Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

7th International Conference
on Electrical Engineering
ICEENG 2010

Robust registration of 3D point clouds using GA with adaptable boundary constraints

By

Mariano A. Imbert R.* Li Xiaoxing*

Abstract:

Registration of 3D point clouds is an important task for many different applications today, like reverse engineering, medical imaging, remote sensing, robotics and automation in general. In this paper we explore a new algorithm for rough registration by using the genetic algorithm (GA) which has showed acceptable results in a reasonable amount of computational time. The algorithm is based on the Median Squared Error (MSE) as the fitness function for the GA, and a threshold that defines the maximum distance between corresponding pair of points to be considered inliers. One of the contributions in this paper is a new scheme for adaptable boundary constrains in the algorithm, which makes the global minimum detection faster than the traditional GAs implementations. Another contribution is the separation of the chromosome parameter, i.e. for the first steps of the algorithm we have separated the chromosome encoding into two groups; three genes representing the translation vector (*Tx*, *Ty* and *Tz*) and three genes representing the rotation matrices (*Rx*, *Ry* and *Rz*). Finally, experimental results are presented and discussed using this algorithm.

Keywords:

Registration, point clouds, reverse engineering.

* Beijing University of Aeronautics and Astronautics, Beijing, China

1. Introduction:

For reverse engineering in general, the most common 3D scanning process consists on taking point clouds from several viewpoints without knowing the exact positions of the scanner, so finally several data sets are produced but not aligned in a common coordinate system, the next step should be alignment of the point clouds, and then to generate 3D information, like surfaces or solids, which can be utilized later in a CAD/CAM/CAE system, for simulation, redesign, inspection or manufacturing. Registration is the process of matching different point clouds or range images that are misaligned, i.e. finding the rigid transformations to match a point cloud *a* with a point cloud *b*. This process is mostly thought as having two main stages; rough or crude registration and fine registration. Fine registration is today accomplished using ICP variations [1-14]. The challenge in crude registration lies in performing it automatically and consistently even when there is small overlap [15].

There are many authors that describe what is the basic principle of the GA [16-18], in this paper we first give a background of the implementations of GAs for the registration problem that were found in a recent literature search, and then we introduce a new procedure implementing GAs for finding a rough registration and an adaptable boundary constraints scheme (reducing each time the search space) in the process and separating the chromosome encoding for the first part of the algorithm; 3 parameters at a time are implemented, the rotation parameters and translation parameters. Our experiments have shown a very good convergence and reasonable computing time for the data sets tried in this research, which will be showed in section 5.

2. Related work:

GAs are not new in the registration of 3D shapes, there are many works that have implemented this algorithm for pairwise and multi-view alignment, in this section we describe what we consider the most important advances in this regard. As mentioned earlier, GAs have shown good results in terms of accuracy and robustness for the registration problem. Perhaps one of the most thorough works in this field is the one presented in [20] and [21], that used the MSE (Median Square Error) function but also introduced the SIM (Surface Interpenetration Measure), which is a metric aimed to quantify the interpenetration between two different range images. Furthermore, an early study in the implementation of GA for solving the registration problem can be found in [22], where it is investigated the performance of such an approach for finding the initial guess of the transformation (translation and rotation), between the object and the model surface and then refine the registration using a local gradient descent method such as ICP. Another study was proposed in [23], that introduced the use of the grid closest point technique (GCT), that basically consists of superimposing a 3D fine grid on the 3D space such that the two surfaces *a* and *b* lie inside the grid. Once the marching process is complete, they used the genetic search algorithm to minimize the registration objective function.

In the 3D medical image registration, there are some contributions as well, the work presented in [24] focused particularly on the problem of extraction of the optimal solution among the final genetic population. Another study in this direction was proposed in [25], which was mainly focused in reverse engineering cases, they introduced a robust system for free-form surface registration based on GA and oversampling the point clouds, which has been tested using several data sets. A hybrid approach for automatic registration applied to free-form shapes was proposed in [26], which employs an asynchronous data communication between an Artificial Neural Network (ANN) and GA. Another interesting approach was introduced in [27], that formulated a robust system for free-form surface registration, based on a GA. Due to the characteristics of GA, the developed system does not depend on a good initial guess of solution. As most of the GA approaches, it doesn't need prior information on correspondences or feature points, and it is insensitive to noise and data occlusion. A fully automatic, robust and precise alignment of measured 3D surfaces for arbitrary orientations was introduced in [28], which combines the precision and robustness of TrICP [13], [29] with the generality of GAs. Another study was introduced in [30], here the point matching problem was addressed employing a method based on nearest-neighbor.

3. GA's for 3D data registration:

As we've seen in the preceding section many research works have applied the GA to solve the rough or crude registration problem, and still today there is research in development to improve this achievements mostly to find a better computing time improvements without sacrificing the accuracy. The only drawback of evolutionary methods, like GAs is the computing time, which is generally longer than other methods, but it has been shown very accurate and robust when implemented in the registration problem, as will be discussed later. An advantage for the GAs on the other hand, is that pre-alignment between views is not necessary to guarantee a good result [20]. Another advantage is the fact that it doesn't depend on specific features or landmarks on the surface, and this aspect is especially important for reverse engineering applications, since today's engineering products, like automobiles, ships, aircrafts and spacecrafts are designed using very complex surfaces that don't necessary have symmetric patterns as previous designs and it is expected that in the future this complex and organic shapes will be used even more in other kinds of daily products.

4. Proposed GA-based algorithm and theoretical analysis:

When working with GAs, it is necessary to define the chromosome encoding for the problem, as stated earlier the goal is to find the transformation that align two 3D data sets (*a* and *b*) in a common coordinate system, this transformation can be defined as:

$$T = ar + t \tag{1}$$

Where,

$$r = (r_X \cdot r_Y \cdot r_Z) \tag{2}$$

$$t = (t_X, t_Y, t_Z) \tag{3}$$

And,

$$r_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{3}$$

$$r_Y = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{4}$$

$$r_Z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Here, *r* represents the rotation matrix around the Euler angles $(\theta_x, \theta_y, \theta_z)$ and *t* the translation vector for a point cloud *a*. So, we can define the chromosome string by 6 genes (three representing the 3 rotational parameters, and the other 3 representing the translational parameters). In order to relate the point cloud *a* with point cloud *b* it is necessary to define the fitness function, which will be based on the MSE (mean squared error) equation:

$$f_1 = \frac{1}{N}\sum_{i=1}^{N} \|b_i - ra_i - t\|^2 \tag{6}$$

Where *N* is the number of points in the image *a*, with $a = \{a_i\}$. This is the fitness function utilized in steps 3 and 5 of our algorithm (see algorithm below)

The fitness function implemented in step 8 of our algorithm is also based on the sum of squared distances between corresponding points of two registered images, but this function uses an inlier/outlier classification based on the MSAC robust estimator [20, 31]. The MSAC is an improved robust estimator based on the RANSAC estimator [20, 32]. The fitness value is obtained by the cost function f_2.

$$f_2 = \frac{1}{N}\sum_{i=1}^{N}\rho(r_i) \tag{7}$$

And the robust residual term ρ is denoted by:

$$\rho(n_i) = \begin{cases} n_i \text{ if } n_i < d \\ d \text{ otherwise} \end{cases} \tag{8}$$

Where *d* is a defined threshold (approximately 10% of z-values of the point clouds) and n_i is defined as the norm:

$$n_i = \|b_i - ra_i - t\|^2 \tag{9}$$

And b_i is the closest point in image *b* to the point a_i, with $b = \{b_i\}$. *r* is the rotation matrix and *t* the translation vector, as given by the chromosome's genes. The threshold defines the maximum distance between a corresponding pair of points to be considered inliers. Figure 1 shows a simplification of what should be the real case, note that the point clouds in this case are represented instead by two curves *a* and *b* in a two dimensional space. The inliers are generally those points in the overlapping area of the aligned views [20].

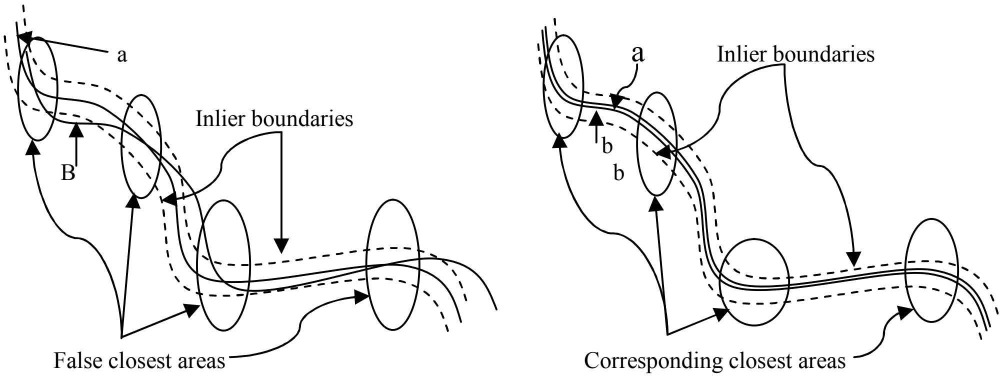

Figure (1): *Threshold for MSAC estimator*

Boundary constrains scheme and procedure:

1. Enter two misaligned point clouds (*a* and *b*)
2. Sample original clouds *a* and *b* uniformly (using approximately 2% of the total points)
3. Apply GA with *registration3t* $(pop = 24, gen = 60, Bc1 = [-z, z])$
4. Transform (translate) cloud *a*: $a_1 = a + t_1$
5. Apply GA with *registration3r* $(pop = 24,\ gen = 60,\ Bc2 = [-180^\circ, 180^\circ])$

6. Transform (rotate) cloud a_1: $a_2 = r_1 a_1$
7. Apply GA *registration6d* $(pop = 80,\ gen = 100,\ Bc3 = t \in [-Bc1/2\ ,Bc1/2]\ r \in\ [-23°\ ,23°])$
8. Transform cloud a_2: $a_3 = r_2 a_1 + t_2$
9. Sample original clouds *a* and *b* uniformly (using approximately 10% of the total points)
10. Apply GA *registration6d* $(pop = 80,\ gen = 100,\ Bc4 = t \in [-0.1 * Bc3\ ,0.1 * Bc3,]\ r \in\ [-5°\ ,5°])$

Because it is a very rough registration in the first steps of the algorithm point clouds *a* and *b* don't need to be too dense, for this reason we can apply a very strong sampling, so just 1% of the point clouds is actually used in steps 3, 5 and 7 (this of course will depend on how dense the point clouds are). Then for step 10, because it is a semi-fine registration process we used denser data sets (10 % of the data is actually used). In step 3, *registration3t* means that we apply the GA using only 3 genes representing the translation vector (*Tx*, *Ty* and *Tz*) and similarly in step 5, three genes representing the rotation matrices (*Rx*, *Ry* and *Rz*). For both step 7 and 10 we have used the MSAC estimator with all the rotation and translation parameters and a threshold of approximately 0.1-1% of z-biggest values in the point clouds for the robust residual term (equation 8).

Boundaries constraints description:

- Bc1 is the biggest *z*-value in clouds *a* or *b*.
- Bc2 is the maximum angles, so the point cloud can rotate 180° with respect to each axis.
- Bc3 is a fraction of Bc1 for the translation parameters and -23° to 23° degrees for the rotational parameters.
- Bc4 is the final boundary constraint which is a fraction of Bc3 for translation and approximately -5° to 5° for rotation.

6. Experimental results:

When implementing GA we consider trying the settings in different configurations, so at the end we can get the best combination for an efficient optimization process. The most important settings in the GA are the population size, chromosome encoding, selection function, mutation function, crossover function and the stopping criteria (mainly the number of maximum generations and function tolerance). In our practice we have used the MATLAB environment, for our scheme phases we have used almost the same parameters, except those explicit in the algorithm above. For the selection function we used stochastic uniform, the crossover function used was scattered and the stopping criteria was established as the maximum generations defined in the algorithm above.

The 3D data sets used for the experimentation in this paper were downloaded from the Stanford 3D scanning repository [33]. We used two partial pairwise data sets, corresponding to the Stanford Bunny (Figure 2) and the Happy Buddha (Figure 3). After having applied our algorithm, we can say that it does the job pretty well for the cases we have studied, however we still think it is necessary to improve and test it with more data sets. Besides the sampling in the algorithm we didn't do any pre-processing to the point clouds.

Experiment 1: Stanford Bunny partial data sets

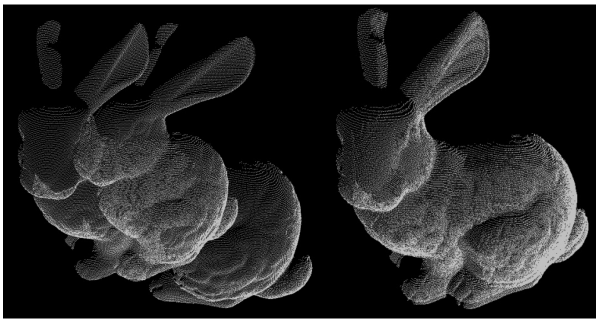

Figure (2): *Stanford bunny point clouds rendering before (a) and after registration (b)*

Experiment 2: Happy Buddha partial data sets

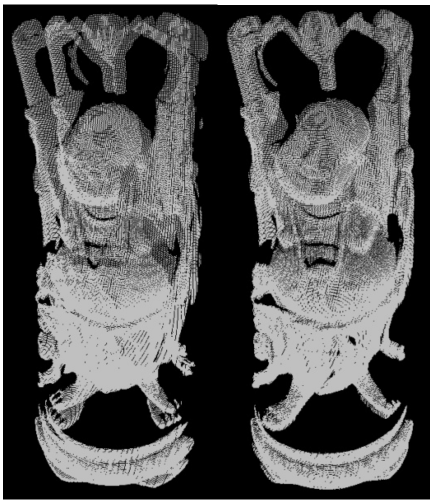

Figure (3): *Happy Buddha point clouds rendering before (a) and after registration (b)*

6. Conclusions:

In this paper a new scheme for boundary constraints was presented, which has showed satisfactory results for rough registration of point clouds. The algorithm presented is GA-based and uses the MSE estimator and the MSAC robust estimator, the latter was difficult to define as a fixed percentage, because for each experiment it was different, i.e. for the bunny we have used 1% of z-biggest values but for the happy Buddha we set it to 0.1%, which means it is greatly problem-dependent. Another contribution in this paper is the separation of the chromosome parameter, i.e. for the initial steps of the algorithm we have separated the chromosome encoding into two groups; three genes representing the translation vector (*Tx*, *Ty* and *Tz*) and three genes representing the rotation matrices (*Rx*, *Ry* and *Rz*). After the experimentation we have found that it does work with different data sets, the sampling scheme was also very useful and did reduce the computation time while avoiding outliers at the same time. Although we haven't tried too many data sets, future work will be focused on testing the algorithm with other data and improve the algorithm's speed and accuracy. Because this is a rough registration algorithm, for fine registration it is possible to implement the ICP or other faster fine registration methods.

References:

[1] P. J. Besl and N. D. McKay, *A method for registration of 3-D shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 14, No. 2, P. 239-256, 1992.

[2] K.-H. Bae and D. D. Lichti, *A method for automated registration of unorganised point clouds* ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 63, No. 1, P. 36-54, 25 July 2007, 2007.

[3] Y. Chen and G. Medioni, *Object modelling by registration of multiple range images*, Image Vision Comput., Vol. 10, No. 3, P. 145-155, 1992.

[4] J. Feldmar and N. Ayache, *Rigid, affine and locally affine registration of free-form surfaces*, International Journal of Computer Vision, Vol. 18, No. 2, P. 99-119, 1996.

[5] E. Trucco, A. Fusiello and V. Roberto, *Robust motion and correspondence of noisy 3-D point sets with missing data*, Pattern Recognition Letters, Vol. 20, No. 9, P. 889-898, 1999.

[6] S. Rusinkiewicz and M. Levoy, *Efficient variants of the ICP algorithm.* P. 145-152.

[7] M. A. Rodrigues and Y. Liu, *On the representation of rigid body transformations for accurate registration of free-form shapes* Robotics and Autonomous Systems, Vol. 39, No. 1, P. 37-52, 2001.

[8] L. Zhu, V. Srivatsan and J. Barhak, *Efficient registration for precision inspection of free-form surfaces*, International journal of advanced manufacturing technology, Vol. 32, No. 5-6, P. 505-515, 2007.

[9] T. Zinsser, J. Schmidt and H. Niemann, *A refined ICP algorithm for robust 3-D correspondence estimation.* P. 695-698.

[10] Y. Liu, L. Li and B. Wei, *3D shape matching using collinearity constraint.* P. 2285-2290.

[11] T. Jost and H. Hugli, *A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images.* P. 427 - 433.

[12] D. Chetverikov, D. Stepanov and P. Krsek, *Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm* Image and Vision Computing, Vol. 23, No. 3, P. 299-309, 1 March 2005, 2005.

[13] D. Chetverikov, D. Svirko, D. Stepanov *et al.*, *The trimmed iterative closest point algorithm.* P. 545- 548.

[14] C. A. Kapoutsis, C. P. Vavoulidis and I. Pitas, *Morphological iterative closest point algorithm*, Image Processing, IEEE Transactions on, Vol. 8, No. 11, P. 1644-1646, 1999.

[15] A. Makadia, A. I. Patterson and K. Daniilidis, "Fully automatic registration of 3D point clouds," in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, 2006.

[16] L. Velho, P. C. Pinto Carvalho, L. H. De Figueiredo *et al.*, *Mathematical Optimization in Computer Graphics and Vision*, P. 300: Morgan Kaufmann, 2008.

[17] P. Venkataraman, *Applied Optimization with MATLAB Programming*, 1st ed.: Wiley Interscience, 2002.

[18] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, 3th ed.: Wiley-Interscience, 2008.

[19] K. F. Man, K. S. Tang and S. Kwong, *Genetic algorithms: concepts and applications [in engineering design]*, Industrial Electronics, IEEE Transactions on, Vol. 43, No. 5, P. 519-534, 1996.

[20] L. Silva, O. R. P. Bellon and K. L. Boyer, *Robust Range Image Registration Using Genetic Algorithms And The Surface Interpenetration Measure*: World Scientific Publishing Company, 2005.

[21] L. Silva, O. R. P. Bellon and K. L. Boyer, *Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 27, No. 5, P. 762-776, 2005.

[22] K. Brunnstrom and A. J. Stoddartt, "Genetic algorithms for free-form surface matching," in Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume IV-Volume 7472 - Volume 7472, 1996.

[23] S. M. Yamany, M. N. Ahmed and A. A. Farag, *A new genetic-based technique for matching 3-D curves and surfaces*, Pattern recognition, Vol. 32, No. 10, P. 1871-1820, 1999.

[24] J. M. Rouet, J. J. Jacq and C. Roux, *Genetic algorithms for a robust 3-D MR-CT registration*, IEEE Transactions on Information Technology in Biomedicine, Vol. 4, No. 2, P. 126 - 136, June 2000, 2000.

[25] G. Percoco and L. M. Galantucci, *Genetic Point Cloud Alignment for Computer Aided Inspection and Reverse Engineering*, 2002.

[26] L. M. Galantuccia, G. Percocoa and R. Spinaa, *An artificial intelligence approach to registration of free-form shapes* CIRP Annals - Manufacturing Technology, Vol. 53, No. 1, P. 139-142 2004.

[27] C. K. Chow, H. T. Tsui and T. Lee, *Surface registration using a dynamic genetic algorithm* Pattern Recognition, Vol. 37, No. 1, P. 105-117, 2003.

[28] E. Lomonosov, D. Chetverikov and A. Ekárt, *Fully automatic, robust and precise alignment of measured 3D surfaces for arbitrary orientations.* P. 39-46.

[29] E. Lomonosov, D. Chetverikov and A. Ekart, *Pre-registration of arbitrarily oriented 3D surfaces using a genetic algorithm*, Pattern Recogn. Lett., Vol. 27, No. 11, P. 1201-1208, 2006.

[30] F. L. Seixas, S. O. L., A. Conci *et al.*, “Image registration using genetic algorithms,” in Proceedings of the 10th annual conference on Genetic and evolutionary computation, Atlanta, GA, USA, 2008.

[31] P. H. S. Torr and A. Zisserman, *MLESAC: a new robust estimator with application to estimating image geometry*, Comput. Vis. Image Underst., Vol. 78, No. 1, P. 138-156, 2000.

[32] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Readings in computer vision: issues, problems, principles, and paradigms*, P. 726-740: Morgan Kaufmann Publishers Inc., 1987.

[33] M. Levoy. "The Stanford 3D Scanning Repository," http://graphics.stanford.edu/data/3Dscanrep/.