Parallel Image Registration using Bio-inspired Computing

Silviu-Ioan Bejinariu, Florin Rotaru, Cristina Niţă, Ramona Luca
Institute of Computer Science
Romanian Academy, Iasi Branch
Iaşi, Romania

Hariton Costin
Faculty of Medical Bioengineering,
"Gr. T. Popa" University of Medicine and Pharmacy, Iaşi,
Institute of Computer Science,
Romanian Academy, Iasi Branch
Iaşi, Romania

Abstract—**In this paper it is proposed a parallel approach for the pixel intensity based image registration (IR) problem on multi-core processors. While IR is an optimization problem which computes the optimal parameters for a geometric transform, two classes of bio-inspired algorithms are studied: Bacterial Foraging Optimization Algorithm (BFOA) and Genetic Algorithm (GA). The optimal transform is applied to a source image in order to align it to a model image by maximizing a similarity measure. In the presented experiment, mutual information (MI) is used to evaluate the IR quality and most of the processing time is spent in this evaluation. The proposed parallel approach aims to reduce the processing time by using the full computing power of multi-core processors. A comparison of the sequential and parallel versions for different registration problems is presented.**

Keywords—bacterial foraging algorithm, genetic algorithm, image registration, optimization, parallel computing.

I. INTRODUCTION

Image registration is the process of geometric overlaying or alignment of two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors [1]. In a simplified approach, a *source* image has to be aligned to a *model* image. IR is a very important step in image fusion and the registration precision is crucial for pixel level fusion final results. Image registration is frequently used in remote sensing applications, geographic information systems, multispectral image analysis, medical image analysis and other domains. Even the scene is the same, the images may differ by the view angle, subject position but also the capture device may add geometric distortions. To align the input images, IR procedures compute a geometric transform that may be linear, rigid, affine or projective, respectively non-rigid or elastic.

There are two different approaches in IR: area (pixel intensity) based methods and feature-based methods [1]. Almost all methods consist of four steps: feature detection, feature matching, transform estimation and image resampling. The feature detection step, in which distinctive and stable features (points, lines, boundaries of regions) are detected, is specific to feature based registration methods. Usually, the features matching and transform estimation steps are combined because the transform estimation is performed while looking for the correspondent features. The images similarity is evaluated using: (a) the normalized correlation, the Fourier representation or Mutual Information in case of area based IR and (b) spatial relations, invariant descriptors, relaxation methods and multi-resolution transforms (pyramids and wavelets) in case of features based IR [1]. In the transform estimation step, IR procedures must: (a) choose the transform type which may be global or local in case of locally deformed images and (b) compute the transform parameters. The most frequently used transforms are the shape preserving mappings (rotation, translation, scaling) and affine transform. In the image resampling step, the computed transform is applied to the image using interpolation methods as: nearest neighbor function, the bilinear and bicubic functions, quadratic splines, cubic and higher-order B-splines [1].

While the estimation of the transform parameters is an optimization problem which is time consuming in case of area based IR, this paper presents a simple method to increase the efficiency of this process using the computing power of multi-core processors. Two different optimization methods are discussed: Bacterial Foraging Optimization Algorithm (BFOA) and Genetic Algorithm (GA). The foraging model is suitable for optimization problems because animals search for nutrients and try to avoid noxious substances in a way that maximize their energy intake per unit time spent foraging [2]. Computational methods can provide decision models for optimal foraging. The BFOA proposed by Passino uses the Escherichia coli bacteria model because it is the most understood microorganism [2], [3], [4]. It is used in image processing to solve optimization problems like edge detection in combination with a probabilistic derivative technique [5] and fuzzy entropy based image segmentation [6]. A modified version of BFOA used for multilevel thresholding segmentation was compared to genetic algorithms and particle swarm optimization algorithm [7]. Image registration BFOA based methods were proposed in [8], [9] and [10].

Genetic Algorithms are search techniques that emulate evolutionary processes to solve optimization problems [11]. Like BFOA, GAs start with a population of individuals (points) in the problem domain and use these points to approximate the optimal solution. The difference is that instead moving in the problem domain, GAs use the recombination of two or more parents to produce offspring

978-1-4799-2373-1/13/$31.00 ©2013 IEEE

[12]. GAs are often used in biomedical or remote sensing image registration [11].

II. Image Registration using Bio-inspired Computing

A. *Optimization using BFOA Algorithm*

The bacterial foraging paradigm [2], [3], [4] is suitable as model for optimization algorithms because animals / bacteria behavior is to search for nutrients and avoid noxious substances to maximize their energy. As in all evolutionary models, individuals with a good strategy to find nutrients are replicated and those having poor foraging strategy are eliminated. Each member of the bacteria colony is characterized by its position in the n-dimensional space which is a possible solution of the optimization problem. The solution is computed as the position in which a bacterium is in the best healthy state or has the lowest cost value. During foraging, the bacteria colony (swarm) proceeds through four foraging steps: chemotaxis, swarming, reproduction and elimination-dispersal. Let's consider a bacteria colony with S individuals; $P(j,k,l)=\{\theta^i(j,k,l), i=1...S\}$ the position of colony members in the j^{th} chemotactic step, k^{th} reproduction step and l^{th} elimination-dispersal step; $J(i,j,k,l)$ the cost of the i^{th} bacterium in position $\theta^i(j,k,l)$.

a. Chemotaxis: E. Coli bacteria have two types of movements: tumble and swim. The chemotactic step is defined as a tumble followed by a tumble or a tumble followed by a run. In the chemotactic step each bacterium changes its position to:

$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\varphi(i)$, where $C(i)$ is the size of the chemotactic step and $\varphi(i)$ is a unit length random generated direction [4]. If the cost computed in the new position is lower than in the previous position then the swim is continued in the same direction as long as the cost is reduced but not more than a maximum number of steps.

b. Swarming: In case the bacteria have the ability to signal to others the existence of a favorable or poisonous environment, they will tend to swarm together in the direction of nutrients. The cell to cell attraction or rejection is modeled by adding to the cost function $J(i,j,k,l)$ computed for a specific bacterium, components computed as function of the status of all the other bacteria in the colony.

c. Reproduction: All bacteria reach the reproduction state after a number of chemotactic steps. The healthy state is computed for all bacteria and it may be expressed as the total quantity of accumulated nutrients or simply by the value of the cost function in the current position. The least healthy bacteria die while and to keep constant the size of the colony, an equal number of healthier bacteria split into two bacteria without mutation.

d. Elimination and Dispersal: After a number of reproduction steps, some bacteria are dispersed into the environment (moved in a random position) with a specified probability, without taking into account their healthy state.

BFOA algorithm starts with a colony of S bacteria placed in randomly generated positions. The evolutionary process goes through N_{ed} elimination/dispersal steps, each of these consists of N_{re} reproduction steps. Each reproduction step consists of N_C chemotactic steps. In each chemotactic step a bacterium may do at most N_S swarming steps if the value of the cost function decreases. The position in which a bacterium has the greatest healthy status is the solution of the optimization problem. In case of image registration, the size of the search space is equal to the number of parameters of the geometric transform and as healthy status is used the value of a measure that evaluates the similarity between the transformed source image and the model image.

B. *Optimization using Genetic Algorithm*

Genetic Algorithms proposed by Holland are iterative processes that use a population of candidate solutions encoded as chromosome strings. The general structure of GAs is: (a) selection of the appropriate encoding method and fitness function, (b) generation of a random initial population and (c) the evolution loop of the algorithm: fitness function evaluations, application of genetic operators and creation of the new generation. After a number a generations, the population is expected to contain chromosomes that approximate the global maximum value of the fitness function. In each generation chromosomes with best fitness values are retained and generate offspring that replace chromosomes with lowest values of the fitness function. Genetic operators used for new generation creation are: selection, crossover and mutation.

In [11] it is proposed an IR procedure using the string encoding of chromosomes. The parameters of the geometric transform are encoded as bit fields in a 32 bit value. In the procedure described bellow, the real encoded is used and each chromosome is characterized by a number of real values equal to the number of geometric transform parameters. Discrete, average and simplex crossover operators are used depending on user defined probabilities (p_d , p_a and p_s).

C. *Image Registration Experiment*

In the experiment described bellow, the Normalized Mutual Information (NMI) is used to evaluate the similarity between the transformed source image and the model image. MI and NMI evaluate the relative independence of two images and do not depend on the specific dynamic range or intensity scaling of the images [1], [10]:

$$MI(A,B)=H(A)+H(B)-H(A,B) \tag{1}$$

$$NMI(A,B)=(H(A)+H(B))/H(A,B), \tag{2}$$

where $H(A), H(B)$ are the image entropies and $H(A,B)$ is the joint entropy of the two images. High values of mutual information indicate high dependence between images. Because the goal of the optimization algorithms is to minimize a cost function, the value of $(-1)*NMI$ will be used to evaluate the quality of a certain solution.

In the cost function evaluation, the geometric transform corresponding to the current solution is applied to the source image and then the NMI value is computed for the model image and the transformed source image. The area based IR implementations are time consuming because each cost evaluation requires a geometric transform to be applied and also image and matrix operations to compute NMI .

D. Experiment

The experiment was made using two different geometric transforms. In the first case (IR1), the source image (Fig. 2.a) was obtained by applying a rotation (angle $\theta = 5°$) against the top-left corner followed by a translation by $t_x = 10$ and $t_y = -10$ pixels on the two axis to the model image (Fig.1, 200×246 pixels, 8 bits/pixel). The transform matrix is

$$T = \begin{bmatrix} cos\,\theta & sin\,\theta & t_x \\ -\,sin\,\theta & cos\,\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}. \qquad (3)$$

The search space of the optimization problem is R^3 .

In the second case (IR2), the source image (Fig.3.a) was obtained by applying a rotation (angle $\theta = 5°$) against the rotation center ($c_x = 10$ and $c_y = -10$) followed by an isotropic scaling ($scale = 1.2$). The transform matrix is:

$$T = \begin{bmatrix} \alpha & \beta & (1-\alpha)\,c_x - \beta\,c_y \\ -\beta & \alpha & \beta\,c_x + (1-\alpha)\,c_y \\ 0 & 0 & 1 \end{bmatrix}, \qquad (4)$$

where $\alpha = scale \cdot cos\,\theta$ and $\beta = scale \cdot sin\,\theta$.

The search space for the optimization problem is R^4.

In Fig. 2.b and 3.b the results are obtained by applying the IR procedure using BFOA and GA respectively, to detect the optimal parameters of the geometric transform.

The BFOA parameters are $S = 40$, $N_c = 20$, $N_S = 10$, $N_{re} = 16$, $N_{ed} = 2$, $P_{ed} = 0.25$, $C_i = 0.001$ in case of IR1. In IR2, the population size is $S = 400$.

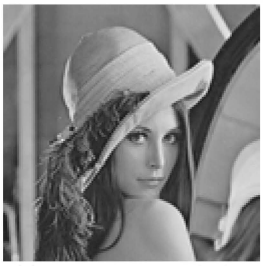
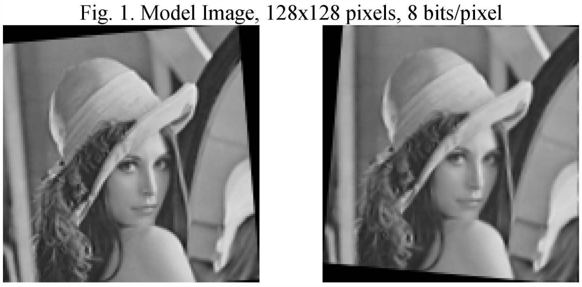
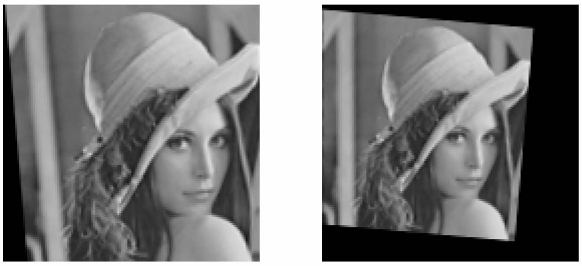


Fig. 1. Model Image, 128x128 pixels, 8 bits/pixel



a. source image b. registered image

Fig. 2. Results obtained in experiment IR1



a. source image b. registered image.

Fig. 3. Results obtained in experiment IR2

The GA parameters are $nGen = 150$, $nCr = 250$ in case of IR1 and $nGen = 500$, $nCr = 1500$ in case of IR2. The crossover probabilities are: $p_d = 0.05$, $p_a = 0.15$ and $p_s = 0.2$.

The results are presented in Table I and II. It must be noted that the expected values for the cost function, computed using inverse transforms are -1.3694 for IR1 and -1.2906 for IR2.

TABLE I
RESULTS OF IR USING SEQUENTIAL BFOA

	Min Cost	# Cost Eval	Time (sec)
IR1	-1.3692	61071	66.3
IR2	-1.2899	568241	606.0

TABLE II
RESULTS OF IR USING SEQUENTIAL GA

	Fitness	# Cost Eval	Time (sec)
IR1	1.3693	5937	6.8
IR2	1.2788	99405	114.5

III. PARALLEL APPROACH OF IMAGE REGISTRATION

A short analysis of the registration tasks made using the performance wizard of Visual Studio 2010 reveals that most of the processing time was spent in the cost function evaluations. In case of BFOA based registration, 99.4% of the execution time was spent in the cost evaluation function and more detailed, 82.9% for mutual information computing and 16.4% to apply the geometric transform to the source image. The situation is similar in case of GA optimization: 96.6% of the execution time is spent in the cost evaluation (81.1% to compute mutual information and 14.4% to apply the geometric transform). It must be noticed also that in both BFOA and GA cases, only about 25% of the computing power is used in case of Core i5 processor.

A closer look at BFOA reveals that it contains 4 nested loops: elimination/dispersal, reproduction and chemotaxis for each bacterium in the colony. The body of the inner loop is executed $N_{ed} \times N_{re} \times N_C \times S$ times, which may be a fairly large number. In fact, the cost function evaluation is performed more than two times this number due to the fact that each bacterium may perform more swim steps in a single chemotactic step. If we consider not using the cell to cell attractant / repellant effect in the optimization algorithm, then the calculations performed for each individual bacterium in the inner loop are independent. So, we can execute in parallel a chemotactic step for all bacteria.

For the GA optimization, the cost function evaluation is called from two different places. First, it is called from the main evolution loop of the algorithm (about 41% of execution time) only for the new created chromosomes evaluation, and second, in the simplex crossover function (about 53% of execution time). In the first case, the cost function is called for all not already evaluated chromosomes, so this task is easily parallelized. In case of simplex crossover that involves more than one chromosome, the crossover function will be executed in parallel for each group of chromosomes.

The parallel implementation of the described fusion procedures was evaluated on two computers: Intel Core i5 3.10 GHz and Core2Duo 2.16 GHz processor. Both systems have 4GB RAM and use Windows 7 64 bits as operating system. The most common evaluation of parallel algorithms is performed using the parallel efficiency $E = t_s / (t_p \times n)$, where t_s is the time used by the sequential version of the algorithm, t_p is the processing time for the parallel version and n is the number of used processors. In Table III the execution time for the sequential and parallel versions of BFOA and GA based IR procedures are presented. The parallel efficiency values are shown in Fig. 4.

The registration quality may be assessed by the following parameters:

1) *accuracy*, as the ratio between correct and all registrations. A registration is correct if the Euclidean distance from the ground truth and final translation is at most 2 pixels, and if the maximum absolute value of the three rotation errors is less than 2. We obtained accuracies between 0.66 and 0.85 for different images.

2) *efficiency*, given by the mean number of function evaluations for a correct registration. This parameter is directly related to the processing time shown in Table III.

Moreover, registered images are less sensitive to *contrast changes* compared to each image alone.

The image registration procedure was implemented and tested in an image processing framework developed by authors. It is implemented in C++ as a Windows application and uses OpenCV library [13] for image manipulation and the parallel programming support available in Microsoft Visual Studio 2010 [14].

IV. CONCLUSIONS

The second section contains a brief description of Bacterial Foraging Optimization Algorithm and Genetic Algorithm. The results obtained using the sequential version of the optimization algorithms are also presented. The third section contains an analysis of the processor usage during the sequential execution of the two algorithms, the proposed parallelization and a comparison of the sequential and parallel versions in terms of the execution time and parallel efficiency. The proposed parallel approach for image registration using BFOA and GA is based on the shared memory parallelization model which may be easily implemented on the common multi-core processors. The parallel implementation may be used for more methods of cost function evaluation while the cost function is a parameter in the algorithm implementation. A next step of our study will be to use specific evolutionary operators, like crossover and mutation, to BFOA, in order to speed up the computing (execution) time, both for sequential and parallel approaches.

TABLE III
EXECUTION TIME FOR THE SEQUENTIAL AND PARALLEL VERSIONS (SEC.)

			Core i5	Core2Duo
IR1	BFOA	Seq	66.25	109.40
		Par	25.79	64.60
	GA	Seq	6.76	11.33
		Par	2.99	6.82
IR2	BFOA	Seq	605.99	992.95
		Par	197.90	553.98
	GA	Seq	114.52	192.98
		Par	40.34	115.50

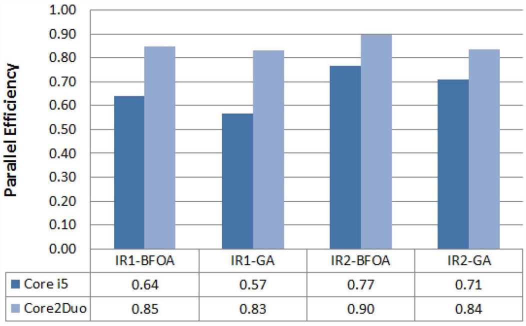


Fig. 4. Parallel efficiency computed for Corei5 and Core2Duo processors

REFERENCES

[1] B. Zitova, J. Flusser, "Image registration methods: a survey", *Image and Vision Computing*, 21, Elsevier, 2003, pp. 977–1000.

[2] Y. Liu, K. M. Passino, "Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors", *Journal of Optimization Theory and Applications*, Vol. 115, No. 3, 2002, pp. 603–628.

[3] K.M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", *IEEE Control Systems Magazine*, June 2002, pp. 52-67.

[4] K.M. Passino, Biomimicry for Optimization, Control, and Automation, Chapter 18: *Cooperative Foraging and Search*, Springer Verlag, 2005.

[5] O. P. Verma, M. Hanmandlu, P. Kumar, S. Chhabra, A. Jindal, "A novel bacterial foraging technique for edge detection", *Pattern Recognition Letters*, 32, Elsevier, 2011, pp. 1187–1196.

[6] N. Sanyal, A. Chatterjee, S. Munshi, "An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation", *Expert Systems with Applications*, 38, Elsevier, 2011, pp. 15489–15498 .

[7] P.D. Sathya, R. Kayalvizhi, "Modified bacterial foraging algorithm based multilevel thresholding for image segmentation", *Engineering Applications of Artificial Intelligence*, 24, Elsevier, 2011, pp. 595–615.

[8] Z. Yudong, W. Lenan, "Multi-resolution rigid image registration using bacterial multiple colony chemotaxis", *5th Int. Conf. on Visual Information Engineering*, 2008, VIE 2008, pp. 528-532.

[9] H. Costin, C. Rotariu, "PET and CT images registration by means of soft computing and information fusion", Proc. of the 1st WSEAS Int. Conf. on Biomed. Electronics and Biomed. Informatics, 2008, pp. 150-161.

[10] H. Costin, S. Bejinariu, "Medical image registration by means of a bio-inspired optimization strategy", *The Computer Science Journal of Moldova*, vol. 20, Nr. 2 (59), 2012, pp. 178-202.

[11] R.Singhai, J.Singhai, "Registration of Satellite Imagery using Genetic Algorithm", *Proc of the World Congress on Engineering*, WCE 2012, London, UK, Vol II.

[12] S.Tsutsui, M.Yamamura, and T.Higuchi, "Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms", *Proc. of the Genetic and Evolutionary Computation Conference, 1, pp. 657--664. Orlando, Florida, USA, January 2000*

[13] G. Bradski, A. Kaehler, *Learning OpenCV. Computer Vision with the OpenCV Library*, O'Reilly Media, Inc., 2008.

[14] C. Campbell, A. Miller, *Parallel programming with Microsoft Visual C++*, Microsoft Corporation, 2012.