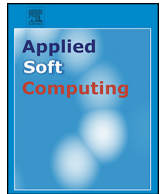




Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc



Prediction of relative position of CT slices using a computational intelligence system

Mauro Castelli^{a,*}, Leonardo Trujillo^b, Leonardo Vanneschi^a, Aleš Popovič^{a,c}

^a NOVA IMS, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal

^b Tree-Lab, Posgrado en Ciencias de la Ingeniería, Instituto Tecnológico de Tijuana, Blvd. Industrial y Ave. ITR Tijuana S/N, Mesa de Otay, Tijuana, B.C. 22500, Mexico

^c University of Ljubljana, Faculty of Economics, Kardeljeva ploščad 17, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:

Received 12 March 2015
Received in revised form 3 September 2015
Accepted 13 September 2015
Available online xxx

Keywords:

Computerized tomography
Radiology
Genetic programming
Semantics
Local search

ABSTRACT

One of the most common techniques in radiology is the computerized tomography (CT) scan. Automatically determining the relative position of a single CT slice within the human body can be very useful. It can allow for an efficient retrieval of slices from the same body region taken in other volume scans and provide useful information to the non-expert user. This work addresses the problem of determining which portion of the body is shown by a stack of axial CT image slices. To tackle this problem, this work proposes a computational intelligence system that combines semantics-based operators for Genetic Programming with a local search algorithm, coupling the exploration ability of the former with the exploitation ability of the latter. This allows the search process to quickly converge towards (near-)optimal solutions. Experimental results, using a large database of CT images, have confirmed the suitability of the proposed system for the prediction of the relative position of a CT slice. In particular, the new method achieves a median localization error of 3.4 cm on unseen data, outperforming standard Genetic Programming and other techniques that have been applied to the same dataset. In summary, this paper makes two contributions: (i) in the radiology domain, the proposed system outperforms current state-of-the-art techniques; (ii) from the computational intelligence perspective, the results show that including a local searcher in Geometric Semantic Genetic Programming can speed up convergence without degrading test performance.

© 2015 Elsevier B.V. All rights reserved.

1. Background

Scanning large parts of a patient's body with computerized tomography (CT) is common practice in radiology. As reported in [1], the amount of image data resulting from a full body scan varies between 40 MB to more than 1 GB, which has to be stored in a medical picture archiving and communication system (PACS). The increasing amount of data poses various problems for physicians and the PACS. A clinician often needs to compare different scans of the same body region for differential diagnoses or needs to compare disease patterns in the same body region between similar patients. In conventional systems, the PACS needs to load the complete dataset to retrieve the required sub-region and this causes an unnecessary load to both the system and the network. In other cases, a physician only receives a single CT slice as part

of a radiology report and does not have access to the complete CT volume set due to constraints caused by the network bandwidth or patient privacy policies. When searching for similar disease patterns in a remote PACS it is impractical to load complete volume sets and then manually navigate to the relevant body regions. Instead, transferring only the most similar body parts is more reasonable, secure and efficient. Nonetheless, in both scenarios it is necessary to determine the relative position of the given CT slice within the body.

This work addresses the problem of determining which portion of the body is shown by a stack of axial CT image slices. For example, as reported in [2], given a small stack of slices, one may want to automatically determine to what part of the body it belongs. If we correctly determine the correct body region or part, such as the heart, then similar slices can be transferred to the user. Another possible application is pruning the search space for image analysis algorithms, such as automatic organ detectors.

As reported in [1], deriving the required information from the DICOM [3] header of the slice is often not a reliable option. Entries like "patient position" or "body part examined" are often imprecise or even wrong. Additionally, single slices are often embedded

* Corresponding author. Tel.: +351 213828610; fax: +351 213872611.

E-mail addresses: mcastelli@novaims.unl.pt (M. Castelli), leonardo.trujillo@tectijuana.edu.mx (L. Trujillo), lvanneschi@novaims.unl.pt (L. Vanneschi), apopovic@novaims.unl.pt (A. Popovič).

into report documents, so that the header information or metadata is either lost or not easily accessible. Due to these reasons, parsing DICOM headers does not yield a viable solution for obtaining the relative position of a single CT slice. All these critical aspects highlight the need of reliable and accurate techniques that can allow physicians to automatically estimate the relative position of a single CT slice.

Therefore, the objective of this work is to develop a Genetic Programming (GP) system for the prediction of the relative position on a single CT slice. Moreover, the system should have the following properties: (i) it should be able to find (quasi-)optimal solutions with high probability; (ii) it should generate models able to produce near-optimal predictions on unseen data (test instances).

GP is one of the most successful methods in computational intelligence (CI). It has obtained excellent results on a large number of complex real-life applications [4], and recently an important breakthrough was made: the definition of geometric semantic operators (GSOs), new search operators that induce a unimodal error surface on any supervised learning problem (including regression and classification). Eliminating local optima, GSOs have a stronger problem solving ability than traditional search operators in GP. However, much work has still to be done in order to use GSOs in complex real-world applications like the one considered in this paper. In particular, GSOs converge to optimal solution(s) very slowly, an important limitation when dealing with large amounts of data. Therefore, this paper proposes a CI system that combines GSOs with a local search algorithm, coupling the exploration ability of GSOs with the exploitation ability of the local searcher. The proposed system is able to find (near-)optimal solutions faster than standard GP, and GP based on GSOs.

1.1. Related work

Although localizing a CT slice within a human body can aid and simplify the work of a physician, so far this area of research has not received much attention. The localization problem could be solved by using the metadata of the DICOM header of a CT image. However, the available information is often erroneous. For instance [5], reports that several entries in the DICOM header are often imprecise or even completely wrong. To find a more appropriate solution to the problem, the research community proposed some methods for registering slices to a general atlas with standardized height. The general approach is to localize invariant landmarks as starting points and from there to interpolate and define a relative coordinate system. The proposal introduced by [6] creates a navigation table using eight landmarks which are detected in various ways, while [7] proposed a method to detect invariant slices and single point landmarks in full body scans, capable of detecting up to 19 salient and robust landmarks within a volume scan. However, these methods cannot be used for localizing a single slice since they operate with full body scans only. Recently, approaches to determine the body region from a topogram appeared. In [8], the proposed system predicts the body region from a topogram based on landmarks with invariant positions. Authors of [1] proposed a new method that uses a single slice of a CT scan for registration by using a multi-represented feature descriptor (MR-descriptor). A similar localization problem has been addressed in [2], using visual words and SURF (Speeded Up Robust Features) features. Visual words are code words of a quantized feature space and can be thought of as classes of image patches with similar appearance. A slice descriptor is formed by sampling a slice on a regular 2D grid and extracting a SURF descriptor at each sample point. Each SURF descriptor extracted from a slice is classified into the closest visual word (or cluster center) and counted in a histogram. A slice is finally described by a spatial pyramid of such histograms. Other

works involving the use of computational intelligence methods to tackle medical images problems can be found in [9,10].

2. Method

The field of evolutionary computation is devoted to the development of search and optimization algorithms based on the core principles of Neo-Darwinian evolutionary theory [11]. Evolutionary algorithms are population-based meta-heuristics, where candidate solutions are stochastically selected and modified to produce new, and possibly better, solutions for a particular problem. In particular, in standard GP each individual is encoded using a tree structure, also known as a program tree, which can express a simple computer program, function, or operator [12]. Individual trees are built using elements from two finite sets, internal nodes contain basic functions from a Function set F , and leaves contain the input variables of the program, taken from a Terminal set T . The basic GP algorithm, like every evolutionary algorithm, consists of the basic loop depicted in Fig. 1. First, a random population of solutions (program trees) are generated; then each individual is evaluated based on a fitness or objective function; afterwards, individuals are selected based on their fitness scores; and finally, genetic operators (or search operators) are used to generate new candidate solutions, these are subtree mutation and subtree crossover as defined by [12]. The process terminates when a user defined stopping criterion is met, such as a maximum number of iterations (generations) or when the system has converged to a predefined fitness value. The algorithm then returns the solution (program tree) with the best fitness value. Hereafter, we will refer to this algorithm as standard GP (STGP) [12].

In the last few years, a rising topic in the field of GP has been the use of semantic methods [13]. These methods provide a way of exploring the input–output behavior of programs, which is ultimately what matters for problem solving. Semantics is usually defined as the vector of outputs obtained after the evaluation of a GP individual on a set of fitness cases (training set). The topic of semantics contrasts with most previous work in GP, where genetic operators work on the syntax of the individuals to transform the program code, while the effect on program output is indirect. Anyway, as discussed in [13], some existing semantic methods are also indirect, in the sense that they create new individuals using traditional syntax-based operators and, as a second step, accept the newly created individuals if and only if some semantic criterion is satisfied. The introduction of a direct method has been proposed in [14], where the authors defined geometric semantic operators (GSOs) for GP. The objective of GSOs is to define modifications on the syntax of GP individuals that have a precise correspondence on their semantics.

Given the definition of semantics, we can identify the semantics of an individual as a point in a multi-dimensional space, that we call semantic space. The idea introduced in [14] is to define syntax transformations that correspond to well known operators of genetic algorithms (GAs). In this way, GP could “inherit” the known properties of those GA operators. Furthermore, contrarily to what typically happens in real-valued GAs or other heuristics, in the GP semantic space the target point is also known (it corresponds to the vector of expected output values of the supervised learning problem) and the fitness of an individual is given by the distance between the point it represents in semantic space and the target point. Moreover, it has been shown that analyzing the problem in semantic space reveals a unimodal error surface with a single global optima [14]. The real-valued GA operators that we want to “map” into the GP semantic space are *geometric crossover* and *ball mutation*. In real-valued GAs, geometric crossover produces an offspring that stands in the segment that joins the parents. It was proven

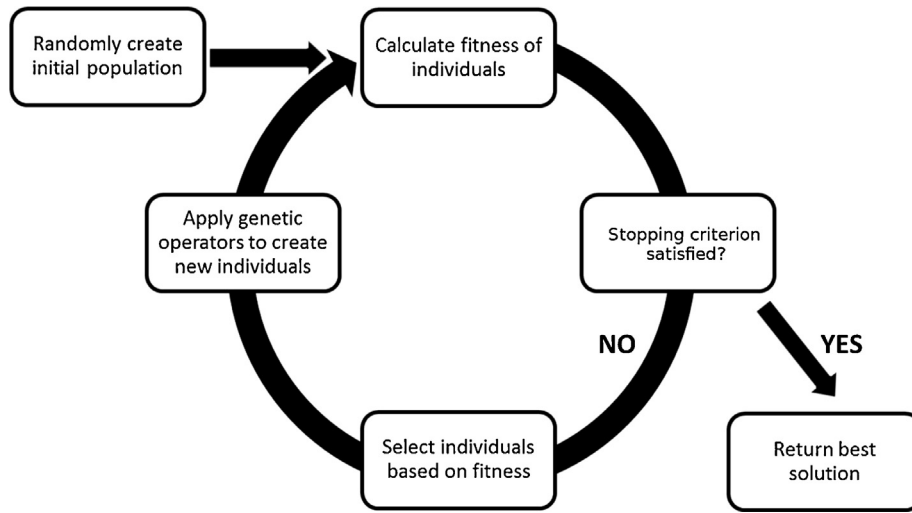


Fig. 1. The basic genetic programming algorithm.

in [15] that in cases where the fitness is a direct function of the distance to the target (like the case we are interested in here) the offspring cannot have a worse fitness than the worst of its parents. Ball mutation consists in a random perturbation of the coordinates of an individual. The definitions of the GSOs that correspond to geometric crossover and ball mutation in the GP semantic space are, as given in [14], respectively:

Definition Geometric Semantic Crossover (GSC). Given two parent functions $T_1, T_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, the geometric semantic crossover returns the real function $T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$, where T_R is a random real function whose output values range in the interval $[0, 1]$.

Definition Geometric Semantic Mutation (GSM). Given a parent function $T : \mathbb{R}^n \rightarrow \mathbb{R}$, the geometric semantic mutation with mutation step ms returns the real function $T_M = T + ms \cdot (T_{R1} - T_{R2})$, where T_{R1} and T_{R2} are random real functions.

A GP search that uses these GSOs, but in all other aspects is basically the same as the one depicted in Fig. 1, is hereafter referred to as Geometric Semantic GP (GSGP). In this work we use the GSGP implementation defined in [16], for which the computational cost of evolving a population of n individuals for g generations is $O(ng)$, while the cost of evaluating a new, unseen, instance is $O(g)$. The pseudocode of the GSGP algorithm is given in Algorithm 1.

Algorithm 1. Pseudocode of the GSGP algorithm.

```

/bf Input: N = number of individuals in the population
/bf Output: The best solution found
//Initialization
p_cross = crossover probability
p_mut = mutation probability
Randomly create a population P
Evaluate the objective function for all the individuals
best ← best individual in P
//Main loop of GP
/bf for g ← 1 /bf to G /bf do
  Create an empty population P'
  Copy best in P'
  for i ← 2 to N /bf do
    c = rand()
    //rand() returns a random number in the range [0;1]
    if c < p_cross then
      T1 ← tournament_selection(P)
      T2 ← tournament_selection(P)
      //Apply crossover
      TXO ← GSC(T1, T2)
      T ← TXO
    end
  end
end

```

```

else if c > p_mut & & c < p_mut + p_cross then
  T1 ← tournament_selection(P)
  //Apply mutation
  Tm ← GSM(T1)
  T ← Tm
end
else
  T1 ← tournament_selection(P)
  T ← T1
end
P[i] ← T
end
Evaluate the objective functions for the individuals in P'
Replace P with P'; Clear P'
best ← best individual in P
end
return best;

```

2.1. Local search in geometric semantic operators

In this work we integrate a local search (LS) strategy within GSGP and, in particular, we include a local searcher within the GSM mutation operator. In particular, the GSM with LS (GSM-LS) of a tree T generates an individual:

$$T_M = \alpha_0 + \alpha_1 \cdot T + \alpha_2 \cdot (T_{R1} - T_{R2}) \tag{1}$$

where $\alpha_i \in \mathbb{R}$; notice that α_2 replaces the mutation step parameter ms of GSM. This in fact defines a basic multivariate linear regression problem, which could be solved, for example, by Ordinary Least Square regression (OLS). However, in this case we have n linear equations, the number of fitness cases, and only three unknowns (the α_i s). This gives an overdetermined multivariate linear fitting problem, which can be solved through SVD.¹ We argue that this should be seen as a LS operator, that attempts to determine the best linear combination of the parent tree and the random trees. It should not be seen as a LS in the entire semantic space, since in that case the LS would necessarily converge to the optimal in this unimodal landscape.

This approach is similar to two previously proposed approaches. First, the linear fitting problem is reminiscent of the linear scaling procedure proposed in [17], which allows GP to fit the form of the desired output without necessarily optimizing the scale or bias. However, in that case the scaling process is only used to

¹ In this work, the GNU Scientific Library is used <http://www.gnu.org/software/gsl/>.

adjust the fitness value of each individual, while the search operators used are standard ones. Second, and more closely related, the non-isotropic Gaussian mutation proposed in [18], that is used to perform a runtime analysis of GSGP. However, the mutation proposed in that work considers a fixed set of basis functions instead of randomly generated GP trees, and perturbs the linear combination with Gaussian-noise instead of providing the best fit coefficients. Finally, the work presented by [19] also uses a multivariate linear regression approach to optimize evolved solutions, with several key differences. Particularly, the search is conducted by standard GP, not GSGP, and each tree is decomposed into a set of subtrees which are then linearly combined. The method is much more explorative than the one presented here.

Moreover, this approach contrasts with previous work [20], that relied on a non-linear local optimizer, since the linear assumption is mostly not satisfied by the expression evolved with standard GP and the corresponding parametrization. On the other hand, in the approach presented here, it is simple to apply a linear regression optimizer, given that the GSM operator defines a linear expression in parameter space.

3. Experiments, results and discussion

3.1. Data set information

Each CT slice is described by a compound radial image descriptor, which is generated using the following steps: unifying the image resolutions, extracting the patient's body and combining the two image descriptors to a single radial descriptor. The complete process used to build the dataset is explained with rigorous detail in [21], but can be summarized as follows. The data was retrieved from 53,500 CT images taken from 74 different patients (43 male, 31 female). Each CT slice is described by two histograms in polar space. The first histogram describes the location of bone structures in the image, while the second describes the location of air inclusions inside of the body. Both histograms are concatenated to form the final feature vector. Bins that are outside of the image are marked with the value -0.25 . The target variable (relative location of an image on the axial axis) was constructed by manually annotating up to 10 different distinct landmarks in each CT volume with known location. The location of slices in between landmarks was interpolated.

Each CT image is described by 385 features. The first feature is the ID of the patient; features 2–241 are related to the histogram describing bone structures; features 242–385 are related to the histogram describing air inclusions. The last feature is the target variable that is the relative location of the image on the axial axis. Values are in the range $[0; 180]$ where 0 denotes the top of the head and 180 the soles of the feet. This dataset was also used in [21,1], which allows us to compare our method with previous attempts at solving the problem of predicting the relative location of a CT slice.

3.2. Experimental settings

Three different systems were experimentally compared: (1) the standard syntax-based GP (STGP); (2) the GSGP algorithm; and (3) a GSGP algorithm that uses the GSM-LS mutation operator defined in Section 2.1, referred to as LSGP. The shared parameter values of all three algorithms are summarized in Table 1. The parameter values have been chosen after a preliminary tuning phase where several combinations were tested, in order to make the comparison between STGP and GSGP as fair as possible. Survival from one generation to the other was always guaranteed to the best individual of the population (elitism). For GSM a random mutation step has been considered in each mutation event.

Table 1
Shared algorithm parameters for all GP algorithms.

Parameters	Description and values
Population	200 individuals
Stopping criterion	2000 generations
Initialization	Ramped Half-and-Half [12]
Max. tree depth	initial depth 6 levels
Crossover & mutation prob.	Crossover $p_c = 0.7$; Mutation $p_\mu = 0.3$
GSM mutation step	$ms \in U(0, 1)$
Selection	Tournament of size 6
Survival	Elitism, best solution always survives
Function set	Arithmetic operators, $F = \{+, -, *, /\}$, division is protected [12]
Terminal set	385 variables, each one corresponding to a different feature

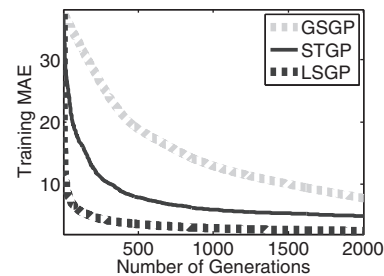


Fig. 2. Training MAE. The plot shows the median over 50 independent runs.

A parameter of particular importance is the stopping criterion. This is a critical factor for evolutionary algorithms in particular, and in general for any iterative learning algorithm. If the search is terminated too early the solution returned may have bad quality, while if the search goes on for too long the system might overfit the training examples. One approach is to stop the algorithm after the best training error has not improved after a specified number of generations. However, since each algorithm converges differently, the computational cost of one search process might not be comparable to another. Therefore, in this work we set the stopping criterion to a maximum of 2000 generations, after experimental results (reported in Fig. 2) showed that at least one of the algorithms converged to a near perfect training error at that point.

A total of 50 runs were performed with each technique: this is a fundamental aspect given the stochastic nature of the considered systems. In each run, a different partition between training and test data has been considered. In particular 70% of the instances have been used as training data, while the remaining have been used as test data. In partitioning the dataset we ensured that slices of the same scan belong either to the training or the test set. Finally, for all the considered systems, performance and fitness is given by the mean absolute error (MAE), a minimization problem defined as

$$MAE(T) = \frac{1}{N} \sum_{i \in Q} |t_i - y_i| \quad (2)$$

where $y_i = T(\mathbf{x}_i)$ is the output of the GP individual T on the input data \mathbf{x}_i , t_i is the target value associated with \mathbf{x}_i , N denotes the number of samples in the training or testing subset, and Q contains the indices of that set. The units of the reported results are centimeters (cm).

In the next section, the obtained experimental results are reported using curves of the median error on the training and test sets. In particular, at each generation the best individual in the population (i.e. the one that has the smaller training error) has been chosen and the value of its error on the training and test sets has been stored. The reported curves contain the median of all these values collected at each generation. The median was preferred over the mean because of its higher robustness to outliers. We also report

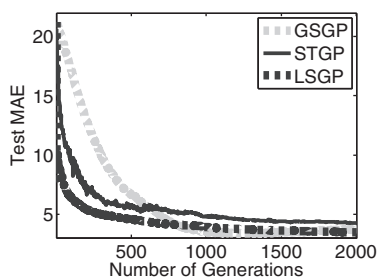


Fig. 3. Test MAE. The plot shows the median over 50 independent runs.

the median running time of each algorithm, to compare the computational cost of each method. Finally, the results discussed in the next section have been obtained using the GSGP implementation freely available at <http://gsgp.sourceforge.net> and documented in [22].

3.3. Discussion

Figs. 2 and 3 plot the training and test MAE for the considered GP systems against generations. These clearly show that LSGP outperforms GSGP and STGP on both training and test sets. In particular, it is interesting to distinguish performance on the training set from the one achieved on the test set. Regarding the training fitness, LSGP is able to reach a lower (i.e., better) MAE, while also converging in a smaller number of generations with respect to STGP and GSGP. The usage local search used by GSM-LS allows the search process to overcome one of the main limitations of GSGP, that is the large number of generations needed to convergence to optimal solutions. Regarding the performance on the test set, it is possible to draw similar conclusions: LSGP is not only the best performer, but it is also able to produce good quality solutions in a smaller number of generations with respect to GSGP and STGP. It is also interesting to report that LSGP quickly converges towards optimal solutions. Moreover, in these experiments none of the algorithms showed any noticeable overfitting on the training set. For instance, STGP and GSGP both show better test performance than training performance, while LSGP showed very similar performance on both sets. The lack of overfitting might be due to the number of generations used, but in an online scenario using less generations is desirable since it reduces the total run time.

To analyze the statistical significance of these results, a set of tests has been performed on the median errors. In particular, we want to assess whether the final results (after generation 2000) produced by the considered GP systems, have a statistically significant difference. As a first step, the Shapiro Wilk test (with $\alpha = 0.1$) has shown that the data are not normally distributed and hence a rank-based statistic is used to compare the algorithms. Therefore, the Mann–Whitney test (with a Bonferroni correction for the value of α) has been used under the alternative hypothesis that the samples have a different median.

The p -values from the statistical test are reported in Table 2. According to the p -values, we can clearly state that LSGP produces solutions that are significantly better (i.e. with lower error) than GSGP and STGP on both training and test instances, if we accept the alternative hypothesis at the $\alpha = 0.01$ significance level.

Table 2
 p -values obtained from the statistical validation procedure.

	Testing		Training	
	STGP	LSGP	STGP	LSGP
GSGP	0.065	1.48E–6	2.60E–11	2.60E–11
LSGP	0.003	–	2.60E–11	–

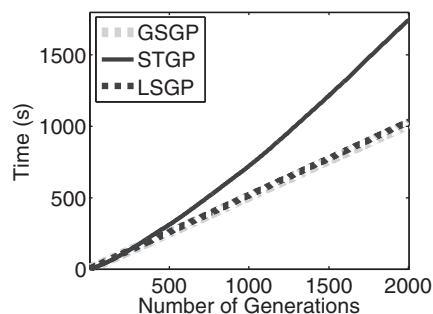


Fig. 4. CPU time of each method, reported as the median over 50 independent runs against generations.

Table 3
Median and standard deviation of the execution time (in seconds) for the three considered GP systems. Median calculated over 50 independent runs.

	Median	Standard deviation
GSGP	1013.51	17.1
STGP	1744.63	191.2
LSGP	1030.8	22.7

While LSGP produces good quality solutions, it is also important to consider the computational cost of each method, since in many application domains the time required to find a solution can be a critical factor. Fig. 4 shows the median CPU time required by each method, executed on a PC running 64-bit Windows 7 Professional® with an Intel® Core™ i7-4750HQ CPU @ 2.0 GHz and 16 GB of RAM. Additionally, Table 3 summarizes the median and standard deviation of the total execution time for each method. From these results, it is reasonable to state that both GSGP algorithms are substantially faster than STGP. Moreover, the use of the local search method by LSGP does not seem to reduce algorithm efficiency: GSGP and LSGP require a comparable CPU time.

3.4. Comparison with other machine learning techniques

In this section, results achieved with LSGP are compared against the ones achieved with other well-known state-of-the-art machine learning techniques on the prediction of the relative position of CT slices. This comparison will allow us to draw some conclusion about the competitiveness of the obtained results. In particular, we consider the techniques proposed in [23,24] and the best approach presented in [1] that reports, to the best of our knowledge, the smallest localization errors from previously published literature.

To perform the comparison with other techniques we used the implementations provided by the Weka public domain software [25]. As done for the previous experimental phase, a preliminary study has been performed in order to find the best tuning of the parameters for the techniques taken into account.

The results of the comparison are summarized in Table 4, which also shows the improvement percentage achieved by the best method (LSGP) on the test data. These results show that LSGP

Table 4
Experimental comparison between different techniques. Error on training and test data reported for each technique. Bold indicates best, and the fourth column shows the relative improvement achieved by the best method on the test data expressed as a percentage.

Method	Training	Test	Improvement
Neural networks [23]	17.08	15.32	77%
Isotonic regression [24]	9.91	12.41	72%
Best method proposed in [1]	N.A.	4.45	22%
STGP	4.89	4.25	19%
GSGP	7.8	3.70	7.5%
LSGP	2.52	3.44	–

performs better than all the other methods on both training and test data. In particular, it is interesting to note that LSGP outperforms the state-of-the-art method proposed in [1].

To summarize the results of the experimental phase, LSGP is a suitable technique for addressing the problem at hand. In particular, it is able to outperform STGP on both training and test instances and it is able to converge faster than GSGP. Moreover, solutions obtained with LSGP do not overfit the training set and are better than the ones achieved by other state-of-the-art techniques that predict the relative position of CT slices.

4. Conclusions

This paper proposes a computational intelligence system to automatically determine the relative position of a single CT slice within a full body scan. Knowing the relative position in a scan allows the efficient retrieval of similar slices from the same body region in other volume scans. Moreover, the relative position is often important information for a non-expert user that only has access to a single CT slice of a scan.

The proposed system is based on a variant of GP. In particular, the GP system makes use of particular genetic operators that, differently from the standard genetic operators used in GP, work on the semantics of the solutions. While the use of semantic methods in GP has been successfully investigated and applied, several important problems that do not allow to efficiently use these methods are still open. In particular, the GP system that uses semantics operators (GSGP) requires a large amount of generations to converge towards optimal solutions. In this perspective, this work integrated the GSGP framework with a local search optimizer. The use of a local searcher improved the convergence speed of GSGP, while not overfitting the training data. That is, by combining the exploration ability of GSGP with the exploitation ability of a local search method the proposed system outperformed state-of-the-art performance.

Experimental results, achieved using a large database of CT images, have shown the suitability of the proposed system for the studied problem. In particular, the new method provides a median localization error of 3.4 cm on unseen data, outperforming standard GP, the basic GSGP algorithm and all other existing state-of-the-art techniques for this application.

Future work will focus on two aspects. First, from an algorithmic perspective, the local search method can be included with the other GSOs, namely geometric semantic crossover. Second, from an application development perspective, the proposed system will be integrated into an online PACS, allowing us to evaluate the usability and the performance of the algorithm during online use. This will of course open up a series of interesting implementation details, such as the use of parallel or distributed architectures that can help make an evolutionary search process more efficient [26].

Acknowledgments

The authors acknowledge project MassGP (PTDC/EEL-CTP/2975/2012), FCT, Portugal.

References

- [1] T. Emrich, F. Graf, H.-P. Kriegel, M. Schubert, M. Thoma, A. Cavallaro, CT slice localization via instance-based regression, in: Proc. SPIE, vol. 7623, 2010, pp. 1–12.

- [2] J. Feulner, S. Zhou, E. Angelopoulou, S. Seifert, A. Cavallaro, J. Hornegger, D. Comaniciu, Comparing axial ct slices in quantized n-dimensional surf descriptor space to estimate the visible body region, *Comput. Med. Imaging Graph.* 35 (3) (2011) 227–236.
- [3] DICOM standard, 2015, <http://medical.nema.org/>.
- [4] J.R. Koza, Human-competitive results produced by genetic programming, *Genet. Programm. Evol. Mach.* 11 (3–4) (2010) 251–284.
- [5] M.O. Güld, M. Kohnen, D. Keysers, H. Schubert, B. Wein, J. Bredno, T.M. Lehmann, Quality of DICOM header information for image categorization, in: *Intl. Symposium on Medical Imaging*, Vol. 4685 of Proc. SPIE, San Diego, CA, 2002, pp. 280–287.
- [6] B. Haas, T. Coradi, M. Scholz, P. Kunz, M. Huber, U. Oppitz, L. André, V. Lengkeek, D. Huyskens, A. van Esch, R. Reddick, Automatic segmentation of thoracic and pelvic CT images for radiotherapy planning using implicit anatomic knowledge and organ-specific segmentation strategies, *Phys. Med. Biol.* 53 (6) (2008) 1751–1771.
- [7] S. Seifert, A. Barbu, S.K. Zhou, D. Liu, J. Feulner, M. Huber, M. Sühling, A. Cavallaro, D. Comaniciu, Hierarchical parsing and semantic navigation of full body CT data, in: *Proceedings of the SPIE*, 2009, pp. 1–8.
- [8] C. Bürger, Automatic Localisation of Body Regions in CT Topograms, VDM Publishing, Germany, 2008.
- [9] L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. Hoi, M. Satyanarayanan, A boosting framework for visibility-preserving distance metric learning and its application to medical image retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (1) (2010) 30–44.
- [10] J. Duncan, N. Ayache, Medical image analysis: progress over two decades and the challenges ahead, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (1) (2000) 85–106.
- [11] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, USA, 1992.
- [12] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [13] L. Vanneschi, M. Castelli, S. Silva, A survey of semantic methods in genetic programming, *Genet. Programm. Evol. Mach.* 15 (2) (2014) 195–214.
- [14] A. Moraglio, K. Krawiec, C.G. Johnson, Geometric semantic genetic programming, in: C.A. Coello Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, M. Pavone (Eds.), *Parallel Problem Solving from Nature, PPSN XII (Part 1)*, Vol. 7491 of Lecture Notes in Computer Science, Springer, Berlin, 2012, pp. 21–31.
- [15] K. Krawiec, P. Lichocki, Approximating geometric crossover in semantic space, in: *GECCO '09: Proceedings of the 11th Annual conference on Genetic and Evolutionary Computation*, ACM, Montreal, 2009, pp. 987–994.
- [16] L. Vanneschi, S. Silva, M. Castelli, L. Manzoni, Geometric semantic genetic programming for real life applications, in: *Genetic Programming Theory and Practice XI*, Springer, New York, 2014, pp. 191–209.
- [17] M. Keijzer, Improving symbolic regression with interval arithmetic and linear scaling, in: *Proceedings of the 6th European Conference on Genetic Programming, EuroGP'03*, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 70–82.
- [18] A. Moraglio, A. Mambrini, Runtime analysis of mutation-based geometric semantic genetic programming for basis functions regression, in: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13*, ACM, New York, NY, USA, 2013, pp. 989–996.
- [19] K. Krawiec, U.-M. O'Reilly, Behavioral programming: a broader and more detailed take on semantic GP, in: *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO '14*, ACM, New York, NY, USA, 2014, pp. 935–942.
- [20] E. Z-Flores, L. Trujillo, O. Schuetz, P. Legrand, Evaluating the effects of local search in genetic programming, in: A.-A. Tantar, et al. (Eds.), *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, no. 288 in *Advances in Intelligent Systems and Computing*, Springer, Berlin, 2014, pp. 213–228.
- [21] F. Graf, H.-P. Kriegel, M. Schubert, S. Pölsterl, A. Cavallaro, 2d image registration in CT images using radial image descriptors, in: *Proceedings of the 14th International Conference on Medical Image Computing and Computer-Assisted Intervention – Volume Part II, MICCAI'11*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 607–614.
- [22] M. Castelli, S. Silva, L. Vanneschi, A C++ framework for geometric semantic genetic programming, *Genet. Programm. Evol. Mach.* 16 (1) (2015) 73–81.
- [23] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, USA, 1999.
- [24] L. Hoffmann, *Multivariate Isotonic Regression and Its Algorithms*, Wichita State University, College of Liberal Arts and Sciences, Department of Mathematics and Statistics, Wichita, Kansas, 2009.
- [25] Weka Machine Learning Project, Weka, 2014, <http://www.cs.waikato.ac.nz/~ml/weka>.
- [26] F. Fernández De Vega, G. Olague, L. Trujillo, D. Lombrana González, Customizable execution environments for evolutionary computation using BOINC + virtualization, *Nat. Comput.: Int. J.* 12 (2) (2013) 163–177.