*INFORMS Journal on Computing*
Vol. 20, No. 1, Winter 2008, pp. 55–68
ISSN 1091-9856 | EISSN 1526-5528 | 08 | 2001 | 0055

DOI 10.1287/ijoc.1060.0216


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

Oscar Cordón
Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 18071 Granada, Spain, and European Centre for Soft Computing, Edificio Científico-Tecnológico, 33600 Mieres (Asturias), Spain, ocordon@decsai.ugr.es

Sergio Damas
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, 18071 Granada, Spain, and European Centre for Soft Computing, Edificio Científico-Tecnológico, 33600 Mieres (Asturias), Spain, sdamas@ugr.es

Jose Santamaría
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Cádiz, Chile 1, 11002 Cádiz, Spain, jose.santamarialopez@uca.es

Rafael Martí
Departamento de Estadística e I.O., Universidad de Valencia, 46100 Burjassot, Valencia, Spain, rafael.marti@uv.es

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.

We present a scatter-search implementation designed to find high-quality solutions for the 3D image-registration problem, which has many practical applications. This problem arises in computer vision applications when finding a correspondence or transformation between two computer images obtained 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. Furthermore, heuristic information taken from a preprocessing of the images is incorporated into the algorithm to improve its performance. Our computational experimentation tackling two different medical registration applications establishes the effectiveness of scatter search in relation to different approaches usually applied to solving the problem. We have considered both simulated magnetic resonance images and real-world computerized tomography images as data sets. To measure the robustness of our proposal, the image data sets are intentionally selected for addressing registration environments with the presence of noise, anatomical lesions, and occlusions between images.

*Key words*: metaheuristics; evolutionary computation; scatter search; image registration; point matching
*History*: Accepted by Michel Gendreau, Area Editor for Heuristic Search and Learning; received June 2005; revised August 2005, May 2006, September 2006; accepted December 2006. Published online in *Articles in Advance* December 3, 2007.

## 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). The practical applications of IR are numerous and 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 scatter-search (SS) methodology (Glover 1977, 1998), which is used for searching the solution space of the optimization problem that appears in the IR process. The proposed SS-based IR algorithm is a significantly improved version of that in Cordón et al. (2004), which allows us to deal with more complex IR problems properly (see §4). In particular, innovative mechanisms to exploit the knowledge of the problem and to create a trade-off between intensification and diversification for an efficient search are incorporated (a restricted version of this method can be found in Cordón et al. 2005).

IR can be simply defined 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 number of pixels, so most IR methods usually apply preprocessing to extract the most relevant geometric primitives (points, 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 finding 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 solving the problem is based on directly searching for the best parameters defining the transformation $f$ (see, for example, Yamany et al. 1999), the latter, called *point matching*, 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 $P_1 = \{x_1, x_2, \ldots, x_n\}$ and $P_2 = \{y_1, y_2, \ldots, y_m\}$, the problem is to find a transformation $f$ such that $y_i = f(x_{\sigma(i)})$ for $i = 1, \ldots, r$, 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 phase, 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 a 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

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

The point-matching problem can be simply stated as minimizing $g(\sigma)$ for any permutation $\sigma$ of $l$ elements and its corresponding transformation $f_\sigma$. We face the IR problem within this point-matching approach, as do previous methods like the well-known iterative closest point algorithm (Besl and McKay 1992, Feldmar and Ayache 1996, Liu 2004), the technique usually applied in computer vision. We propose an SS implementation to find high-quality solutions for this combinatorial optimization problem.

Our solution method presents contributions in both optimization and IR. Evolutionary methods have been widely applied to solving IR problems and typically use genetic operators for combination (Matsopoulos et al. 1999, Yamany et al. 1999, Rouet et al. 2000, He and Narayana 2002, Chow et al. 2004, Wachowiak et al. 2004). On the other hand, SS (Glover 1977, Laguna and Martí 2003) is based on a systematic combination between solutions (instead of a randomized one like that usually employed in genetic algorithms (GAs)) taken from a considerably reduced evolved pool of solutions called the *reference set* (between five and 10 times lower than the usual GA population sizes). In this way, an efficient and accurate search process is encouraged for the IR problem in this paper thanks to the latter and to other innovative components that will be described later. Likewise, 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 2006). This information is a priori extracted from the shapes of the objects existing in the images and results in the association of heuristic values to each image point. This information is used in a twofold way: On the one hand, the differences between the heuristic values of the matched points in the current solution are incorporated into the solution evaluation better to 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 intensify the search properly, 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. In this way, we implement candidate-list strategies in which permutations assigning feature points with similar heuristic values are ranked first, because they seem more promising than do those with relatively different values. The consideration of this additional information in the point-matching process allows the SS algorithm to obtain high-quality solutions more quickly than do other previous approaches.

We first briefly describe SS methodology and then, in §3, we present our implementation to solve the point-matching problem. The paper ends with the computational experiments and associated conclusions.

## 2. Scatter Search

Scatter search 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 *SS template* (Glover 1998) has served as the main reference for most of the SS implementations

to date. SS methodology is very flexible because each of its components can be implemented in a variety of ways and degrees of sophistication. In the Online Supplement to this paper (available at http://joc.pubs.informs.org/ecompanion.html), we give a description to implement SS based on the well-known five-method template (Laguna and Martí 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 certain elements (as in the case of tabu search or other metaheuristics).

## 3. The Point-Matching Search Method

As described on the supplement, 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 the specific conditions 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. We consider the same design of the preliminary version (Cordón et al. 2004) for the two context-independent components. However, the implementation of two of the three specific elements (the diversification-generation method and the improvement method) has been changed to improve the performance of our SS-based IR technique, allowing it to deal with significantly more complex problem instances.

We have implemented an advanced design of the reference set that complements the *RefSet* creation mechanism introduced in the supplement by an updating process that proactively injects diversification into the search. This strategy is called a *2-tier* design (Laguna and Martí 2003) and is based on partitioning the *RefSet* into two tiers. The first tier, $RefSet_1$ (Quality *RefSet*), consists of $b_1$ high-quality solutions $\{S^1, \ldots, S^{b_1}\}$, while the second tier, $RefSet_2$ (*Diversity RefSet*), consists of $b_2 = b - b_1$ diverse solutions $\{S^{b_1+1}, \ldots, S^{b_2}\}$. The solutions in $RefSet_1$ are ordered according to their objective function 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 the diversity value of the solutions, 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 point-matching distance defined in §3.2.

Following the guidelines given in Laguna and Martí (2003), we implement the combination and the subset-generation methods with all 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 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 each 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 into a pool of solutions, *Pool*, and decreasingly ordered according to their objective-function value. Those worst solutions in $RefSet_1$ will be replaced by 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 describing 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, improvement method, and 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 preprocessing step (a 3D crest lines edge detector (Monga et al. 1991)) is applied to extract the most relevant feature points for each image, $P_1 = \{x_1, x_2, \ldots, x_n\}$ for the scene and $P_2 = \{y_1, y_2, \ldots, y_m\}$ 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 the 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 from $x$ in each direction with respect to the tangent plane at this point. Hence, iso-surfaces allow us to reduce the huge amount of data with which we are dealing. If we focus our attention on the zero-crossings of the curvature function $C(x)$, such points (known as *crest-line* 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

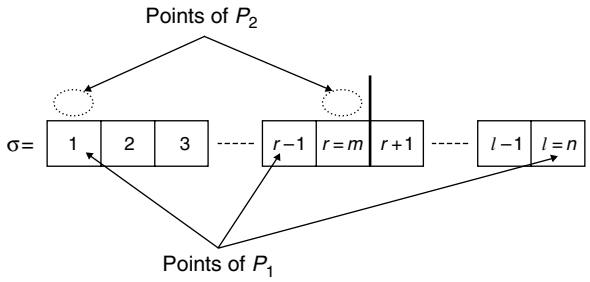


**Figure 1** **Implementation Details of the Point-Matching Permutation $\sigma$ with Size $n$**

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 $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_l)$ 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 than or equal to $P_2$ ($n \geq m$). We have implemented our solution method in such a way that the first $r$ elements of the permutation ($r = m$ in our case) are the $P_1$ points associated to each of the $m$ points in $P_2$. Figure 1 illustrates these implementation details in which we can see that the $P_1$ points located between positions $m+1$ and $n$ are not assigned to any point in $P_2$. Meanwhile, the first $m$ points of the permutation define a matching between the smaller point set $P_2$ (of size $m$) and the larger one $P_1$ (of size $n$); i.e., $\sigma(20) = 45$, defines a matching between the 20th point of $P_2$ and the 45th of $P_1$ (with $20 \leq m$ and $45 \leq n$).

Then, 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. 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 multimodal brain images (Goshtasby 2005).

Once we know the expression of $f_\sigma$, i.e., the $(R, t, s)$ parameters defining the similarity transformation, 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). We estimate the registration error by simply computing the Euclidean distance from each transformed point in $P_1$ (using the aforementioned $f_\sigma$ parameters) to its corresponding matching (considering $\sigma$), as shown in (2):

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

Note that $g(\sigma)$ computes only 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 that are aimed only at minimizing the previous $g(\sigma)$ error function. However, by considering only this objective evaluation function, search algorithms suffer from several problems such as their inability to handle large initial misalignments between the two images, and those situations where the images have rotational or translational symmetries, both due to the fact of dealing only with the object geometry (Gagnon et al. 1994, Weik 1997). The latter aspects usually make the given IR algorithm more likely to become trapped in local optima. A good explanation of such undesirable behavior is found in Luck et al. (2000), who use a simulated annealing method to address these problems.

To overcome the latter problems in our SS-based IR procedure, we make use of problem-dependent (context) information in the search method. To do so, we again take 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 with respect to the similarity transformation $f_\sigma$ with which we are dealing, i.e., for each point $x$, we have 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 described by 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,

$$d_{ij} = \sqrt{(k_1(x_i) - k_1(y_j))^2 + (k_2(x_i) - k_2(y_j))^2}, \quad \forall x_i \in P_1,\ y_j \in P_2. \quad (3)$$

We will use these distances between curvature values in both a diversification generation method and an 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 neighborhood of a given solution for an efficient search. In both cases, this curvature information avoids the aforementioned problems, as will be shown in §4.

### 3.2. Diversification Generation and Reference-Set Construction

Instead of considering a completely random generation of the initial solutions as in Cordón et al. (2004), we use the heuristic information related to image curvature described in §3.1 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 (heuristic) values $k_1$ and $k_2$, i.e., having the lower distances $d_{ij}$.

We can make use of this information to generate the initial set $P$ of diverse solutions for our SS procedure, thus obtaining solutions of 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 $y_j$ (with regard to the curvature values) not yet considered to each of them (which would result in a deterministic, greedy heuristic), we introduce randomness into both processes, allowing each decision to be randomly taken among the best candidates. In this way, our diversification generation 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 $CL_2$) that, 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 the matching assignment. 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 $CL_2$ (the one corresponding to the image with the fewest points associated) becomes empty, and the remaining $l - r$ points of the permutation are taken from the points still stored in the nonempty 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 solutions generated. Because two different solutions can produce the same improved solution, we apply the construction step a number of extra times, if necessary, 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. We consider the distance between two permutations $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_l)$ and $\rho = (\rho_1, \rho_2, \ldots, \rho_l)$ as the number of times $\sigma_i$ differs from $\rho_i$ for $i = 1, \ldots, r$. Additionally, to favor 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 the *point-matching distance* (PMD) to differentiate it from the point-curvature distance $d$ and its definition in mathematical terms is

$$\mathrm{PMD}(\sigma, \rho) = \frac{\sum_{i=1}^{r} \min(1, |\sigma_i - \rho_i|)}{F(\sigma) + F(\rho)}. \tag{4}$$

In this expression, we use an alternative solution evaluation $F(\sigma)$ that incorporates the distance between curvature values to overcome the limitations of the objective-function evaluation shown in the §3.1. Specifically, the value of $F(\sigma)$ is given by

$$F(\sigma) = w_g \cdot g(\sigma) + w_{n_{error}} \cdot m_{error}(\sigma) \quad \text{with } m_{error}(\sigma) = \sum_{i=1}^{r} d^2_{i\sigma(i)}, \tag{5}$$

where the error function $m_{error}(\sigma)$ measures the goodness of the matching $\sigma$ by using the extra curvature-information attributes associated with 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 search in the solution space, addressing the drawbacks of the previous IR methods. 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 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. The solution with the maximum of these minimum distances is then 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 reference set obtained 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 $move(\sigma_i, \sigma_j)$, $i \in \{1, \ldots, r = \min(n, m)\}$, $j \in \{1, \ldots, l = \max(n, m)\}$, $j \neq i$, as consisting of exchanging $\sigma_i$ and $\sigma_j$ in the current solution $\sigma$. This operation results in the ordering $\sigma' = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_j, \sigma_{i+1}, \ldots, \sigma_{j-1}, \sigma_i, \sigma_{j+1}, \ldots, \sigma_l)$ when $i < j$ (and symmetrically when $j < i$).

An important difference with other combinatorial optimization problems is that 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 of calculating $g(\sigma')$ without computing the new transformation $f_{\sigma'}$ and the corresponding transformed scene points.

In Cordón et al. (2004), two improvements were considered to speed up the local search procedure. On the one hand, a primary strategy was applied in the neighborhood generation by considering only promising swapping moves that take the curvature information as a base. On the other hand, a selective application of the local optimizer was also considered. Both are explained in detail in the remainder of this section.

A solution $\sigma$ represents the matching $(x_{\sigma(i)}, y_i)$, for $i = 1, \ldots, r$. It is then to be expected that, in a good matching, points $x_{\sigma(i)}$ and $y_i$ will have similar curvature values. In mathematical terms, $d_{\sigma(i)i}$ should be relatively low for $i = 1, \ldots, r$. Because 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 $x_{\sigma(j)}$:

$$N(\sigma) = \{move(\sigma_i, \sigma_j) / d_{\sigma(j)i} \leq d_{\sigma(i)i} \text{ or } d_{\sigma(i)j} \leq d_{\sigma(j)j}, \; 1 \leq i \leq r, \; 1 \leq j \leq l, \; j \neq i\}. \quad (6)$$

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

$$\delta_i = \| f_\sigma(x_{\sigma(i)}) - y_i \|^2. \quad (7)$$

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 swapping. We then scan $N(\sigma)$ (in the order given by the curvature distance $d_{\sigma(i^*)j}$) in search of the first element $\sigma_j$ whose swapping $move(\sigma_{i^*}, \sigma_j)$ results in a strictly positive move value (i.e., a move such that $g(\sigma') \leq g(\sigma)$). As documented in Laguna et al. (1999), the first strategy does not necessarily select the best available solution in the neighborhood but after several iterations it can lead the search to a better solution than a greedy strategy based on the selection of the best solution at each iteration. 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 number of iterations is reached.

Computing the $\delta$ value, ordering the elements and selecting the most influential one is a computationally expensive calculation. To speed up our neighborhood operator, these $\delta$ values are 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 on 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 each 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 has been completed (i.e., the selected move has been executed) because most of these move values either remain the same or their relative merit remains almost unchanged. Application of this strategy is particularly useful when the updating of the move values is computationally expensive, as in our case. 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 described above is also considered to speed up the whole SS procedure. Previous studies 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. 2006). 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, called partially mapped crossover (PMX), is based on random elements and is widely used in the context of GAs. The second one, called the voting method, 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.

**3.4.1. Partially Mapped Crossover.** This is an implementation of the classical recombination operator for order-based representations called 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 in which they appear in the second reference solution. 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. To limit the randomness of the method and to insure the contribution of both reference solutions to the new trial solution, we randomly generate the first crossover point $cp_1$ in $\{1, 0.5l\}$ (assuming that $0.5l < r$) and set the second crossover point $cp_2$ to $cp_2 = cp_1 + 0.25l$.

As stated by Cotta and Troya (1998), this is a *respectful* operator because it transmits a relevant number of features from the original solutions to the combined solution. 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.

**3.4.2. 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 has still not been 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 were 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 number of votes. The element with highest weighted vote is then selected, where the weight of a vote is directly proportional to the objective-function value of the corresponding reference solution. Additional details concerning this combination method can be found in Campos et al. (2001).

## 4. Computational Experiments

In this section we present a number of experiments developed to estimate several registration transformations for two different 3D medical image data sets to study the performance of our proposal. These tests have been carried out under the same conditions because we wanted to extend our conclusions to other possible situations. The most important challenge associated with the current experimentation is that the goal of the IR process is to register *two different images of similar objects*, instead of two images corresponding to the same object, thus reflecting a more realistic situation in medical IR.

On the one hand, we make use of a first data set from the BrainWeb public repository. It contains four simulated real-world magnetic resonance images (MRIs) of four human brains with noise, anatomical lesions, and a certain degree of occlusion. The main reason for choosing such an image simulator is to ease the comparison of our results with those from other previous or upcoming contributions. On the other hand, this first set of four images will be used as a benchmark to tune the parameters related to the best SS variant.

The second one corresponds to a couple of images recently used in Marai et al. (2006), kindly provided by the Rhode Island Hospital. They are computerized tomography (CT) images and correspond to two real human wrists. In this case we want to highlight the complexity of the problem to be tackled as described in §4.1.

This experimental setup is significantly more complex than those considered in Cordón et al. (2004, 2005). Of course, the registration of different objects is much more complicated than that of different views of the same object, and this motivated the extension of our previous algorithm to obtain good performance in the new IR scenario.

The results obtained by the SS algorithm proposed in this contribution for the 3D feature-based IR problem (from now on noted by SS*) will be compared with the old version of the SS-based IR process (noted simply by SS) (Cordón et al. 2004), and with several IR techniques belonging to the two existing approaches mentioned in §1, 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 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 a likely fall in local optima. We should note that, although these two variants of ICP work in the point-matching space (as does our SS-based proposal), they are based on assigning each transformed scene point to the closest model image point. This way, different scene points can be matched with the same model point, thus resulting in the point matching not being a permutation.

On the other hand, an evolutionary approach, the fast real-coded dynamic GA (Dyn-GA) introduced by Chow et al. (2004), will be used from the latter group. Every algorithm maintains its original form and only the fitness function of the GA has been adapted to deal with the uniform scaling factor, not considered in the original proposal (which used only a rigid transformation $f$, i.e., only rotations and translations were involved in the IR problem).

**Table 1 Global Similarity Transformations Considered**

| | $\lambda$ | $axis_x$ | $axis_y$ | $axis_z$ | $t_x$ | $t_y$ | $t_z$ | $s$ |
|---|---|---|---|---|---|---|---|---|
| $T_1$ | 115.0 | −0.863868 | 0.259161 | 0.431934 | −26.0 | 15.50 | −4.60 | 1.0 |
| $T_2$ | 168.0 | 0.676716 | −0.290021 | 0.676716 | 6.0 | 5.50 | −4.60 | 0.8 |
| $T_3$ | 235.0 | −0.303046 | −0.808122 | 0.505076 | 16.0 | −5.50 | −4.60 | 1.0 |
| $T_4$ | 276.9 | −0.872872 | 0.436436 | −0.218218 | −12.0 | 5.50 | −24.60 | 1.2 |

Section 4.1 provides a description of the experimental setup, detailing the 3D images, the IR problems tackled, and the parameter settings. We then present the experiments and the analysis of results in §4.2.

### 4.1. Experimental Setup

This section describes the experimental setup considered to estimate several registration transformations in the two different 3D medical image data sets mentioned above. See the Online Supplement for a description of the 3D data sets used to design the different IR scenarios in our experimentation as well as the parameter settings for the different IR algorithms considered. We introduce here the IR problems considered by describing the pair of images to be registered in each scenario and the four registration transformations applied to each of them.

Our results correspond to a number of IR problem instances for the different 3D images considered that 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 in the two data sets, thus enabling us to compute the objective-function value associated with the optimal solution of the problem in the BrainWeb images (see §4.2).

As mentioned in §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 ($s$). To achieve a good solution, every algorithm must estimate these eight parameters accurately. Values in Table 1 have been selected within the appropriate ranges so 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°. 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). In this way, complex IR problem instances are likely to be generated.

**Table 2 From Top to Bottom: Increasing Complexity Ranking of the IR Problem Scenarios**

| | Scene image | | Model image | |
|---|---|---|---|---|
| IR problem | 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. $T_i(I_4)$ | No | 1 | Yes | 5 |

Moreover, to deal with a set of problem instances with different complexity levels in the human brain MRI data set (see Table 2), we will consider the following scenarios (from lower to higher difficulty): $I_1$ versus $T_i(I_2)$, $I_1$ versus $T_i(I_3)$, $I_1$ versus $T_i(I_4)$, and $I_2$ versus $T_i(I_4)$. Therefore, every algorithm will face 16 different IR problem instances in this case, resulting from the combination of the four scenarios and the four different transformations $T_i$.

On the other hand, just four IR problems are defined in the second CT data set because only two images are available. We will consider the following four IR problem instances: $I_6$ versus $T_1(I_5)$, $I_6$ versus $T_2(I_5)$, $I_6$ versus $T_3(I_5)$, and $I_6$ versus $T_4(I_5)$. Therefore, in this case, every algorithm will face four different IR problem instances, resulting from the combination of the $I_6$ versus $I_5$ scenario and the four different transformations $T_i$, shown in Table 1.

### 4.2. Experiments and Analysis of Results

This section reports the results obtained in the experiments. In §4.2.1, 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 vectors in the objective function (see §3.2) are tested. Note that the weights in the objective function have been previously normalized as $w_g = w_g$,

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

with $m_{error(\sigma_0)}$ and $g(\sigma_0)$ being, respectively, the matching and registration errors of the initial solution, $\sigma_0$, to get a uniform measure of both the matching error (curvature-derived error) and the registration error ($g()$). Then, the best SS* variant is compared with the previous SS proposal in Cordón et al. 2004 (noted SS) and with the remaining state-of-the-art IR techniques, I-ICP (Liu 2004), ICP + SA (Luck et al. 2000), and Dyn-GA (Chow et al. 2004), to measure the actual performance of our proposal in the problem solving, when considering both the BrainWeb images (§4.2.2) and the real-world CT ones (§4.2.3).

**4.2.1. SS-PMX vs. SS-Voting in the BrainWeb Data Set.** We have undertaken a detailed comparison of the two combination methods considered, based on the use of the PMX and voting operators, to make a subsequent selection of the best one for inclusion in our final proposal of a SS*-based IR method.

For each of the 16 IR problem instances (specified by a given IR scenario and by one of the four transformations, see Tables 1 and 2), the comparison between both combination methods is made 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 §3.2). The weight vectors range from $(w_g, w_{n_{error}}) = (0, 1)$, where the search process is guided only by the problem-dependent (image-specific) information (measuring the point-matching quality by the curvature information and not taking into account the registration error $g()$), to $(w_g, w_{n_{error}}) = (1, 0)$, where the search is guided only by the registration error, as is the usual practice. 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.

From this analysis, we can conclude that using the PMX operator in the SS*-based IR method achieves significantly better results than does the voting method. The SS*-PMX algorithm obtains both the best minimum and mean results in each of the 16 IR instances. With regard to the influence of the weight vector values in the behavior of the two SS* algorithm variants, it can be concluded that the weight combination $WC_{(0.,5,0,5)}$ allows us to obtain the best mean value in 13 of the 16 cases overall.

We will include only a summarized, representative set of results from the broad experimentation developed. For the complete study, see Santamaria (2006).

Figure 2 shows the average results for the minimum and mean values obtained by each weight combination along the 16 IR problem instances for the PMX combination method. It is easy to observe how the intermediate weight combinations result in the best performance for both indices. Therefore, the worst results are obtained when considering a single term of the objective function, reinforcing our initial intention of making use of additional information (the curvature information in our case) as an additional term to guide the search process in a better way.

**4.2.2. Comparison Between SS* and Previous Methods in the BrainWeb Data Set.** In this section we compare our SS* proposal (considering the PMX combination operator and the $WC_{0.5-0.5}$ weight values) with SS (the previous version of SS*) and with

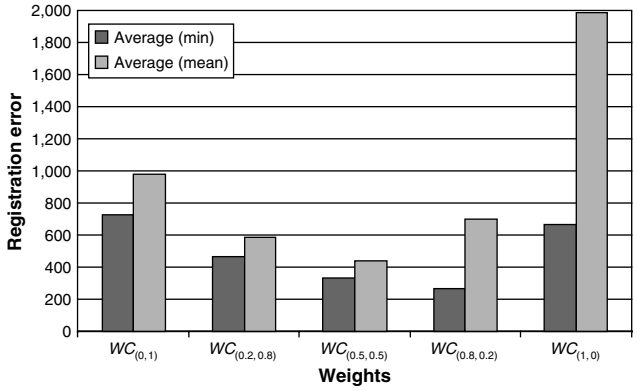


**Figure 2 Average Registration Error Values of the PMX Combination Method for Different Weighing Coefficients WC**

*Note.* A good trade-off between both terms of the objective function results in a more suitable result.

the best state-of-the-art IR algorithms in the literature: I-ICP (Liu 2004), ICP + SA (Luck et al. 2000), and Dyn-GA (Chow et al. 2004). We compare the quality of the solution obtained with these five methods when solving the 16 instances under consideration. We report the MSE (mean squared error) value of each method on each instance. The MSE value is more adequate for comparing general IR methods than the $g()$ value described in §1, which restricts its application to permutation-based approaches. It is also a well-known metric in the feature-based IR community (Besl and McKay 1992, Feldmar and Ayache 1996, Yamany et al. 1999, etc.). Moreover, the MSE has recently been proposed as a standard performance measure to prevent unfair comparisons between IR algorithms and motivate statistically accurate analysis (Robinson and Milanfar 2004).

The MSE, in which each transformed scene point is assigned to the closest model image point (regardless of whether the latter had been previously assigned to another scene point), is

$$\mathrm{MSE} = \frac{\sum_{i=1}^{r} \|f(x_i) - y_i\|^2}{r}, \tag{8}$$

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

We provide the minimum, maximum, average, and standard deviation of the MSE values on 15 independent runs. Table 3 reports these values for the BrainWeb data set experiments obtained by the five IR algorithms. Another advantage of using the BrainWeb data set is that we know in advance the ground-truth solution of every IR problem instance. This value is shown in brackets together with the name of the transformation, so that we can compare it with the outcomes of every algorithm. Note that each number in this table is rounded up or down.

First, we observe that the MSE values obtained are progressively further 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 15 of the 16 cases, as well as the best minimum MSE value in 13 of them. Moreover, we should note 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 a single value for all the instances. However, we preferred to keep them unchanged for all instances to provide a robust method (as we did by selecting the same SS* variant for every case).

The poor results of the previous scatter search version, SS, related to the new one, SS*, show that SS does not use that image-derived information properly. At the same time, the low standard deviations show the robustness of SS*. In particular, we achieve one of our goals with respect to our previous work (Cordón et al. 2004), which is to design a competitive IR method even in a complex scenario of inter-subject medical IR.

Table 3 is split into four subtables considering every IR problem scenario. The best mean and minimum values are shown using the underlined bold font. The results obtained by each method can be compared with the optimal solution (in brackets) for every IR problem instance.

To show clearly the actual performance of each IR technique when tackling the previously introduced BrainWeb IR problems, Figure 3 collects the graphical representations of the real overlapping achieved by each IR algorithm in four of the 16 instances considered, one from each IR scenario. The first column of this figure shows the original IR problem instance to be solved by every method, while the next five columns correspond to the best registration estimations achieved by each IR algorithm (from left to right: I-ICP, ICP + SA, Dyn-GA, SS, and SS*). It can be seen how SS* always obtains the best registration (see the little resemblance of the images in the four central columns of the figure with regard to those in the rightmost column). 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 other IR technique that is able to solve the IR problem properly for the complex scenarios considered in the experimentation developed. The first column in Figure 3 corresponds to the original pose of the brains of four of the 16 IR instances, from top to bottom: $I_1$ versus $T_1(I_2)$, $I_1$ versus $T_2(I_3)$, $I_1$ versus $T_3(I_4)$, and $I_2$ versus $T_4(I_4)$. The next five columns correspond to

**Table 3 MSE Values Obtained by the Three State-of-the-Art IR Algorithms, Our Previous SS Contribution (Cordón et al. 2004), and Our Current SS* IR Method**

| | $I_1$ vs. $T_i(I_2)$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $T_1$ [32] | | | | | $T_2$ [21] | | | | |
| | I-ICP | ICP + SA | Dyn-GA | SS | SS* | I-ICP | ICP + SA | Dyn-GA | SS | SS* |
| $m$ | 344 | 247 | 101 | 45 | **35** | 131 | 131 | 44 | 40 | **37** |
| $M$ | — | 344 | 264 | 202 | 40 | — | 131 | 284 | 155 | 50 |
| $\mu$ | — | 307 | 195 | 142 | **37** | — | 131 | 108 | 98 | **43** |
| $\sigma$ | — | 38 | 51 | 40 | 2 | — | 0 | 52 | 30 | 4 |
| | $T_3$ [32] | | | | | $T_4$ [47] | | | | |
| $m$ | 894 | 457 | 87 | 69 | **57** | 632 | 283 | 139 | 85 | **49** |
| $M$ | — | 711 | 678 | 192 | 67 | — | 611 | 600 | 344 | 59 |
| $\mu$ | — | 559 | 211 | 139 | **63** | — | 465 | 302 | 210 | **54** |
| $\sigma$ | — | 81 | 137 | 32 | 3 | — | 101 | 121 | 58 | 3 |
| | $I_1$ vs. $T_i(I_3)$ | | | | | | | | | |
| | $T_1$ [43] | | | | | $T_2$ [30] | | | | |
| $m$ | 518 | 305 | 132 | 132 | **90** | 330 | 237 | 56 | 99 | **50** |
| $M$ | — | 432 | 741 | 278 | 132 | — | 297 | 534 | 180 | 66 |
| $\mu$ | — | 343 | 299 | 201 | **112** | — | 261 | 154 | 133 | **57** |
| $\sigma$ | — | 32 | 144 | 45 | 12 | — | 18 | 114 | 23 | 4 |
| | $T_3$ [43] | | | | | $T_4$ [62] | | | | |
| $m$ | 438 | 279 | 139 | 71 | **43** | 478 | 336 | 221 | 119 | **112** |
| $M$ | — | 389 | 839 | 264 | 235 | — | 429 | 841 | 431 | 143 |
| $\mu$ | — | 347 | 326 | 188 | **64** | — | 382 | 354 | 315 | **123** |
| $\sigma$ | — | 33 | 174 | 59 | 46 | — | 24 | 147 | 86 | 8 |
| | $I_1$ vs. $T_i(I_4)$ | | | | | | | | | |
| | $T_1$ [46] | | | | | $T_2$ [30] | | | | |
| $m$ | 704 | 236 | 124 | **89** | 149 | 1,493 | 314 | **48** | 88 | 51 |
| $M$ | — | 466 | 1,083 | 317 | 269 | — | 388 | 299 | 179 | 167 |
| $\mu$ | — | 385 | 255 | 231 | **184** | — | 359 | 163 | 136 | **89** |
| $\sigma$ | — | 61 | 228 | 50 | 33 | — | 22 | 58 | 31 | 41 |
| | $T_3$ [46] | | | | | $T_4$ [67] | | | | |
| $m$ | 951 | 312 | 158 | 67 | **52** | 416 | 342 | 207 | 134 | **95** |
| $M$ | — | 433 | 468 | 298 | 227 | — | 413 | 1,222 | 407 | 375 |
| $\mu$ | — | 381 | 225 | 207 | **82** | — | 367 | 415 | 301 | **154** |
| $\sigma$ | — | 43 | 87 | 68 | 45 | — | 17 | 258 | 77 | 86 |
| | $I_2$ vs. $T_i(I_4)$ | | | | | | | | | |
| | $T_1$ [29] | | | | | $T_2$ [18] | | | | |
| $m$ | 237 | 230 | **108** | 112 | 128 | 341 | 142 | 58 | 59 | **52** |
| $M$ | — | 237 | 348 | 347 | 298 | — | 341 | 270 | 209 | 188 |
| $\mu$ | — | 236 | **178** | 231 | 193 | — | 268 | 106 | 142 | **75** |
| $\sigma$ | — | 2 | 60 | 64 | 62 | — | 71 | 51 | 49 | 41 |
| | $T_3$ [29] | | | | | $T_4$ [45] | | | | |
| $m$ | 609 | 399 | 110 | 96 | **70** | 1,588 | 962 | 164 | 119 | **105** |
| $M$ | — | 439 | 611 | 309 | 278 | — | 1,533 | 751 | 414 | 362 |
| $\mu$ | — | 407 | 192 | 213 | **104** | — | 1,247 | 298 | 256 | **150** |
| $\sigma$ | — | 10 | 116 | 67 | 67 | — | 209 | 145 | 95 | 78 |

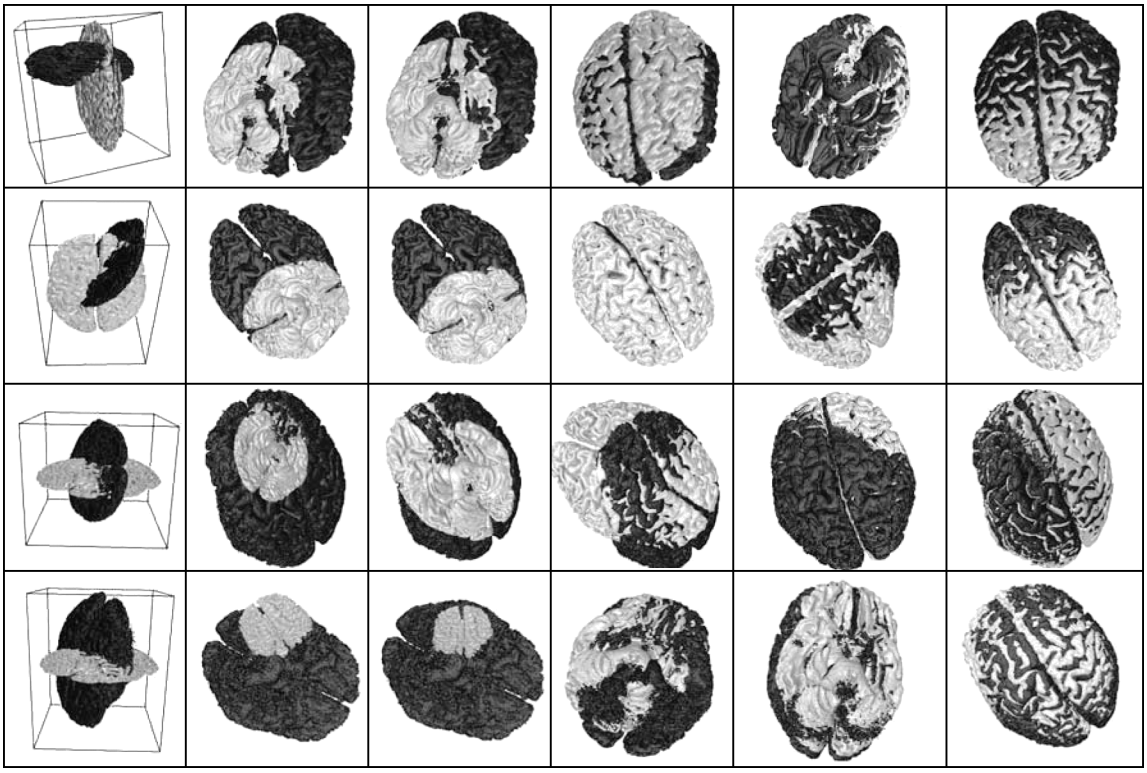

**Figure 3 Registration of Brain Images**

the best registration estimations achieved by each IR algorithm (from left to right: I-ICP, ICP+SA, Dyn-GA, SS, and SS*).

**4.2.3. SS* vs. Previous Methods in the CT Human Wrist Data Set.** Once the performance of our proposal has been validated against the simulated MRI data set and compared with the IR techniques considered, the final goal of our work regarding the consideration of nonsimulated and more complex anatomical medical images will be addressed in this subsection by using the second CT data set.

In Table 4 and Figure 4 the quantitative and qualitative performance obtained by each of the IR methods considered in the four IR instances (see §4.1) are shown. In view of the statistics shown in Table 4, our SS* proposal exhibits more robust behavior according to the MSE mean and standard deviation values. This difference is especially important when comparing the MSE mean value of SS* and the rest of the algorithms. Specifically, the SS* method obtains a mean MSE value of 18, 17, 18, and 19 for $T_1$, $T_2$, $T_3$, and $T_4$, respectively, while ICP_SA obtains 51, 40, 49, and 28, and Dyn_GA obtains 53, 52, 58, and 51. This fact, together with the low standard deviation values, shows that SS* behaves regularly and accurately in the 15 runs performed. Table 4 is split into four subtables considering every IR problem instance. The best mean and minimum values are shown using the underlined bold font. Nevertheless, the Dyn-GA method achieves the best individual estimation as revealed by the minimum MSE value. This is due to the fact that the objective function of the original proposal of Dyn-GA takes into account the median square error, favoring better performance in severe occluding IR instances, as in the CT data set. This suggests developing new extensions for the current SS proposal with the intention of achieving more accurate estimations by considering more robust mechanisms that are able to deal with these kinds of IR

**Table 4 MSE Values Obtained by the Three State-of-the-Art IR Algorithms and Our Current SS* IR Method**

| | $I_6$ vs. $T_i(I_5)$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $T_1$ | | | | $T_2$ | | | |
| | I-ICP | ICP + SA | Dyn-GA | SS* | I-ICP | ICP + SA | Dyn-GA | SS* |
| *m* | 52 | 51 | **6** | 12 | 42 | 36 | **6** | 12 |
| *M* | — | 52 | 92 | 49 | — | 41 | 87 | 30 |
| $\mu$ | — | 51 | 53 | **18** | — | 40 | 52 | **17** |
| $\sigma$ | — | 0 | 25 | 9 | — | 2 | 26 | 6 |
| | $T_3$ | | | | $T_4$ | | | |
| *m* | 55 | 45 | **6** | 12 | 37 | 23 | **6** | 10 |
| *M* | — | 52 | 90 | 49 | — | 37 | 86 | 49 |
| $\mu$ | — | 49 | 58 | **18** | — | 28 | 51 | **19** |
| $\sigma$ | — | 2 | 25 | 9 | — | 5 | 24 | 11 |

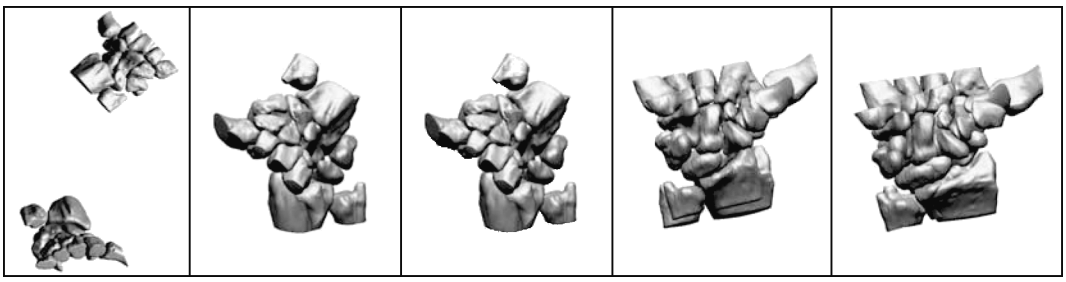

**Figure 4 Original Pose and Best Registration Estimations of IR Problem $T_1$**

instances. In Figure 4, from left to right, the first image corresponds to the original pose of the first IR problem considering $T_1$. The next four images correspond to the best registration estimations achieved by each IR algorithm (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 with respect to our previous proposal in Cordón et al. (2004) by implementing advanced reference-set designs as well as by strategically including context information derived from the images' features. This heuristic information is incorporated into an improved solution evaluation, candidate list strategies within the local search method, and into the diversification-generation method, so that we can deal with significantly more complex problem instances. Indeed, one of the main goals of our effort has been to test the proposed procedure by employing real-world data in realistic scenarios. To make a valid comparison with competing procedures, we have used the well-established MSE metric as well as graphical outputs.

Our computational experiments showed that SS* is of merit when compared with IR procedures previously identified as being the best. More specifically, when dealing with the realistic scenarios, we have seen how our SS* IR method achieves the best mean performance in 15 of the 16 cases, as well as the best minimum MSE value in 13 of them. The poor results of the old SS version showed that it did not make a suitable use of image-derived information. On the other hand, when tackling the real-world CT images, our proposal is the most robust method. This outcome can even be improved to obtain more accurate results by means of more suitable mechanisms for this specific class of IR instances (i.e., new objective functions, more complex transformations, hierarchical approaches, etc.).

We are planning to tackle the IR problem of 3D range images under the point-matching approach for the 3D model reconstruction of real-life objects, which is an emerging topic in computer graphics. We also want to extend our work to medical scenarios where real-time registration is a major concern. To do so, we will compare the outcomes of the present matching-space SS* contribution, as well as other previous proposals in the transformation-parameter space (Cordón et al. 2006a, b). These latter approaches could be redesigned to become more competitive in terms of computation time.

### Acknowledgments

Research by the first three authors was supported by the *Ministerio de Educación y Ciencia* (Ref. TIN2006-00829), including European Regional Development Fund fundings. Research by the last author was partially supported by the *Ministerio de Educación y Ciencia* (Ref. TIN2006-02696). The authors acknowledge Professor J. J. Crisco, supported by Grant NIH AR44005, for providing them with the CT images.

### References

Arun, K. S., T. Huang, S. D. Blostein. 1987. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Machine Intelligence* **9** 698–700.

Besl, P. J., N. D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intelligence* **14** 239–256.

Brown, L. G. 1992. A survey of image registration techniques. *ACM Comput. Surveys* **24** 325–376.

Campos, V., F. Glover, M. Laguna, R. Martí. 2001. An experimental evaluation of a scatter search for the linear ordering problem. *J. Global Optim.* **21** 397–414.

Chow, C. K., H. T. Tsui, T. Lee. 2004. Surface registration using a dynamic genetic algorithm. *Pattern Recognition* **37** 105–117.

Cordón, O., S. Damas. 2006. Image registration with iterated local search. *J. Heuristics* **12** 73–94.

Cordón, O., S. Damas, J. Santamaría. 2004. A scatter search algorithm for the 3D image registration problem. X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiño, A. Kabán, H. P. Schwefel, eds. *Parallel Problem Solving from Nature—PPSN VIII, 8th Internat. Conf., Birmingham, UK, September 18–22. Lecture Notes in Computer Science*, Vol. 3242. Springer, Heidelberg, Germany, 471–480.

Cordón, O., S. Damas, J. Santamaría. 2006a. A fast and accurate approach for 3D image registration using the scatter search evolutionary algorithm. *Pattern Recognition Lett.* **27** 1191–1200.

Cordón, O., S. Damas, J. Santamaría. 2006b. Feature-based image registration by means of the CHC evolutionary algorithm. *Image Vision Comput.* **24** 525–533.

Cordón, O., S. Damas, J. Santamaría, R. Martí. 2005. 3D inter-subject medical image registration by scatter search. M. J. Blesa, C. Blum, A. Roli, M. Sampels, eds. *Hybrid Metaheuristics 2005—HM 2005, 2nd Internat. Workshop, Barcelona, Spain, August 29–30. Lecture Notes in Computer Science*, Vol. 3636. Springer, Heidelberg, Germany, 90–103.

Cotta, C., J. M. Troya. 1998. Genetic forma recombination in permutation flowshop problems. *Evolutionary Comput.* **6** 25–44.

Eisert, P., E. Steinbach, B. Girod. 2000. Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views. *IEEE Trans. Circuits Systems Video Tech.* **10** 261–277.

Feldmar, J., N. Ayache. 1996. Rigid, affine and locally affine registration of free-form surfaces. *Internat. J. Comput. Vision* **18** 99–119.

Gagnon, H., M. Soucy, R. Bergevin, D. Laurendeau. 1994. Registration of multiple range views for automatic 3-D model building. *IEEE Conf. Comput. Vision and Pattern Recognition, Seattle, WA, June 21–23*. IEEE Press, Washington, D.C., 581–586.

Glover, F. 1977. Heuristics for integer programming using surrogate constraints. *Decision Sci.* **8** 156–166.

Glover, F. 1998. A template for scatter search and path relinking. J. K. Hao, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers, eds. *Artificial Evolution. Lecture Notes in Computer Science*, Vol. 1363. Springer, Heidelberg, Germany, 13–54.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Boston, MA.

Goldberg, D. E., R. Lingle. 1985. Alleles, loci, and the traveling salesman problem. J. J. Greffenstette, ed. *First Internat. Conf. Genetic Algorithms*. Lawrence Erlbaum Associates, Mahwah, NJ, 154–159.

Goshtasby, A. A. 2005. *2-D and 3-D Image Registration for Medical, Remote Sensing, and Industrial Applications*. Wiley Interscience, Hoboken, NJ.

Hart, W. E. 1994. Adaptive global optimization with local search. PhD thesis, University of California, San Diego, San Diego.

He, R., P. A. Narayana. 2002. Global optimization of mutual information: Application to three-dimensional retrospective registration of magnetic resonance images. *Computerized Medical Imaging Graphics* **26** 277–292.

Herrera, F., M. Lozano, D. Molina. 2006. Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies. *Eur. J. Oper. Res.* **169** 450–476.

Horn, B. K. P. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Optical Soc. America A* **4** 629–642.

Laguna, M., R. Martí. 2003. *Scatter Search—Methodology and Implementations in C*. Kluwer Academic Publishers, Boston.

Laguna, M., R. Martí, V. Campos. 1999. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Comput. Oper. Res.* **26** 1217–1230.

Liu, Y. 2004. Improving ICP with easy implementation for free-form surface matching. *Pattern Recognition* **37** 211–226.

Lozano, M., F. Herrera, N. Krasnogor, D. Molina. 2004. Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Comput.* **12** 273–302.

Luck, J., C. Little, W. Ho. 2000. Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. *IEEE Internat. Conf. Robotics and Automation, San Francisco, CA, April 24–28*. IEEE Press, Washington, D.C., 3739–3744.

Marai, G. E., D. H. Laidlaw, J. J. Crisco. 2006. Super-resolution registration using tissue-classified distance fields. *IEEE Trans. Medical Imaging* **25** 177–187.

Matsopoulos, G. K., N. A. Mouravliansky, K. K. Delibasis, K. S. Nikita. 1999. Automatic registration of retinal images with global optimization techniques. *IEEE Trans. Inform. Tech. Bio-Engrg.* **3** 47–60.

Monga, O., R. Deriche, G. Malandain, J. P. Cocquerez. 1991. Recursive filtering and edge tracking: Two primary tools for 3D edge detection. *Image Vision Comput.* **9** 203–214.

Rangarajan, A., H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. S. Goldman-Rakic, J. S. Duncan. 1997. A robust point matching algorithm for autoradiograph alignment. *Medical Image Anal.* **1** 379–398.

Resende, M. G. C., C. C. Ribeiro. 2001. Greedy randomized adaptive search procedures. F. Glover, G. Kochenberger, eds. *State-of-the-Art Handbook in Metaheuristics*. Kluwer Academic Publishers, Boston, 219–250.

Robertson, C., R. B. Fisher. 2002. Parallel evolutionary registration of range data. *Comput. Vision Image Understanding* **87** 39–50.

Robinson, D., P. Milanfar. 2004. Fundamental performance limits in image registration. *IEEE Trans. Image Processing* **13** 1185–1199.

Rouet, J. M., J. J. Jacq, C. Roux. 2000. Genetic algorithms for a robust 3-D MR-CT registration. *IEEE Trans. Inform. Tech. Biomedicine* **4** 126–136.

Santamaria, J. 2006. Scatter Search para el registrado de imágenes 3D: Aplicación en antropología forense (in Spanish). PhD thesis, University of Granada, Granada, Spain.

Wachowiak, M. P., R. Smolíkova, Y. Zheng, J. M. Zurada, A. S. Elmaghraby. 2004. An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Trans. Evolutionary Comput.* **8** 289–301.

Weik, S. 1997. Registration of 3-D partial surface models using luminance and depth information. *Internat. Conf. Recent Adv. 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, May 12–15*, IEEE Press, Washington, D.C., 93–100.

Yamany, S. M., M. N. Ahmed, A. A. Farag. 1999. A new genetic-based technique for matching 3D curves and surfaces. *Pattern Recognition* **32** 1817–1820.

Yuille, A. L., J. J. Kosowsky. 1994. Statistical physics algorithms that converge. *Neural Comput.* **6** 341–356.