GA-based Parallel Image Registration on Parallel Clusters

Prachya Chalermwat[1], Tarek El-Ghazawi[1], and Jacqueline LeMoigne[2]

1 Institute for Computational Sciences and Informatics
George Mason University
prachya@science.gmu.edu, tarek@gmu.edu

2 Code 935
NASA Goddard Space Flight Center
lemoigne@cesdis.gsfc.nasa.gov

Abstract. Genetic Algorithms (GAs) have been known to be robust for search and optimization problems. Image registration can take advantage of the robustness of GAs in finding best transformation between two images, of the same location with slightly different orientation, produced by moving spaceborne remote sensing instruments. In this paper, we have developed sequential and coarse-grained parallel image registration algorithms using GA as an optimization mechanism. In its first phase the algorithm finds a small set of good solutions using low-resolution versions of the images. Based on the results from the first phase, the algorithm uses full resolution image data to refine the final registration results in the second phase. Experimental results are presented and we found that our algorithms yield very accurate registration results and the parallel algorithm scales quite well on the Beowulf parallel cluster.

1 Introduction

Digital image registration is very important in many applications, such as medical imagery, robotics, visual inspection, and remotely sensed data processing. The NASA's Earth Observing System (EOS) project will be producing enormous Earth global change data, reaching hundreds of Gigabytes per day, that are collected from different spacecrafts and different perspectives using many sensors with diverse resolutions and characteristics. The analysis of such data requires integration, therefore, accurate registration of these data. Image registration is defined as the process that determines the most accurate relative orientation between two or more images, acquired at the same or different times by different or identical sensors. Registration can also provide the orientation between a remotely sensed image of the Earth and a map.

Given a reference and an input images, the image registration process determines the amount of rotation and the amount of translation (in both the x-axis and y-axis), the input image has with respect to the reference image, figure 1. Due to the tremendous amount of data generated from the EOS remote sensing satellites and the amount of images to be registered, the time required to register input images to existing reference images is quite large. This prompted the need for parallel processing to conquer the lengthy computation time[1][2][3]. In [3] we discussed the computational savings in image registration resulted from using the Wavelet-based technique which exploits the multiresolution property of Wavelet.

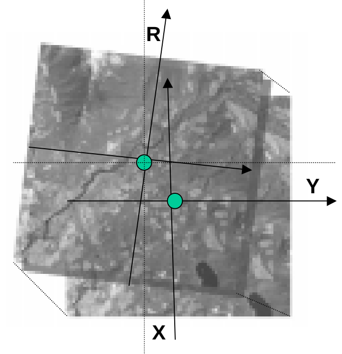


Fig. 1. Image Registration is to find transformations (rotation R, X and Y translation) between input and reference images

Genetic Algorithms (GAs) have been long known to be very robust for search and optimization problems. It is based on the principles of natural biological evolution that operates on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution [4][5]. In this paper, two-phase GA-based image registration algorithms (sequential and parallel) are proposed. In the first phase near optimal transformations of the registration are found using low-resolution version of the images (32x32). In the second phase we employ full resolution images to refine the results from the first phase and obtain the best solution. The transformation, consisting of rotation, translations in X-axis and Y-axis (R, X, Y), is encoded into a bit string of chromosome and is decoded for an evaluation of the fitness values. We will show that our GA-based sequential registration is very computationally efficient and robust for the subset of image test suite. Additionally, we will demonstrate that our GA-based parallel image registration scales and provides fast registration. Our implementation is based on SuGal library[1] that allows us to design our image registration evaluation function. We will discuss this evaluation function in details in section 2. In this paper, the used Beowulf cluster (at NASA GSFC) has 50 200MHz Pentium Pro, and runs Linux kernel 2.0.35. Interprocessor communication is provided by a combination of GigabitEthernet and switched FastEthernet.

This paper is organized as follows. Section 2 describes GA-based sequential image registration. Section 3 discusses parallel approach to GA-based image registration followed by some experimental results in section 4. Conclusions and future research are listed in section 5.

---

[1] A library supports a large number of variants of Genetic Algorithms, and has extensive features to support customization and extension. Sugal can be freely downloaded at http://www.trajan-software.demon.co.uk/sugal.htm

2 GA-based Image Registration

2.1 Genetic Algorithms (GAs): Concept and Definitions

Genetic algorithm (GA) was formally introduced by John Holland and his colleague [4]. It is based on the natural concept that diversity helps to ensure a population's survival under changing environmental conditions. GAs are simple and robust methods for optimization and search and have intrinsic parallelism. GAs are iterative procedures that maintain a population of candidate solutions encoded in form of chromosome string. The initial population can be selected heuristically or randomly. For each generation, each candidate is evaluated and is assigned the fitness value that is generally a function of the decoded bits contained in each candidate's chromosome. These candidates will be selected for the reproduction in the next generation based on their fitness values. The selected candidates are combined using the genetic recombination operation "crossover". The crossover operator exchanges portions of bit string hopefully to produce better candidates with higher fitness for the next generation. The "mutation" is then applied to perturb the string of chromosome as to guarantee that the probability of searching a particular subspace of the problem space is never zero [5]. It also prevents the algorithm from becoming trapped on local optima [8][9]. Then, the whole population is evaluated again in the next generation and the process continues until it reaches the termination criteria. The termination criteria may be triggered by finding an acceptable approximate solution, reaching a specific number of generations, or until the solution converges. Table 1 shows some basic GAs terminology used in this paper.

Table 1. Some important GAs definitions

Chromosome	Individual encoded strings of bits, integer, etc. that are uniquely mapped onto the decision domain
Population	Number of individuals to be evaluated for each generation
Fitness value	Performance of each individual derived from the objective function
Crossover	An operator applied to pairs of individuals to exchange their characteristics
Selection	A process of determining which individual will be chosen for reproduction and the number of offspring that an individual will produce
Mutation	An operator used to alter bits in the chromosomes to possibly avoid the local optima convergence

2.2 GA-based Image Registration

GA-based image registrations for medical images have been reported in [10][11][12]. The first two targeted finding transformation vector between two medical images, while in [12] Turton applied the GAs to real-time computer vision. The transformation vector used is called "bilinear geometric transformation" which is basically a combination of four values indicating the distances between the four corners of the two images [11]. The work by Ozkan is based on a 5-processor transputer architecture applied to images of 100x100 pixels and using 80-bit chromosome. Turton's work is based on VLSI-level hardware architecture focusing on vision systems that require fast and robust processing. The effort, however, did not address MIMD parallel GAs or multiresolution image registration. These studies have also provided no quantitative insight into the accuracy of the registration results.

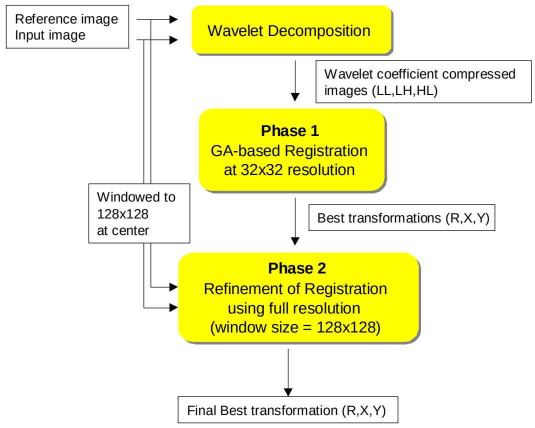


Fig. 2. Flow diagram of GA-based image registration using wavelets

In this paper, a two-phase approach is proposed. In the first phase, called GA32, we apply GAs technique to the 32x32-pixel compressed version of the image, there the image is compressed using the wavelet technique [13]. As shown in figure 2, using the results from GA32, we refine the registration using full resolution images in the second phase. At full resolution of 512x512, registration is computationally very expensive. We found, however, that 128x128 of full resolution image, the registration requires much less computations but still maintains the accuracy of the registration [14]. While the rotation is not sensitive to the image size the translation in both x-axis and y-axis must be adjusted to the corresponding original resolution. In this second phase, called FULL phase, we only use a 128x128 window at the center of the original reference and input images for registration. In this work, compression using the wavelet decomposition is done off-line. Our work in [13] has shown promising scalability of wavelet decomposition on various parallel platforms and we believe that incorporating parallel wavelet decomposition to the current implementation of GA-based parallel image registration in the future will not degrade the overall performance.

2.3 Accuracy of GA-based Image Registration

The accuracy of our 2-phase GA-based image registration is shown in table 2. The table shows that the GA-based image registration at 32x32 pixels is very accurate despite the fact that it has been compressed by a factor of 256. The final results from the two-phase algorithm is very accurate and robust, and have found the solution in thematic mapper and digitized photo images with an average error of 0.3 degrees, 1.7 pixels, and 0.4 pixels for rotation, translation x, translation y respectively. Note also that in the second phase, it is sufficient to use a 128x128 subimage to refine the results. This subimage is selected from the center of the full image, where image sensor distortion is minimal.

Table 2. Accuracy using full resolution to refine the registration (with 128x128 centered window) and FP FACTOR = 200

				32x32						512x5152					
	ORIGINAL			REGISTERED			ERROR			REGISTERED			ERROR		
	R	X	Y	R	X	Y	R-ERR	X-ERR	Y-ERR	R	X	Y	R-ERR	X-ERR	Y-ERR
tm.ref	0	0	0	-0.2	3.2	3.2	*0.2*	*-3.2*	*-3.2*	0	0	0	*0*	*0*	*0*
tm.r4	4	0	0	5.17	-1.12	-0.64	*-1.17*	*1.12*	*0.64*	4	0	0	*0*	*0*	*0*
tm.r4tx50	4	50	0	3.43	46.08	1.92	*0.57*	*3.92*	*-1.92*	4	50	0	*0*	*0*	*0*
tm.r4tx5ty2	4	5	2	4.2	3.52	1.92	*-0.2*	*1.48*	*0.08*	4	5	2	*0*	*0*	*0*
tm.tx50	0	50	0	0.58	43.84	-4.48	*-0.58*	*6.16*	*4.48*	0	50	0	*0*	*0*	*0*
tm.tx5ty2	0	5	2	0.1	5.28	2.56	*-0.1*	*-0.28*	*-0.56*	0	5	2	*0*	*0*	*0*
	R	**X**	**Y**	**R**	**X**	**Y**	**R-ERR**	**X-ERR**	**Y-ERR**	**R**	**X**	**Y**	**R-ERR**	**X-ERR**	**Y-ERR**
girl.ref	0	0	0	0.33	-2.4	3.2	*-0.33*	*2.4*	*-3.2*	0	0	0	*0*	*0*	*0*
girl.r5	5	0	0	5.21	0.8	1.92	*-0.21*	*-0.8*	*-1.92*	5	0	0	*0*	*0*	*0*
girl.r5tx20ty60	5	20	60	4.65	22.88	59.2	*0.35*	*-2.88*	*0.8*	5	19	56	*0*	*1*	*4*
girl.r5tx6ty4	5	6	4	5.8	2.4	4.8	*-0.8*	*3.6*	*-0.8*	5	6	4	*0*	*0*	*0*
girl.tx20ty60	0	20	60	1.1	12.8	49.6	*-1.1*	*7.2*	*10.4*	0	19	57	*0*	*1*	*3*
Average							**-0.31**	**1.70**	**0.44**				**0**	**0.182**	**0.636**

3 GA-based Parallel Image Registration

3.1 Mapping Image Registration to GAs

3.1.1 Chromosome Encoding

Using a bit encoding scheme for chromosome string, our transformation R, X, and Y is encoded as shown in figure 3. A 12-bit field is used to represent possible relative rotation of the input image to the reference image. Likewise, 10 bits are used to express translation in x-axis and 10 more for the y-axis. All representations are signed magnitude, using one bit for the sign and the rest of the bits to represent the magnitude of the rotation or translation. Thus, relative rotation has the range of ±2048 degrees, while relative translation in the x (or y) direction has the range of ±512 pixels. As shown in table 3, to increase the precision of the search into sub-degrees using floating-point numbers, we divide these numbers by FP_FACTOR. For example if FP_FACTOR is equal to 400 then we have the search scope within ± 5.2, ±1.28, and ± 1.28 for R, X, and Y respectively. Note the translation at 32x32 pixels is compressed by a factor of 16. This means the translation X and Y will be equivalent to $1.28 \times 2^4 = \pm 20.5$ pixels at full resolution (512x512).

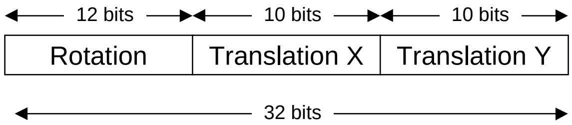


Fig. 3. Encoding of transformation R, X, and Y into 32-bit chromosome string

3.1.2 Objective Function and Other GA Operations

Our objective function is to maximize the correlation between the reference and input images. This means the fitness function is simply the correlation function of the input (transformed) image and the reference image. This works quite well regardless of the simplicity of the GAs operation. The correlation function between two images A and B is shown in equation (1).

$$C(A,B)=\frac{\sum_i (A_i-\bar{A})\times(B_i-\bar{B})}{\sqrt{\sum_i (A_i-\bar{A})^2}\times\sqrt{\sum_i (B_i-\bar{B})^2}} \qquad (1)$$

For selection of individuals, we select highly fit individuals with higher correlation (fitness) value. Thus, the objective function is to maximize the correlation values between the reference image and the input image. The crossover used is a single-point crossover and the termination condition is to stop GA after the solution converges or reaches a pre-specified number of generations.

3.2 Parallel Genetic Algorithms

Chipperfield and Fleming [5] classified parallel GAs into three categories: Global GAs, Migration GAs, and Diffusion GAs.

Global GAs treat the entire populations as a single breeding unit. Near linear speed-up may be obtained using global GAs when the objective function is significantly more computationally expensive than the GA itself (our image registration fits this). This is even true on networks of workstations that demand low communication overhead.

Migration GAs divide the population into a number of sub-populations, each of which is treated as a separate breeding unit under the control of a conventional GA. To encourage the proliferation of good genetic material throughout the whole population, individuals migrate between the sub-populations from time to time. It is also considered coarse-grained. In this category, the population is distributed into sub-population and the local breeding is used (sometimes called *island model*)[5]. The highly fit individuals migrate to other subgroups within specific paths to be mated with other individuals in other subgroups to produce better-fit offspring. The Migration GAs are suitable for MIMD architectures.

Diffusion GAs treat each individual as a separate breeding unit, the individuals it might mate with being selected from within a small local neighborhood. The use of local selection and reproduction rules leads to a continuous diffusion of individuals over the population. The Diffusion GAs are considered fine-grained GAs. It is different from the Migration GAs in that it migrates only between local neighborhood.

Local selection results in performance superior to global and migration GAs. It is suitable for SIMD machines but can also be implemented efficiently on MIMD systems.

We found, from the timing profile of our sequential GA-based image registration, that the time to compute the objective function dominates the overall execution time (97.8%). The fraction of other GA-related functions is very small (2.2%). Therefore, our parallel GA-based image registration can take advantage of the simplicity of the global GAs approach while providing good scalability. In our implementation, a master node keeps track of all genetic-related processes except for the evaluation process of the chromosomes. The parallelism resides in the evaluation of the fitness of the chromosomes can be exploited.

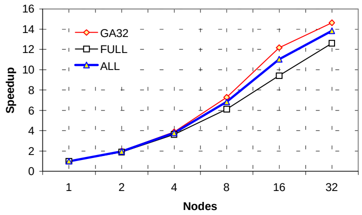
4 Experimental Results

We used subset of images from test suite of NASA and NOAA images [3][1]. The selected images include LandSat/Thematic Mapper (TM) images and digitized photo images (GIRL). Our implementation is based on SUGAL (V-2.1)—SUnderland Genetic Algorithm package. It has been extended for sequential and parallel image registration using MPI. We used two pool sizes (32 and 100) as to observe the effects of increasing computation to the performance of GA-based image registration. We ran our code on a Beowulf parallel computer (ecgtheow.gsfc.nasa.gov), a 50-node Pentium Pro 200 MHz cluster with a fat-tree network topology[2] [6][7].

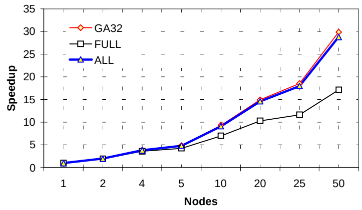
Figure 4 shows experimental results of GA-based parallel image registration with final refinement on a Beowulf parallel computer. The legend GA32 represents the first phase of our parallel algorithm that uses wavelet compressed images of size 32x32 pixels. The legend FULL denotes the second phase that uses the windowed image (128x128) of the full resolution image to refine the registration results obtained from the GA32. The legend ALL represents the measurements of the whole algorithm (GA32+FULL). This figure shows scalability, communication overhead, and computation time for pool size of 32 and 100, on 32 and 50 nodes respectively.

Note that as the number of nodes increases, the communication overhead of the GA32 phase increases, figure 4 c and d. However, the communication overhead of the FULL phase only slightly increases with the number of processors. This is due to the fact that the FULL phase has much fewer communication transactions after the GA32 has narrowed the solution search space dramatically. Figure 4 f shows the pool size effect on the ratio between the computation of GA32 and the FULL phases. In figure

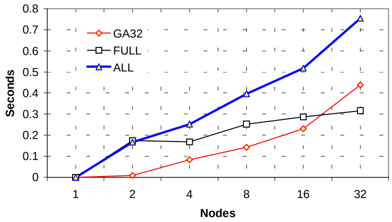
[2] See http://beowulf.gsfc.nasa.gov for more details.



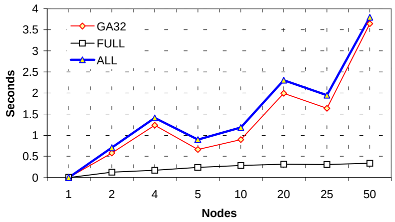
(a) Scalability (pool size = 32)



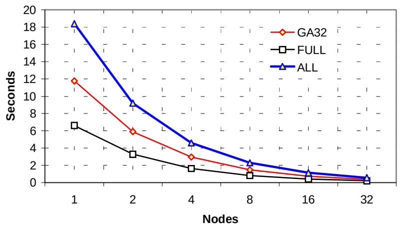
(b) Scalability (pool size = 100)



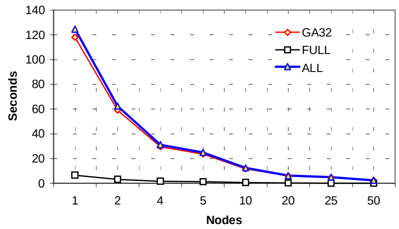
(c) Communication overhead (pool size = 32)



(d) Communication overhead (pool size = 100)



(e) Computation time only (pool size = 32)



(f) Computation time only (pool size = 100)

Fig. 4. Experimental results of the 2-phase GA-based parallel image registration on Beowulf parallel computer: Scalability (a and b), Communication overhead (c and d), Computation only (e and f).

4 f, the computation time of the FULL phase is almost identical to the computation in figure 4 e for the pool size of 32. This is because the FULL case does not use GA's and is independent to the GA operations and population size. Figure 4 c and b show also (for the same reason) that communication increases for the GA phase only by increasing the pool size.

5 Conclusions and Future Works

Despite the low resolution of the compressed wavelet images, the GA-based image has a high degree of accuracy. Additionally, using the full-resolution registration on windowed images at the center in the second phase, the algorithm leads to perfect registration in case of TM images and minor errors with the digitized image. Our parallel approach scales with the number of processors almost with linear speedup. High efficiency has been obtained for both pool size = 32 and pool size = 100. For pool size = 32, the algorithm maintain an average of 90% efficiency up to 8 nodes. At 16 nodes, efficiency is about 70% before dropping to 43% at 32 nodes, due to increasing communication overhead.

The current algorithm expects the pool size to be divisible by number of processors used to evenly distribute the load among all processors. Future work will incorporate dynamic load balancing to eliminate the dependency between the pool size and number of processors and will test against a larger set of images.

Acknowledgment The authors would like to thank Donald Becker, Erik Hendrik, and Phil Merky at CESDIS GSFC for their help with the Ecgtheow Beowulf cluster at CESDIS, NASA Goddard Space Flight Center.

References

1. J. Le Moigne. “Towards a Parallel Registration of Multiple Resolution Remote Sensing Data”, Proceedings of IGARSS’95, Firenze, Italy, July 10-14, 1995.
2. M. Corvi and G. Nicchiotti, “Multiresolution Image Registration,” in Proceedings 1995 IEEE International Conference on Image Processing, Washington, D.C., Oct. 23-26, 1995.
3. T. El-Ghazawi, P. Chalermwat, and J. LeMoigne, “Wavelet-based image Registration on parallel computers," in SC'97: High Performance Networking and Computing: Proceedings of the 1997 ACM/IEEE SC97 Conference: November 1521, 1997.
4. J. H. Holland, “Adaptation in Natural and Artificial System,” University of Michigan Press, Ann Arbor, 1975.
5. A. Chipperfield and P. Fleming, “Parallel Genetic Algorithms,” in *Parallel & Distributed Computing Handbook* by A. Y. H. Zomaya, McGraw-Hill, 1996, pp. 1118-1143.
6. Mike Berry and Tarek El-Ghazawi. "Parallel Input/Output Characteristics of NASA Science Applications". Proceedings of the International Parallel Processing Symposium (IPPS'96), IEEE Computer Society Press. Honolulu, April 1996.
7. D. Ridge, D. Becker, P. Merkey, T. Sterling, “Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs,” Proceedings, IEEE Aerospace, 1997.
8. P. Husbands, “Genetic Algorithms in Optimisation and Adaptation,” in *Advances in Parallel Algorithms* Kronsjo and Shumsheruddin ed., 1990, pp. 227-276.
9. D. E. Goldburg, Genetic Algorithms in Search: optimization and machine learning, Reading, Mass. Addison-Wesley, 1989.
10. J. M. Fitzpatrick, J. J. Grefenstette, and D. Van-Gucht. "Image registration by genetic search," Proceedings of Southeastcon 84, pp. 460-464, 1984.
11. M. Ozkan, J. M. Fitzpatrick, and K. Kawamura, “Image Registration for a Transputer-Based Distributed System,” in proceedings of the 2nd International Conference on Industrial & Engineering Applications of AI & Expert Systems (IEA/AIE-89), June 6-9, 1989, pp. 908-915.
12. B. Turton, T. Arslan, and D. Horrocks, “A hardware architecture for a parallel genetic algorithm for image registration,” in Proceedings of IEE Colloquium on Genetic Algorithms in Image Processing and Vision, pp. 111-116, Oct. 1994.
13. T. El-Ghazawi and J. L. Moigne, "Wavelet decomposition on high-performance computing systems," in Proceedings of the 1996 International Conference on Parallel Processing, vol. II, pp. 19-23, May 1996.
14. P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, “Image registration by parts," in Image Registration Workshop (IRW97), pp. 299-306, NASA Goddard Space Flight Center, MD, Nov. 1997.