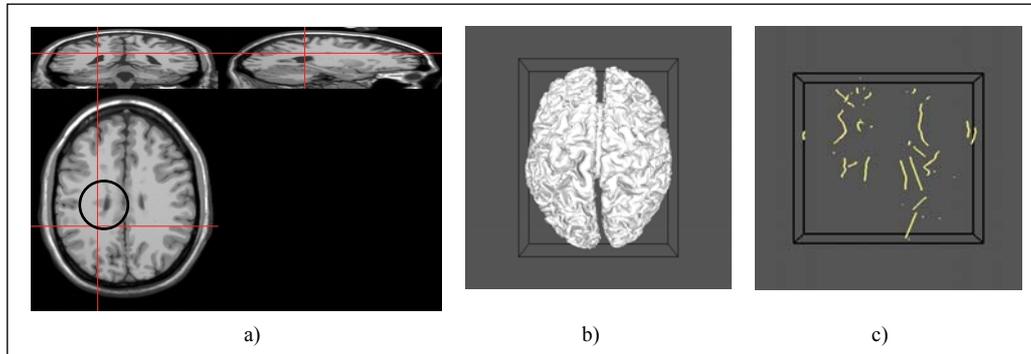


Figure 5: Image  $I_3$ . a) MRI with a 1% of Gaussian noise. Sclerosis multiple lesion is located using a circle. b) Isosurface corresponding to the brain from the MRI. c) Crest line points with relevant curvature information

Image “ $I_4$ ” (Figure 6) corresponds to a sclerosis multiple patient but the MRI has been acquired using a poor device (5% of Gaussian noise is introduced). Blurring can be easily observed in the extracted isosurface. Finally, 284 points with relevant curvature have been identified.

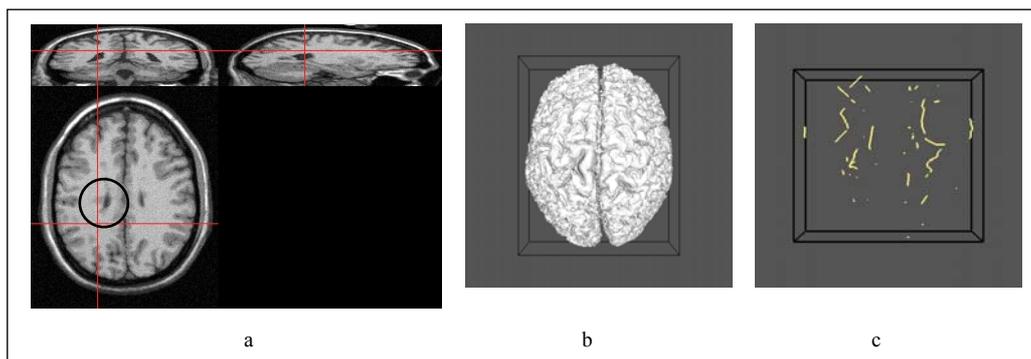


Figure 6: Image  $I_4$ . a) Original MRI with a 5% of Gaussian noise. Sclerosis multiple lesion is located using a circle. b) Isosurface corresponding to the brain from the MRI. Blurring is easily identified. c) Crest line points with relevant curvature information

### **IR problems considered**

Our results correspond to a number of IR problem instances for the different 3D images presented above which have suffered the same four global similarity transformations (noted as  $T_1$ ,  $T_2$ ,  $T_3$  and  $T_4$  in Table 1) to be estimated by the different 3D IR algorithms applied. These are ground truth transformations and they will allow us to quantify the accuracy of the IR solution returned by every algorithm. Hence, we will know in advance the optimal (i.e., the exact) registration transformation relating every scene and model input, thus being able to compute the objective function value associated to the problem optimal solution (see Section 4.2).

As mentioned in Section 3.1, similarity transformations involve rotation, translation and uniform scaling. They can be represented by eight parameters: one for the rotation magnitude ( $\lambda$ ), three for the rotation axis ( $axis_x$ ,  $axis_y$ ,  $axis_z$ ), three for the translation vector ( $t_x$ ,  $t_y$ ,  $t_z$ ), and one more for the uniform scaling factor ( $s$ ). In order to achieve a good solution, every algorithm must estimate these eight parameters accurately. Values in Table 1 have been selected within the appropriate ranges in such a way that important transformations have to be estimated. Both rotation and translation vectors represent a strong change in the object location. In fact, the lowest rotation angle is  $115^\circ$ . Meanwhile, translation values are

also high. Likewise the scaling factor ranges from 0.8 (in the second transformation) to 1.2 (in the fourth one). This way, complex IR problem instances are likely to be generated.

Table 1: Global similarity transformations considered

	$\lambda$	$axis_x$	$axis_y$	$axis_z$	$t_x$	$t_y$	$t_z$	$s$
$\mathbf{T}_1$	115.0	-0.863868	0.259161	0.431934	-26.0	15.50	-4.60	1.0
$\mathbf{T}_2$	168.0	0.676716	-0.290021	0.676716	6.0	5.50	-4.60	0.8
$\mathbf{T}_3$	235.0	-0.303046	-0.808122	0.505076	16.0	-5.50	-4.60	1.0
$\mathbf{T}_4$	276.9	-0.872872	0.436436	-0.218218	-12.0	5.50	-24.60	1.2

Moreover, in order to deal with a set of problem instances with different complexity levels (see Table 2), we will consider the following scenarios (from lower to higher difficulty):  $I_1$  vs.  $T_i(I_2)$ ,  $I_1$  vs.  $T_i(I_3)$ ,  $I_1$  vs.  $T_i(I_4)$ , and  $I_2$  vs.  $T_i(I_4)$ . Therefore, every algorithm will face sixteen different IR problem instances, resulting from the combination of the four scenarios and the four different transformations  $T_i$ .

Table 2: From top to bottom: increasing complexity ranking of the IR problem scenarios considered

IR problem	Scene image		Model image	
	Lesion	Noise	Lesion	Noise
$I_1$ vs. $T_i(I_2)$	No	No	No	1%
$I_1$ vs. $T_i(I_3)$	No	No	Yes	1%
$I_1$ vs. $T_i(I_4)$	No	No	Yes	5%
$I_2$ vs. $T_i(I_4)$	Yes	1%	Yes	5%

### Parameter Settings

Before performing the final experimentation, we have just made a preliminary study on the more suitable parameter values for the different IR algorithms to be considered in the subsequent experimentation. Both the preliminary test and the later experimentation have been made on a platform with an Intel Pentium IV 2.6 MHz processor and the SuSe GNU/linux 9.2 operative system. All the algorithms have been implemented with the C++ programming language and compiled with GNU/gcc.

For the I-ICP algorithm, it has been fixed a maximum of 40 iterations to ensure its success under some needed favourable initial conditions. For example, an initial registration transformation close to the ground truth one is needed by the algorithm to achieve good results. Unfortunately, such information from which an optimal starting point could be inferred is not usually available, thereby we have chosen an arbitrary rotation, a translation given from the subtraction of both scene and model centroids, and a uniform scaling factor estimated as done in (Horn 1987). Furthermore, the parameter associated to I-ICP has been established to the same value the author used in (Liu 2004).

For the ICP+SA algorithm, a maximum of 40 iterations for the wrapped I-ICP algorithm has also been fixed. The annealing process has 20 iterations and 50 trial movements around each annealing iteration, with an initial temperature value estimated with  $T_0 = [\mu/\ln(\phi)] \cdot C(S_0)$ , where  $C(S_0)$  is the cost of the given solution generated by the previous run of I-ICP, and both the  $\mu$  and the  $\phi$  factors take value 0.3. Each ICP+SA two-step iteration involves subsequently applying the I-ICP algorithm, optimizing the previous generated solution with the annealing process (if it achieves a best solution reducing the  $g()$  error function), and again starting a new iteration. We have only considered one iteration for the ICP+SA procedure.

In the Dynamic Genetic algorithm (Dyn-GA), it has been established the size of the initial population to 100 individuals and maintained the remaining of the specific parameters with their original values (Chow et al. 2004).

Our SS proposal deals with an initial set  $P$  comprised by 80 diverse solutions, and a RefSet composed of  $b = b_1 + b_2 = 10$  solutions, with  $b_1=7$  in the Quality subset and  $b_2=3$  in the Diversity subset. The local

search algorithm putting into effect the Improvement Method is run a maximum of 80 iterations at each execution, updating the  $\delta$  matching vector each  $k=10$  local search iterations (see Section 3.3). This way,  $\delta$  will be updated seven times each local search run thereby.

For both the Dyn-GA and the SS algorithms, we have established a maximum CPU time of 20 seconds at each run. Furthermore, for each one of the two latter algorithms as well as for the ICP+SA one, we have performed a total of 15 runs (with different random seeds) for each of the sixteen problem instances in order to avoid the usual random bias of probabilistic algorithms.

## 4.2 Experiments and Analysis of Results

This section is devoted to report the results obtained in the experimentation developed. For the sake of clarity, we have divided it into two different parts. In the first subsection, a preliminary experimental study is made to analyze the performance of the different variants of our SS proposal, where the two combination operators implemented and several weight<sup>2</sup> vectors in the objective function (see Section 3.2) are tested. The results of the simple greedy heuristic are considered as a lower quality threshold for this first experimentation. Then, the best SS variant is compared against the remaining state-of-the-art IR techniques, I-ICP, ICP+SA, and Dyn-GA, in order to measure the actual performance of our proposal in the problem solving.

### SS-PMX versus SS-Voting

Table 3 shows a detailed comparison between the two considered combination methods, based on the use of the PMX and Voting operators, to make a subsequent selection of the best one for being included in our final proposal of SS-based IR method. The structure of such table is organized in four subtables, every one referencing to each of the four IR scenarios presented in Table 2, where every subtable is divided into four parts devoted to the four transformations considered (see Table 1).

For each IR problem instance (specified by a given IR scenario and by one of the four transformations), the comparison between both combination methods is done by considering five different values of the coefficients in the  $F$  evaluation function. Each of these variants is denoted by  $WC_{x-y}$ , where  $x$  and  $y$  correspond to the  $w_g$  and  $w_{n_{error}}$  weighting coefficients in the objective function (see Section 3.2). The weight vectors considered range from a search process only guided by the use of the problem dependent (image-specific information) measuring the point matching quality by means of the curvature information and not taking into account the registration error  $g()$  ( $(w_g, w_{n_{error}}) = (0,1)$ ), to another only guided by the registration error ( $(w_g, w_{n_{error}}) = (1,0)$ ), as usual in the area. Three intermediate situations are also tested ( $(w_g, w_{n_{error}}) = \{(0.2,0.8), (0.5,0.5), (0.8,0.2)\}$ ), where a different trade-off is established between both optimization criteria.

The results shown in each table cell concern to the  $g()$  error function, the final evaluation measure of the overlapping obtained between both images, for the registration estimation obtained by every one of the two combination methods over the fifteen runs performed. The minimum, maximum, mean and standard deviation values, noted by  $m$ ,  $M$ ,  $\mu$ , and  $\sigma$ , respectively, are reported. Moreover, the error value obtained by the baseline greedy algorithm is also shown for each of the sixteen instances.

A special mention must be firstly done about the relationship between the different complexity levels of the four scenarios and the registration estimations performed by each IR algorithm in Table 3. We see how for those IR scenarios where one of the two involved images corresponds to a lesion situation (the last three subtables), the  $g()$  values are considerably increased.

---

<sup>2</sup> The weights in the objective function have been previously normalized as  $w_g = W_g$  and

$$w_{n_{error}} = w_{n_{error}} \cdot \left( \frac{m_{error(\sigma_0)}}{g(\sigma_0)} \right),$$

with  $m_{error(\sigma_0)}$  and  $g(\sigma_0)$  being respectively the matching and registration errors of the initial solution,  $\sigma_0$ , in order to get a uniform measure of both the matching error (curvature-derived error) and the registration error ( $g()$ ).

Analyzing the results shown in Table 3, we can see how the SS-based IR method considering the PMX combination operator achieves the best results over both the other SS-based IR method variant, which considers the use of the Voting combination operator, and the greedy algorithm. The SS-PMX algorithm obtains both the best minimum and mean results in each of the sixteen IR instances. On the other hand, the worst performance is obtained by the greedy heuristic, as expected.

On the other hand, for a deeper comparison between the two SS variants, we will have a look to the frequency of application of the Improvement Method. Having in mind that a given new trial solution will be considered for the local search algorithm in case it overcomes one of its two parents, we have noticed how for the SS based on the PMX operator, the 32% of trial solutions (mean value over the fifteen runs developed) were considered for the application of the Improvement Method, in front of the 16% achieved by the Voting one. This proves the better properties offered by the former method which builds solutions with better quality and increases the chance to improve those new trial solutions by using the Improvement Method.

As regards the influence of the weight vector values in the behavior of the two SS algorithms, it can be seen that the weight combination  $WC_{0,5-0,5}$  allows us to obtain the best mean value in thirteen of the sixteen cases overall, while the best minimum value is obtained in twelve of the sixteen cases by using  $WC_{0,8-0,2}$ . In view of the latter, it seems that giving the same importance to both objective function criteria allows us to get the most robust results, while slightly reducing the importance of the registration error to consider the image curvature information results in the best individual performance.

Table 4: Average values corresponding to each objective function weight combination in the two SS variants

	SS-PMX					SS-Voting				
	$WC_{0-1}$	$WC_{0,2-0,8}$	$WC_{0,5-0,5}$	$WC_{0,8-0,2}$	$WC_{1-0}$	$WC_{0-1}$	$WC_{0,2-0,8}$	$WC_{0,5-0,5}$	$WC_{0,8-0,2}$	$WC_{1-0}$
$\bar{m}$	725	465	334	<b>267</b>	665	2525	2438	2428	2393	<b>2344</b>
$\bar{\mu}$	980	586	<b>440</b>	701	1988	2849	2676	2597	2507	<b>2454</b>

In order to confirm the latter assumption, Table 4 collects the average results obtained by each weight combination with the two SS variants. In view of the data shown in the table, the best performance for SS-Voting is obtained with the  $WC_{1-0}$  combination (i.e., guiding the search by only considering the registration error and not using the image-specific heuristic information at all) both for the minimum and mean indices. Besides, it can be seen how the higher and the lower the values of  $w_g$  and  $w_{n_{error}}$ , respectively, the lower (the better) the  $g()$  value. However, the small differences existing between the different weight combinations together with the general bad accuracy of this variant clearly make us conclude that this analysis is not very significant.

On the other hand, specifically focusing the attention on the PMX method (which gets the best results in every case, thus allowing us to perform a more interesting analysis); it is easy to observe how the intermediate weight combinations result in the best performance in both indices. It is evident that the worst results are obtained when considering a single term in the objective function: the worst mean values are derived when the heuristic information is not considered, i.e., when the weight combination  $WC_{1-0}$  is used; while the worst minimum values are obtained when the registration error  $g()$  is not taken into account ( $WC_{0-1}$  combination). Besides, the second worst result in each index is obtained by the other combination. This behavior shows how a good trade-off between the both terms of the objective function achieves a more suitable convergence. This way, the experimental results of the SS-PMX variant reinforce our initial intention to make use of additional information (the image curvature information, in our case) as a second term in the objective function for a better guided search process, showing how an appropriate trade-off between the two error criteria is needed to get accurate results.

Table 3: Registration errors obtained by the SS algorithm when considering the PMX and Voting combination operators and the five weight vector values. The table is split into four subtables considering every IR problem scenario. The best mean and minimum  $g()$  values are shown using the underlined bold font

		$T_1$										$T_2$											
		$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$				$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$	
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
$m$		2849	256	2135	165	2135	123	2102	107	2102	<u>96</u>	2102	1823	156	1366	93	1366	<u>57</u>	1345	64	1345	76	1345
$M$		-	410	2783	220	2589	197	2355	165	2288	2164	2221	-	960	1781	423	1657	109	1495	101	1464	1402	1405
$\mu$		-	331	2389	186	2329	144	2206	<u>127</u>	2184	851	2154	-	249	1528	139	1489	92	1416	<u>81</u>	1397	736	1377
$\sigma$		-	41	191	16	140	18	68	16	49	928	26	-	192	122	77	89	12	44	10	31	613	14

**$I_1$  vs.  $T_i(I_2)$**

		$T_3$										$T_4$											
		$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$				$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$	
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
$m$		2849	259	2135	135	2135	113	2102	<u>100</u>	2102	105	2102	4102	363	3075	210	3075	146	3027	<u>128</u>	3027	162	3027
$M$		-	438	2783	252	2589	193	2355	552	2263	2195	2221	-	562	4007	320	3728	254	3365	929	3295	3116	3199
$\mu$		-	313	2389	189	2328	<u>143</u>	2214	155	2181	861	2154	-	471	3438	259	3352	<u>202</u>	3174	240	3145	1417	3102
$\sigma$		-	55	191	29	140	20	71	108	44	942	27	-	61	275	30	201	30	95	187	71	1386	38

		$T_1$										$T_2$											
		$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$				$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$	
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
$m$		4072	911	2562	627	2558	479	2549	<u>358</u>	2430	401	2377	2606	562	1639	368	1639	284	1631	252	1555	<u>236</u>	1521
$M$		-	1982	3296	1136	2992	820	2841	2629	2629	2577	2577	-	1215	2109	638	1915	408	1818	1682	1682	1649	1649
$\mu$		-	1283	2885	773	2782	<u>567</u>	2648	766	2543	2095	2497	-	792	1863	484	1767	<u>330</u>	1690	680	1627	1433	1597
$\sigma$		-	262	162	145	143	79	87	563	57	705	54	-	168	120	75	88	35	55	573	36	422	34

**$I_1$  vs.  $T_i(I_3)$**

		$T_3$										$T_4$											
		$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$				$WC_{0-1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$	
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
$m$		4072	906	2562	584	2562	488	2549	<u>407</u>	2430	433	2377	5864	1313	3689	842	3689	668	3689	645	3499	<u>623</u>	3423
$M$		-	1724	3296	1095	2992	941	2841	2629	2629	2577	2577	-	3281	4747	1519	4309	944	4747	3626	3786	3712	3712
$\mu$		-	1293	2911	762	2791	<u>601</u>	2650	1132	2545	2158	2495	-	1991	4197	1069	3984	<u>787</u>	4197	970	3661	3052	3597
$\sigma$		-	276	187	143	136	111	85	881	58	696	53	-	596	267	159	199	73	267	721	82	1116	79

		$T_1$										$T_2$											
		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$			
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
<i>m</i>		4471	885	2549	451	2378	327	2378	<u>282</u>	2378	692	2365	2861	573	1631	315	1522	214	1522	<u>186</u>	1522	477	1513
<i>M</i>		-	1595	3230	1186	2893	970	2698	2657	2679	2591	2591	-	1102	2067	671	1851	553	1727	1714	1714	1642	1642
$\mu$		-	1213	2934	714	2613	<u>477</u>	2546	767	2513	2348	2467	-	739	1886	462	1668	<u>326</u>	1634	739	1609	1512	1578
$\sigma$		-	176	169	224	145	175	108	739	90	446	61	-	177	106	136	96	119	72	633	58	278	37

**$I_1$  vs.  $T_i(I_4)$**

		$T_3$										$T_4$											
		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$			
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
<i>m</i>		4471	847	2549	407	2378	293	2378	<u>283</u>	2378	387	2365	6439	1134	3671	667	3425	450	3425	<u>399</u>	3425	439	3406
<i>M</i>		-	1695	3230	1103	2893	907	2698	2679	2679	2566	2566	-	2095	4660	1502	4166	1261	3923	3502	3857	3695	3695
$\mu$		-	1153	2945	688	2605	<u>438</u>	2546	1088	2513	2214	2464	-	1689	4232	937	3751	<u>687</u>	3679	866	3620	3179	3550
$\sigma$		-	221	173	245	148	151	108	970	90	642	57	-	303	258	323	213	246	167	748	131	934	84

		$T_1$										$T_2$											
		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$			
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
<i>m</i>		4374	934	2544	480	2480	463	2480	<u>328</u>	2475	2318	2318	2799	534	1671	308	1620	266	1620	<u>219</u>	1584	472	1484
<i>M</i>		-	1387	3217	872	3217	1002	2844	2670	2741	2598	2598	-	969	2058	729	2058	542	1820	1754	1754	1662	1662
$\mu$		-	1003	2927	<u>628</u>	2773	637	2677	738	2592	2504	2504	-	630	1878	461	1809	<u>338</u>	1725	760	1669	1453	1606
$\sigma$		-	164	156	141	180	166	93	755	85	62	62	-	101	97	128	119	77	45	671	50	381	41

**$I_2$  vs.  $T_i(I_4)$**

		$T_3$										$T_4$											
		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$		$WC_{0.1}$		$WC_{0.2-0.8}$		$WC_{0.5-0.5}$		$WC_{0.8-0.2}$		$WC_{1-0}$			
		<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>Greedy</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>	<i>PMX</i>	<i>Voting</i>
<i>m</i>		4374	755	2544	477	2480	387	2480	<u>331</u>	2475	391	2439	6299	1211	4081	681	3571	583	3571	<u>504</u>	3564	3339	3339
<i>M</i>		-	1423	3217	921	3217	825	2844	2673	2741	2598	2598	-	2280	4632	1816	4632	1134	4096	3806	3947	3741	3741
$\mu$		-	1001	2927	649	2773	<u>506</u>	2677	1024	2595	2379	2512	-	1530	4251	973	4000	<u>766</u>	3871	1082	3725	3610	3610
$\sigma$		-	160	157	135	180	127	93	984	85	532	42	-	282	183	310	252	156	130	1080	122	94	94

### Comparison between SS and previous methods

In this section we compare our SS proposal (including the PMX combination operator and the  $WC_{0.5-0.5}$  weight values) with the best state-of-the-art IR algorithms in the literature: I-ICP, ICP+SA, and Dyn-GA. We compare the quality of the solution obtained with these four methods when solving the sixteen instances under consideration. We report the MSE (*Mean Square Error*) value of each method on each instance. As in previous experiments, we report the minimum, maximum, average and standard deviation of these values on fifteen independent runs.

As it is well known in the IR community, the MSE value is more adequate to compare general IR methods than the  $g$ -value described in the introduction, which restricts its application to permutation-based approaches. The expression of the MSE, in which each transformed scene point is assigned to the closest model image point (regardless the latter had been previously assigned to other scene point), follows:

$$MSE = \frac{\sum_{i=1}^r \|f(x_i) - y_i\|^2}{r},$$

where  $y_i$  is the closest model point to the transformed scene point  $x_i$ .

Table 5 reports the MSE values obtained by the four IR algorithms, as well as the MSE values of the optimal solutions (those corresponding to the registration of the two considered images by means of the actual registration transformations collected in Table 1), which are shown in brackets after the sixteen instance names. Notice that, every value in this table is rounded. Our first conclusion in view of the results collected in this table is that the relationship between the different complexity levels in the IR instances and the results obtained by each algorithm again happens in the same manner as depicted in Table 3. Notice how the obtained MSE values are each time more far away from the optimal ones when increasing the complexity of the IR scenario. On the other hand, it can be seen how our SS-based IR method achieves the best mean performance in fifteen of the sixteen cases, as well as the best minimum MSE value in thirteen of the sixteen cases. Moreover, we should notice that the results obtained by our approach in those instances where it does not achieve the best mean and minimum values of performance can be improved by choosing a different configuration for the  $w_g$  and the  $w_{n_{error}}$  weights instead of fixing them to one value for all the instances. However, we preferred to keep them unchanged across different instances to provide a robust method (as we did by selecting the same SS variant for every case).

Table 5: MSE values obtained by the three state-of-the-art IR algorithms and our SS-based IR method. The table is split into four subtables considering every IR problem scenario. The best mean and minimum values are shown using the underlined bold font

	<b>T<sub>1</sub> [32]</b>				<b>T<sub>2</sub> [21]</b>			
	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>	344	247	101	<b><u>35</u></b>	131	131	44	<b><u>37</u></b>
<i>M</i>	-	344	264	40	-	131	284	50
$\mu$	-	307	195	<b><u>37</u></b>	-	131	108	<b><u>43</u></b>
$\sigma$	-	38	51	2	-	0	52	4
<b>I<sub>1</sub> vs. T<sub>1</sub>(I<sub>2</sub>)</b>								
	<b>T<sub>3</sub> [32]</b>				<b>T<sub>4</sub> [47]</b>			
	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>	894	457	87	<b><u>57</u></b>	632	283	139	<b><u>49</u></b>
<i>M</i>	-	711	678	67	-	611	600	59
$\mu$	-	559	211	<b><u>63</u></b>	-	465	302	<b><u>54</u></b>
$\sigma$	-	81	137	3	-	101	121	3

		<b>T<sub>1</sub> [43]</b>				<b>T<sub>2</sub> [30]</b>			
		<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>		518	305	132	<u>90</u>	330	237	56	<u>50</u>
<i>M</i>		-	432	741	132	-	297	534	66
$\mu$		-	343	299	<u>112</u>	-	261	154	<u>57</u>
$\sigma$		-	32	144	12	-	18	114	4
<b>I<sub>1</sub> vs. T<sub>i</sub>(I<sub>3</sub>)</b>									
		<b>T<sub>3</sub> [43]</b>				<b>T<sub>4</sub> [62]</b>			
		<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>		438	279	139	<u>43</u>	478	336	221	<u>112</u>
<i>M</i>		-	389	839	235	-	429	841	143
$\mu$		-	347	326	<u>64</u>	-	382	354	<u>123</u>
$\sigma$		-	33	174	46	-	24	147	8

		<b>T<sub>1</sub> [46]</b>				<b>T<sub>2</sub> [30]</b>			
		<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>		704	236	<u>124</u>	149	1493	314	<u>48</u>	51
<i>M</i>		-	466	1083	269	-	388	299	167
$\mu$		-	385	255	<u>184</u>	-	359	163	<u>89</u>
$\sigma$		-	61	228	33	-	22	58	41
<b>I<sub>1</sub> vs. T<sub>i</sub>(I<sub>4</sub>)</b>									
		<b>T<sub>3</sub> [46]</b>				<b>T<sub>4</sub> [67]</b>			
		<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>		951	312	158	<u>52</u>	416	342	207	<u>95</u>
<i>M</i>		-	433	468	227	-	413	1222	375
$\mu$		-	381	225	<u>82</u>	-	367	415	<u>154</u>
$\sigma$		-	43	87	45	-	17	258	86

		<b>T<sub>1</sub> [29]</b>				<b>T<sub>2</sub> [18]</b>			
		<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>		237	230	<u>108</u>	128	341	142	58	<u>52</u>
<i>M</i>		-	237	348	298	-	341	270	188
$\mu$		-	236	<u>178</u>	193	-	268	106	<u>75</u>
$\sigma$		-	2	60	62	-	71	51	41
<b>I<sub>2</sub> vs. T<sub>i</sub>(I<sub>4</sub>)</b>									
		<b>T<sub>3</sub> [29]</b>				<b>T<sub>4</sub> [45]</b>			
		<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>	<i>I-ICP</i>	<i>ICP+SA</i>	<i>Dyn-GA</i>	<i>SS</i>
<i>m</i>		609	399	110	<u>70</u>	1588	962	164	<u>105</u>
<i>M</i>		-	439	611	278	-	1533	751	362
$\mu$		-	407	192	<u>104</u>	-	1247	298	<u>150</u>
$\sigma$		-	10	116	67	-	209	145	78

Finally, in order to clearly show the actual performance of each IR technique, Figure 7 collects the graphical representations of the real overlapping achieved by each IR algorithm in four of the sixteen instances considered, one from each IR scenario. It can be seen how our SS proposal always obtains the best registration (see the right-most column images). Besides, it must be noted that the performance improvement regarding the remaining algorithms is much more remarkable as the IR scenario complexity increases, and that the Dyn-GA approach is the only IR technique (apart from our proposal, of course) being able to properly solve the IR problem for the complex scenarios considered in the experimentation developed.

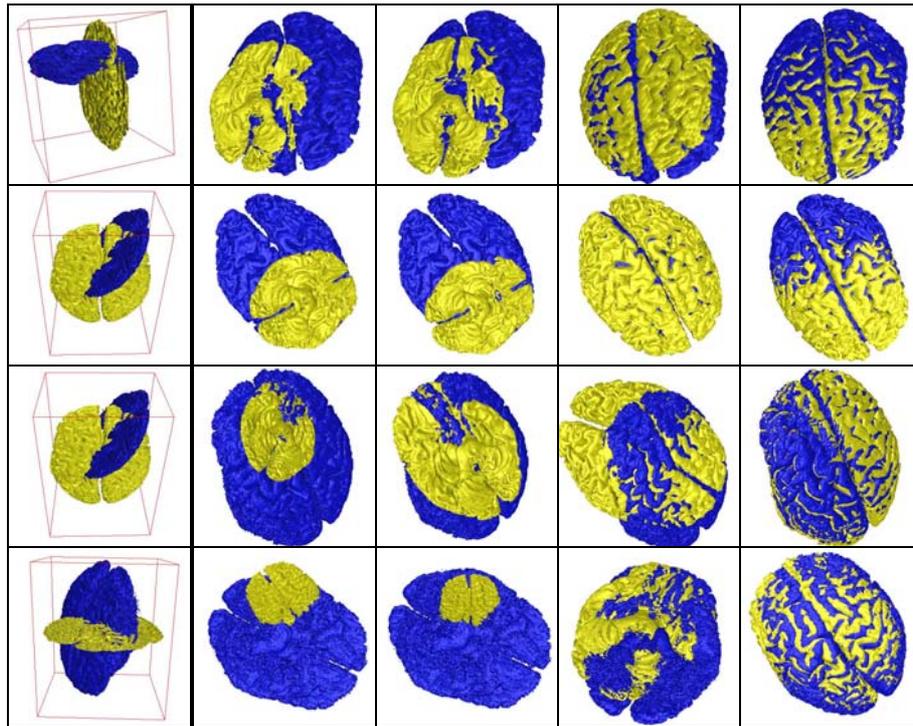


Figure 7: From top to bottom, the first column corresponds to the rendering of four of the sixteen IR instances:  $I_1$  vs  $T_1(I_2)$ ,  $I_1$  vs  $T_2(I_3)$ ,  $I_1$  vs  $T_3(I_4)$  and  $I_2$  vs  $T_4(I_4)$ ; while the next four columns correspond to the best registration estimations achieved by each IR algorithm (from left to right: I-ICP, ICP+SA, Dyn-GA and SS)

## 5. Concluding Remarks

We have described the development and implementation of a metaheuristic procedure for the optimization of IR. Our procedure extends the application of SS in an innovative way by implementing advanced reference set designs as well as by strategically including context information derived from the images characteristics. This information is incorporated in an improved solution evaluation, candidate list strategies within the local search method, and in the Diversification Generation Method.

One of the main goals of our effort has been to test the proposed procedure by employing real world data in realistic scenarios. In order to make a valid comparison against competing procedures, we have used the well establish MSE metric as well as graphical output. Our computational experiments show that SS has merit when compared to IR procedures previously identified to be the best.

## Acknowledgments

Research by Rafael Martí is partially supported by the *Ministerio de Educación y Ciencia* (refs. TIN2004-20061-E and TIC2003-C05-01) and by the *Agencia Valenciana de Ciència i Tecnologia* (ref. GRUPOS03/189).

Research by Oscar Cerdón and Sergio Damas is supported by the *Ministerio de Educación y Ciencia* (ref. TIC2003-00877), including FEDER fundings.

## References

- Arun, K.S., Huang, T., Blostein, S.D. 1987. Least-squares fitting of two 3-D point sets, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9 698-700.
- Besl, P.J., McKay, N.D. 1992. A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 239-256.
- Brown, L.G. 1992. A survey of image registration techniques, *ACM Computing Surveys* 24:4 325-376.
- Campos, V., Glover, F., Laguna, M., Martí, R. 2001. An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem, *Journal of Global Optimization* 21 397-414.
- Chow, C.K., Tsui, H.T., Lee, T. 2004. Surface registration using a dynamic genetic algorithm, *Pattern Recognition* 37 105-117.
- Collins, D.L., Zijdenbos, A.P., Kollkian, V., Sled, J.G., Kabani, N.J., Holmes, C.J., Evans, A.C. 1998. Design and construction of a realistic digital brain phantom, *IEEE Transactions on Medical Imaging* 17 463-468.
- Cordón, O., Damas, S., 2005, Image Registration with Iterated Local Search, *Journal of Heuristics* (to appear).
- Cotta, C., Troya, J.M. 1998. Genetic Forma Recombination in Permutation Flowshop Problems, *Evolutionary Computation* 6:1 25-44.
- Eisert, P., Steinbach, E., Girod, B. 2000. Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views, *IEEE Transactions on Circuits and Systems for Video Technology* 10:2 261-277.
- Feldmar, J., Ayache, N. 1996. Rigid, affine and locally affine registration of free-form surfaces, *International Journal of Computer Vision* 18:2 99-119.
- Gagnon, H., Soucy, M., Bergevin, R., Laurendeau, D. 1994. Registration of multiple range views for automatic 3-D model building, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* 581-586.
- Glover, F. 1977. Heuristics for Integer Programming Using Surrogate Constraints, *Decision Sciences* 8 156-166.
- Glover, F. 1998. A Template for Scatter Search and Path Relinking. In: Hao, J.K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (Eds.), *Artificial Evolution, Lecture Notes in Computer Science* 1363, Springer 13-54.
- Glover, F., Laguna, M. 1997. *Tabu Search*, Kluwer Academic Publisher.
- Goldberg, D.E., Lingle, R., Jr. 1985. Alleles, loci, and the traveling salesman problem. In Greffenstette, J.J. (Ed.), *Proceedings of the First International Conference on Genetic Algorithms* 154-159.
- Goshtasby, A.A. 2005. *2-D and 3-D Image Registration for Medical, Remote Sensing, and Industrial Applications*. Wiley Interscience.
- Hart, W.E. 1994. *Adaptive global optimization with local search*, PhD Thesis, University of California, San Diego, California.
- He, R., Narayana, P.A. 2002. Global optimization of mutual information: application to three-dimensional retrospective registration of magnetic resonance images, *Computerized Medical Imaging and Graphics* 26 277-292.
- Held, M., Weiser, W., Wilhelmstötter, F. 2004. Fully Automatic Elastic Registration of MR Images with Statistical Feature Extraction, *Journal of WSCG* 12:1 153-160.
- Herrera, F., Lozano, M., Molina, D. 2005. Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies, *European Journal of Operational Research* (to appear).
- Horn, B.K.P. 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4:4 629-642.

- Kwan, R.K.S., Evans, A.C., Pike, G.B. 1999. MRI simulation-based evaluation of image-processing and classification methods, *IEEE Transactions on Medical Imaging* 18 1085-1097.
- Laguna, M., Martí, R. 2003. *Scatter Search – Methodology and Implementations in C*, Kluwer Academic Publishers, Boston.
- Laguna, M., Martí, R., Campos, V. 1999. Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem, *Computers and Operations Research* 26 1217-1230.
- Liu, Y. 2004. Improving ICP with easy implementation for free-form surface matching, *Pattern Recognition* 37 211-226.
- Lozano, M., Herrera, F., Krasnogor, N., Molina, D. 2004. Real-coded memetic algorithms with crossover hill-climbing, *Evolutionary Computation* 12:3 273-302.
- Luck, J., Little, C., Ho, W. 2000. Registration of range data using a hybrid simulated annealing and iterative closest point algorithm, *Proceedings of IEEE International Conference on Robotics and Automation* 3739-3744.
- Monga, O., Deriche, R., Malandain, G., Cocquerez, J.P. 1991. Recursive filtering and edge tracking: two primary tools for 3D edge detection, *Image and Vision Computing* 9:4 203-214.
- Rangarajan, A., Chui, H., Mjolsness, E., Pappu, S., Davachi, L., Goldman-Rakic, P.S., Duncan, J.S. 1997. A Robust Point Matching Algorithm for Autoradiograph Alignment, *Medical Image Analysis* 1:4 379-398.
- Resende, M.G.C., Ribeiro, C.C. 2001. Greedy Randomized Adaptive Search Procedures, in Glover, F., Kochenberger, G. (Eds.), *State-of-the-art Handbook in Metaheuristics*, Kluwer Academic Publishers, Boston 219-250.
- Robertson, C., Fisher, R.B. 2002. Parallel Evolutionary Registration of Range Data, *Computer Vision and Image Understanding* 87 39-50.
- Rogelj, P., Kovačič, S. 2002. Validation of a Non-Rigid Registration Algorithm for Multi-Modal Data, Sonka, M., Fitzpatrick, J.M. (Eds.), *Proceedings of SPIE in Medical Imaging* 4684 299-307.
- Wachowiak, M.P., Smolíkova, R., Zheng, Y., Zurada, J.M., Elmaghraby, A.S. 2004. An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization, *IEEE Transactions on Evolutionary Computation* 8:3 289-301.
- Weik, S. 1997. Registration of 3-D partial surface models using luminance and depth information, *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling* 93-100.
- Yamany, S.M., Ahmed, M.N., Farag, A.A. 1999. A new genetic-based technique for matching 3D curves and surfaces, *Pattern Recognition* 32 1817-1820.
- Yuille, A.L., Kosowsky, J.J. 1994. Statistical physics algorithms that converge, *Neural Computation* 6:3 341-356.