Morphological Iterative Closest Point Algorithm

C. P. Vavoulidis and I. Pitas

Department of Informatics, University of Thessaloniki, Thessaloniki 540 06, GREECE

Abstract. This paper describes a method for accurate and computationally efficient registration of 3−D shapes including curves and surfaces. The method is based on the iterative closest point (ICP) algorithm. The real strength of our algorithm is the use of Morphological Voronoi tessellation method to construct the Voronoi regions around the seed points with respect to a certain distance metric. The tessellated domain can be used to avoid the most computationally expensive step in the ICP, namely to find the closest points. Thus the proposed algorithm is much faster than classical ICP.

1 Introduction

Mathematical morphology [1, 4] can be used to construct the Voronoi diagram. Algorithms from mathematical morphology can be found in [2]. Performance analysis of morphological Voronoi tessellation and an implementation of Voronoi diagram based on a region growing method can be found in [6]. The classical ICP algorithm is given in [5]. An accelerated version of the classical ICP algorithm is also given in this paper.

Our algorithm proposes a solution to a key registration problem in computer vision: Given a model 3−D shape and and a data 3−D shape, estimate the optimal rotation and translation that registers the model shape and the actual data shape by minimizing the mean square distance between the shapes. The algorithm requires no extracted features, no curve or surface derivatives and no preprocessing of 3−D data.

An important application of this method is to register actual data sensed from a 3−D object with an ideal 3−D model. Another application is multimodal 3−D image registration in medical imaging (e.g. between NMR and CT volumes). The described method is also useful in the shape equivalence problem as well as in estimating the motion between point sets where the correspondences are not known.

2 The Iterative Closest Point Algorithm

2.1 ICP Algorithm Description

In the ICP algorithm, a data shape P is registered to be in best alignment with a model shape X. In our algorithm the data and the model shape must be decomposed first into point sets if they are not already in this form. The number of 3$-$D points in the data shape will be denoted N_p and the number of points in the model shape will be denoted N_x.

The distance metric d between an individual data point $\mathbf{p}$ and a model shape X will be denoted $d(\mathbf{p}, X) = \min_{\mathbf{x} \in X} \| \mathbf{x} - \mathbf{p} \|$. We use the Euclidean distance as the distance metric. Let C be the closest point operator and Y denote the resulting set of closest points: $Y = C(P, X)$. Given the resultant corresponding point set Y, the least squares registration is computed as: $(\mathbf{q}, d) = Q(P, Y)$. The positions of the data shape point set are then updated via $P = \mathbf{q}(P)$.

The ICP algorithm is given below:

1. The data point set P with N_p points $\mathbf{p}_i$ and the model point set X with N_x points $\mathbf{x}_i$ are given.
2. The iteration is initialized by setting $P_0 = P, \mathbf{q}_0 = [1,0,0,0,0,0,0]^t$ and $k = 0$. The registration vectors are defined relative to the initial data set P_0 so that the final registration represents the complete registration. The four steps are applied until convergence within a tolerance τ.
   (a) Compute the closest points: $Y_k = C(P_k, X)$.
   (b) Compute the registration vector $(\mathbf{q}_k, d_k) = Q(P_0, Y_k)$.
   (c) Apply the registration: $P_{k+1} = \mathbf{q}_k(P_0)$.
   (d) Terminate the iteration when the change in mean-square error falls bellow a threshold $\tau > 0$ specifying the desired precision of the registration $(d_k - d_{k+1} < \tau)$.

If a dimensionless threshold is desired we can replace τ with $\tau\sqrt{tr(\Sigma_x)}$, where the square root of the trace of the covariance of the model shape indicates the rough size of the model shape.

If someone compute the closest point set in each step of the algorithm then the computational complexity of this step is $O(N_p N_x)$. If N_t is the number of initial translation states and N_q is the number of initial rotation states then the cost of local matching is $O(N_t N_q N_p N_x)$ and the cost of global matching is $O(N_q N_p N_x)$. In our algorithm we compute the Voronoi diagram by using an Euclidean distance metric and then we use it in order that the closest point sets are specified in every step of the algorithm by simple reference. The Voronoi regions can be computed as a preprocessing step independently of the rest of the algorithm. The major drawback of our method is that it is space consuming because it is necessary to keep the entire volume containing the Voronoi region labels in RAM. Another problem is the integer arithmetic used when we assign a point to a Voronoi region instead of the floating point arithmetic. However, according to our experimental results, the error caused by this approximation is not significant.

The unit quaternion is a four vector $\mathbf{q}_R = [q_0 q_1 q_2 q_3]^t$ where $q_0 \geq 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The 3×3 rotation matrix generated by a unit quaternion is the one below:

$$\mathbf{R}=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2+q_2^2-q_1^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2+q_3^2-q_1^2-q_2^2 \end{bmatrix} \quad . \tag{1}$$

Let $\mathbf{q}_T = [q_4q_5q_6]^t$ be a translation vector. The complete registration state vector $\mathbf{q}$ is denoted $\mathbf{q} = [\mathbf{q}_R \mid \mathbf{q}_T]^t$. Let $P = \{\mathbf{p}_i\}$ be a measured data point to be aligned with a model point set $X = \{\mathbf{x}_i\}$, where $N_x = N_p$ and where each point $\mathbf{p}_i$ corresponds to the point $\mathbf{x}_i$ with the same index. The mean square objective function to be minimized is:

$$f(\mathbf{q}) = \frac{1}{N_p}\sum_{i=1}^{N_p} \| \mathbf{x}_i - \boldsymbol{R}(\mathbf{q}_R)\mathbf{p}_i - \mathbf{q}_T \|^2 \quad . \tag{2}$$

The center of mass $\boldsymbol{\mu}_p$ of the measured point set P and the center of mass $\boldsymbol{\mu}_x$ of the model point set X are given by:

$$\boldsymbol{\mu}_p = \sum_{i=1}^{N_p} \mathbf{p}_i \quad \text{and} \quad \boldsymbol{\mu}_x = \sum_{i=1}^{N_x} \mathbf{x}_i \quad . \tag{3}$$

The cross-covariance matrix Σ_{px} of the sets P and X is given by:

$$\Sigma_{px} = \frac{1}{N_p}\sum_{i=1}^{N_p}[(\mathbf{p}_i - \boldsymbol{\mu}_p)(\mathbf{x}_i - \boldsymbol{\mu}_x)^t] = \frac{1}{N_p}\sum_{i=1}^{N_p}[\mathbf{p}_i\mathbf{x}_i^t] - \boldsymbol{\mu}_p\boldsymbol{\mu}_x^t \quad . \tag{4}$$

The cyclic components of the anti-symmetric matrix $A_{ij} = (\Sigma_{px} - \Sigma_{px}^T)_{ij}$ are used to form the column vector $\Delta = [A_{23}A_{31}A_{12}]^T$. This vector is then used to form the symmetric 4×4 matrix $Q(\Sigma_{px})$:

$$Q(\Sigma_{px}) = \begin{bmatrix} tr(\Sigma_{px} & \Delta^T \\ \Delta & \Sigma_{px} + \Sigma_{px}^T - tr(\Sigma_{px})I_3 \end{bmatrix} \tag{5}$$

where I_3 is the 3×3 identity matrix. The unit eigenvector $\mathbf{q}_R = [q_0q_1q_2q_3]^t$ corresponding to the maximum eigenvalue of the matrix $Q(\Sigma_{px})$ is selected as the optimal rotation. The optimal translation vector is given by $\mathbf{q}_T = \boldsymbol{\mu}_x - \mathbf{R}(\mathbf{q}_R)\boldsymbol{\mu}_p$. The least squares quaternion operation is denoted as $(\mathbf{q}, d_{ms}) = Q(P, Y)$ where d_{ms} is the mean square point matching error. The notation $\mathbf{q}(P)$ is used to denote the point set P after transformation by the registration vector $\mathbf{q}$.

2.2 The Set of Initial Registrations

The converge of the algorithm in a local minimum depends on the initial registration vector. The only way to be sure that we reach the global minimum is to find all the local minima and to select the smallest one.

Consider a $6-$D state space Ω where the quaternion component q_0 is determined from the other quaternion components as $q_0 = \sqrt{1 - (q_1^2 + q_2^2 + q_3^2)}$. Thus, the state space Ω is a subset of the space $\Omega' = [-1, 1]^3 \times \mathrm{I\!R}^3$, where $\mathrm{I\!R}$ is the real line. If one have fixed the data and model shapes then there is a finite number of local minima $N_m(X, P)$. Let $\Psi(X, P)$ be the set of all local minima: $\Psi(X, P) = \{\psi_n\}_{n=1}^{N_m}$. This induces a natural partitioning of Ω into equivalence classes Ψ_n, where every value of $\mathbf{q}$ that converges to ψ_n is a member of the class Ψ_n. More typically it is stated as:

$$\Omega = \bigcup_{n=1}^{N_m} \Psi_n \quad \text{and} \quad \Psi_n \cap \Psi_m = \emptyset \quad \text{if} \quad n \neq m \ . \tag{6}$$

To guarantee that the global minimum will be reached, we must use an appropriate set of initial registration states so that transforming P by at least one initial registration will place the point set into the equivalence class of registrations that leads to global minimum.

From an implementation point of view, we can use all the normalized combinations of $q_0 = \{1, 0\}$, $q_1 = \{+1, 0, -1\}$, $q_2 = \{+1, 0, -1\}$ and $q_3 = \{+1, 0, -1\}$. For more complicated shapes all the normalized combinations of $q_0 = \{1, 0.5, 0\}$, $q_1 = \{+1, 0.5, 0, -0.5, -1\}$, $q_2 = \{+1, 0.5, 0, -0.5, -1\}$ and $q_3 = \{+1, 0.5, 0, -0.5, -1\}$ provides an adequate set of 312 initial rotation states. If we want to achieve local matching, then we have to combine all the initial rotation with all the initial translation vectors in order to construct the set of all the initial registration vectors.

3 The Morphological Voronoi Tessellation

3.1 The Region Growing Approach

The Voronoi diagram partitions the Euclidean space $\mathbb{R}^3$ into a number of regions. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ be a set of m points in $\mathbb{R}^3$. These points are called *seeds*. The Voronoi tessellation algorithm assigns a region of the space $\mathbb{R}^3$ to each seed. The points in a certain Voronoi region $V(i)$ have the following property: they are closer to the seed point $\mathbf{x}_i \in X$ than to any other seed point $\mathbf{x}_j \in X, j \neq i$, according to a specified distance metric. The Voronoi diagram of X is the union of Voronoi regions for all $\mathbf{x}_i \in X$. The process of constructing the Voronoi diagram of a set $X \subseteq \mathbb{R}^3$ is called *Voronoi tessellation* of $\mathbb{R}^3$.

Voronoi tessellation is a classical topic in computational geometry. There are several algorithms that based on a divide and conquer strategy and find intersections of half planes. Although the computational complexity of this method is relatively low, $O(N log N)$ where N is the number of seeds, its computation is rather complicated. Furthermore this algorithm poses practical problems in the construction of on the Voronoi diagram on the Euclidean grid $\mathbb{Z}^3$.

The approach we present in the following based on a region growing algorithm from mathematical morphology. The construction of the Voronoi diagram in $\mathbb{Z}^3$ could be implemented with a growth mechanism of the points of X. At each step n new points will be appended to each $\mathbf{x}_i \in X$. The set of new points appended to $\mathbf{x}_i$ at each step n is called the $n-$Voronoi neighborhood and denoted $D_n(i)$.

$$D_n(i) = \{\mathbf{x} \in J : d(\mathbf{x}, \mathbf{x}_i) = n, \quad n \in \mathbb{N}, \quad d(\mathbf{x}, \mathbf{x}_i) \leq d(\mathbf{x}, \mathbf{x}_j), \quad j \neq i\} \tag{7}$$

where $J \subseteq \mathbb{Z}^3$, $n \in \mathbb{N}$ is the distance of a point $\mathbf{x} \in J$ from $\mathbf{x}_i$ and $d()$ is a distance function. The n-Voronoi region $X_n(i)$ is defined as:

$$X_n(i) = \bigcup_{k=0}^{n} D_k(i), \quad n \in \mathbb{N} \tag{8}$$

where n is the growth step.

The proposed algorithm consists of an iterative region growing of all the n-Voronoi regions $X_n(i)$ that corresponds to every seed point $\mathbf{x}_i$. When we have collisions, then the growth stops and the collision points are labeled as border. The algorithm stops when all the Voronoi regions cannot grow any more. If we assume that each growth step has the same computational cost, the computational complexity of this method depends only on the number of growth steps executed.

3.2 Implementation of the Morphological Voronoi Tessellation

The input to the algorithm is the seeds points. We choose a structuring element and in each growth step we evaluate the new Voronoi regions. The structuring element corresponds to an approximation of the Euclidean distance in $\mathbb{Z}^3$. This process is repeated until all the points are assigned to a certain Voronoi region or are specified as borders. The dilation mechanism is implemented by scanning all the volume points at each step. A window that represents the structuring element is center on each point. If the center is zero labeled, that is not yet specified, we check the visible points through the window and label this point with a value that depends on the values of the observed points. The center point can be labeled with:

1. a zero value, when all the observed points are zero labeled.
2. A negative value that represents the border value when two or more points labeled with different positive values are observed from the window.
3. A certain positive value when one or more visible points from the window are labeled with the same value.

4 Experimental Results

4.1 3−D Curve Matching

In this section we demonstrate the ability of morphological ICP algorithm in 3−D curve matching. We take 11 points from a curve in a volume, we rotate them 20^o around Z axis and 40^o around the X axis and translate all the points with the translation vector (56,55,52). The initial points and the points after rotation and translation are given in Fig.1.

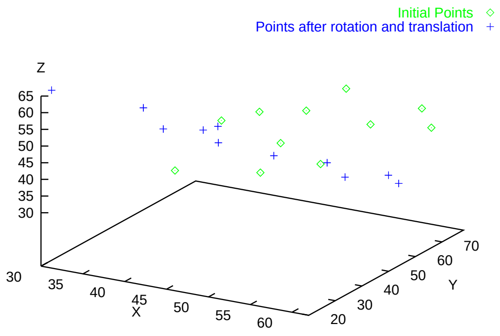

Fig. 1. Model points and Data points after rotation and translation

After the computation of the morphological Voronoi diagram by using the generalization of octagonal distance function in $\mathbb{Z}^3$, it takes only 4 iterations in order the ICP algorithm to converge to the global minimum. The initial points and the points after the registration are given in Fig.2.

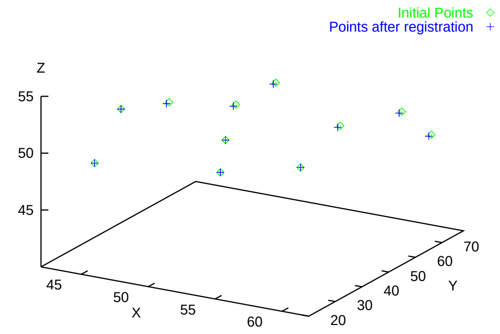

Fig. 2. Model points and Data points after registration

4.2 3−D Surface Matching

In this section we demonstrate the ability of morphological ICP algorithm in surface matching. We take 119 points from the CT frames of a human skull. We rotate the points in a similar way with the curve matching example and translate all the points with the translation vector (81,80,77). The initial points and the points after rotation and translation are given in Fig.3.

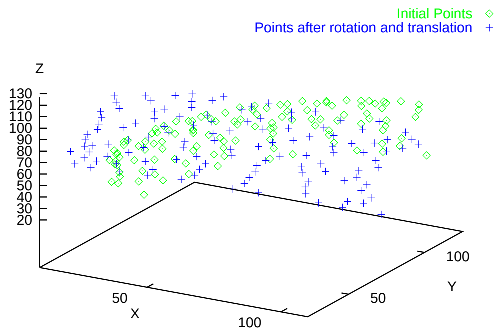

Fig. 3. Model points and Data points after rotation and translation

First we calculate the morphological Voronoi tessellation. It takes only 13 iterations in order the ICP algorithm to lock in the global minimum. The initial points and the points after registration are given in Fig.4.

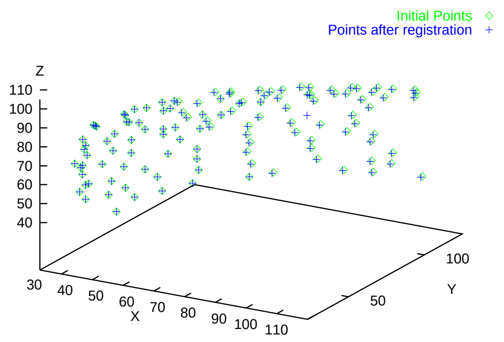

Fig. 4. Model points and Data points after registration

5 Conclusions

The Morphological Iterative Closest Point algorithm is a strong and fast method for the registration of actual data sensed from a 3−D object with an ideal 3−D model. The Voronoi tessellation is used to eliminate the time consuming step of computing the closest point set in each step of the algorithm as the classical ICP algorithm does. Thus, our method is much faster than classical ICP. This method is useful in 3−D registration, registration of medical images, shape equivalence problems and motion estimation between point sets.

References

1. Pitas, I., Venetsanopoulos, A.N.: Nonlinear Digital Filters: Principles and Applications. Boston: Kluwer Academic (1990)
2. Pitas, I.: Digital Image Processing Algorithms. Prentice Hall (1993)
3. Preparata, F.P., Shamos, M.I.: Computational Geometry. New York: Springer-Verlag (1985)
4. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision. Addison-Wesley (1992)
5. Besl, P.J., McKay, N.D.: A method for Registration of 3−D Shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence **14** (1992) 239–256
6. Pitas, I.: Performance Analysis and Parallel Implementation of Voronoi Tessellation Algorithms Based on Mathematical Morphology. IEEE Trans. on Pattern Analysis and Machine Intelligence (Submitted)
7. Kotropoulos, C., Pitas, I., Maglara, A.: Voronoi Tessellation and Delaunay Triangulation Using Euclidean Disk Growing in $\mathbb{Z}^2$. Int. Conf. on Acoust. Speech and Signal Processing (1993)

This article was processed using the LaTeX macro package with LLNCS style