

① 94.17(2) 1-5

# 连续断层图象计算机三维重建 轮廓点匹配插补算法的研究

Study on Computer 3D-Reconstruction Algorithm of the Contour  
Points Matching Interpolation for Serial Cross Images

张双腾  
Zhang Shuangten

张太怡  
Zhang Taiyi

TP391.4

(重庆大学电子信息工程学院, 重庆, 630044)

**A** 摘要 提出了一种连续断层图象计算机三维重建轮廓点匹配插补算法。该算法对连续断层图象相邻轮廓进行轮廓点匹配, 继而插补出中间轮廓, 形成被重建物体的三维形貌。

关键词 三维重建; 插补; 匹配; 叠加; 消隐

中国图书资料分类法分类号 O235

计算机, 断层图象

**ABSTRACT** A computer 3D-reconstruction algorithm was proposed for serial cross images, which is called contour points matching interpolation algorithm. The intermediate contours between two neighbouring contours are produced by contour points matching interpolation to form 3D-shape of the reconstructed object. This algorithm is simple and can be easily implemented. Its advantages are fast speed and strong adaptability for medical serial cross images 3D-reconstruction. Experiments show that the results are satisfactory.

**KEYWORDS** 3D reconstruction; interpolaton; matching; superposition; hidden surface

## 0 引 言

计算机三维重建技术是80年代后期迅速发展起来的一门新兴技术。由于它在各个领域, 尤其是在生物医学领域的巨大的理论及应用价值, 计算机三维重建技术越来越受到各国研究者的重视。

连续断层图象计算机三维重建就是要从一系列平行截面图象数据中恢复被重建对象原有的三维形貌。它的主要步骤是: 首先从各个截面图象中分割出兴趣区的轮廓曲线, 然后由这些轮廓曲线通过算法来构造出其原有的三维形貌。如果各个截面之间的间隔足够小, 那么三维形貌的恢复就可以通过简单地叠加这些轮廓曲线而得。但是各个截面之间的间隔往往不是足够小, 因而需要在每两相邻轮廓之间进行轮廓或表现插补。

\* 收文日期 1993-09-01

现在出现的插补算法有两类。一类是以三角形剖分(Triangulation Technige)为基础的线性插补技术。该类算法的关键是要从巨大数量的三角形中找出最能代表物体表面的三角形<sup>[1~7]</sup>。另一类插补算法是非线性插补算法<sup>[8~12]</sup>,这类算法运用非线性插补技术对物体相邻截面轮廓之间进行轮廓插补,从而形成三维形貌。

两类算法各有优缺点。文献[11]的算法重建效果较好,文献[4]的算法对各种医学断层图象三维重建具有较强通用性,文献[12]的算法重建速度最快。但是两类算法均比较复杂,计算量大,速度慢,不适于微机处理。本文提出的轮廓点匹配插补法,克服了文献中算法的局限性,具有较快的速度和较强的通用性。经实验验证,达到满意效果。

## 1 轮廓点匹配插补算法

### 1.1 插补模型

假设有两轮廓,分别命名为源轮廓和目标轮廓。并假设两轮廓为分别位于两平行平面  $Z = Z_n$  和  $Z = Z_m$  上的可以用一系列点来近似的封闭曲线。每一轮廓曲线可用下式来表示:

$$\bar{C} = \{P_i, 1 \leq i \leq N\}$$

这里  $P_i$  代表轮廓曲线上第  $i$  点,假设

$$\bar{C}_n = \{P_i, 1 \leq i \leq N_n\} \text{ 和 } \bar{C}_m = \{P_j, 1 \leq j \leq N_m\}$$

分别表示一源轮廓和目标轮廓,两轮廓分别位于  $Z = Z_n$  和  $Z = Z_m$  平面上。对源轮廓和目标轮廓进行插补即为在一系列平面  $Z = Z_k (Z_n < Z_k < Z_m, K = n+1, n+2, \dots, m-1)$  上生成轮廓线  $\bar{C}_k$ , 并使  $\bar{C}_k$  由  $\bar{C}_n$  变形逐渐相似于  $\bar{C}_m$ 。

假设所有中间轮廓插补完毕后两相邻轮廓之间的间隔为  $\Delta Z$ , 则有如下插补过程。

1) 按照一定的准则把源轮廓  $\bar{C}_n$  上的点与目标轮廓  $\bar{C}_m$  上的点进行点配对, 可以是 1) 多, 也可以是多对一。

2) 对每一点配对, 无论是一对多或是多对一, 均按照从源轮廓  $\bar{C}_n$  到目标轮廓  $\bar{C}_m$  的方向用直线两两相连。

3) 对于 2) 生成的直线用一平面  $Z = Z_n + \Delta Z$  相切, 所有直线与平面  $Z = Z_n + \Delta Z$  的交点即为插补出的中间轮廓点。由此生成中间轮廓曲线  $\bar{C}_{n+1}$ 。

4) 把插补的中间轮廓曲线作为源轮廓曲线, 目标轮廓曲线不变重复执行 1) ~ 3) 步骤, 直到生成的中间轮廓曲线  $\bar{C}_{n+1}$  与目标轮廓曲线  $\bar{C}_m$  之间的间隔小于或等于  $\Delta Z$ 。

### 1.2 轮廓点配对

假设有一源轮廓曲线  $\bar{C}_2 = \{Q_j, 1 \leq j \leq n_2\}$

及目标轮廓曲线  $\bar{C}_1 = \{P_i, 1 \leq i \leq n_1\}$

如图 1 所示:

假设  $n_2 \geq n_1$  (如果  $n_2 < n_1$ , 那么把目标轮廓曲线作为源轮廓曲线, 而源轮廓曲线作为目标轮廓曲线处理) 在把源轮廓曲线上每一点与目标轮廓曲线上点配对时, 我们规定可一对一, 也可以多对一, 但不能一对多或交叉配对。所谓交叉配对应为  $Q_j$  与  $P_i$  配对, 而  $Q_{j+1}$  与  $P_{i-1}$  配对。即配对必须按照轮廓点下标升序或降序进行。

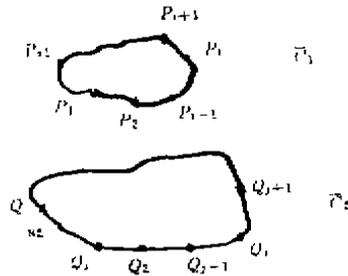


图1 源目标轮廓

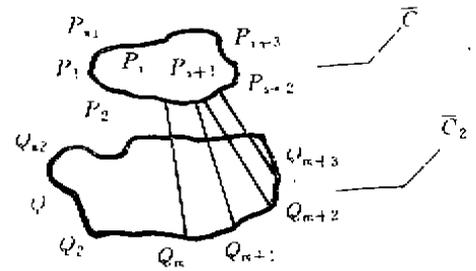


图2 轮廓点配对

如图2所示,假设点 $Q_n$ 为源轮廓上任意一点,则以 $Q_n$ 点为起点,按照轮廓点下标升序方向对源轮廓上每一点与目标轮廓上相应点配对方法如下:

- 1) 寻找 $Q_n$ 点在目标轮廓上的配对点.计算 $Q_n$ 与目标轮廓上每一点的欧氏空间距离 $D_j(1 \leq j \leq n_1)$ .设 $D_j$ 最小值 $\min(D_j) = D_n$ ,那么目标轮廓上 $P_i$ 点即为 $Q_n$ 的配对点.
- 2) 计算目标轮廓与源轮廓点个数之比  $K = n_1/n_2$
- 3) 依次把源轮廓上剩余点 $Q_{n+1}, Q_{n+2}, \dots, Q_{n_2}, Q_{n_2-1}, \dots, Q_1$ 与目标轮廓上点 $P_{i+1}, P_{i+2}, \dots, P_{n_1}, P_{n_1-1}, \dots, P_1$ 配对. $Q_{n+1}$ 与目标轮廓上第 $\text{int}(i+k)$ 个点配对, $Q_{n+2}$ 与目标轮廓上第 $\text{int}(i+2k)$ 即 $P_{\text{int}(i+2k)}$ 点配对,依此类推。 $(\text{int}(i+k)$ 或 $\text{int}(i+2k)$ 表示对 $i+k$ 或 $i+2k$ 取整)

### 1.3 中间轮廓插补

假设有两相邻轮廓曲线:

$$\text{源轮廓曲线 } \bar{C}_n = \{P_i, 1 \leq i \leq n_1\}$$

$$\text{目标轮廓曲线 } \bar{C}_m = \{P_j, 1 \leq j \leq n_2\}$$

其中 $\bar{C}_n$ 位于 $Z = Z_n$ 平面上,而 $\bar{C}_m$ 位于 $Z = Z_m$ 平面上,假设两轮廓已按前一节描述的配对算法进行轮廓点配对,并假设 $\bar{C}_n$ 上一点 $P_i$ 与 $\bar{C}_m$ 上一点 $Q_j$ 配对,如图3所示。

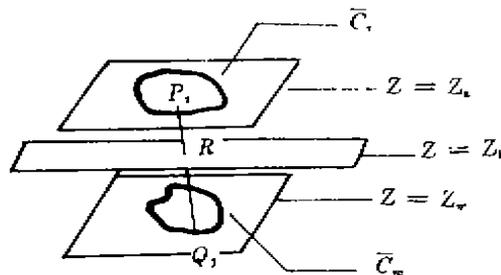


图3 轮廓点配对图

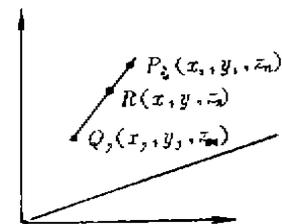


图4 中间轮廓点坐标

假设 $P_i$ 点坐标值为 $(x_i, y_i, z_n)$ , $Q_j$ 点坐标值为 $(x_j, y_j, z_m)$ ,那么位于平面 $Z = Z_k$ 上对应 $P_i$ 及 $Q_j$ 点中间轮廓点 $R$ 的坐标值可以通过下述得出。

如图4所示,有

$$\begin{cases} (z_n - z_k)/(z_n - z_m) = (x_i - x)/(x_i - x_j) \\ (z_n - z_k)/(z_n - z_m) = (y_i - y)/(y_i - y_j) \end{cases} \quad (1)$$

由(1)式有

$$\begin{cases} x = x_i - (x_i - x_j)(z_n - z_k)/(z_n - z_m) \\ y = y_i - (y_i - y_j)(z_n - z_k)/(z_n - z_m) \end{cases} \quad (2)$$

令  $\Delta x = x_1 - x_2$ ,  $\Delta y = y_1 - y_2$ , 及  $\Delta z = z_1 - z_2$ ,  $\Delta z' = z_1 - z_2$ , 则有

$$\begin{cases} x = x_1 - \Delta x \cdot \Delta z' / \Delta z \\ y = y_1 - \Delta y \cdot \Delta z' / \Delta z \end{cases} \quad (3)$$

由式(3)即可以计算出中间轮廓  $\bar{c}_i$  相应点的坐标。

#### 1.4 轮廓分枝的处理

假设有一源轮廓  $\bar{c}$  及  $n$  个目标轮廓  $\bar{c}_i (i = 1, 2, \dots, n)$ , 首先利用前面描述的轮廓配对及插补算法分别对  $(\bar{c}, \bar{c}_1), (\bar{c}, \bar{c}_2), \dots, (\bar{c}, \bar{c}_n)$  进行处理, 由此可得到  $n$  个中间轮廓曲线  $I_i (i = 1, 2, \dots, n)$  则最后的插补结果可以由下式得出  $\bigcup_{i=1}^n I_i$

## 2 三维构形

### 2.1 轮廓曲线叠加与消隐

在三维构形过程中, 把一系列位于平行平面上的原有轮廓曲线及插补出的中间轮廓曲线通过斜平行投影按由前到后的次序叠加。如果只简单地进行叠加, 那么构造出的三维图形中可见及不可见部分均显示在屏幕上。这样的三维图形显得杂乱, 无立体感。因此, 必须进行消隐处理。考虑到轮廓曲线数据特点, 本文采用一种经改进的深度缓冲器消隐法进行消隐。该算法在保持原有算法简单、通用等特点的基础上, 克服了原算法在写屏操作上重复费时等缺点。算法一次性地找出被投影曲面的被遮掩及不被遮掩部分, 通过只投影不被遮掩部分曲面, 从而大大提高了处理速度。算法消隐效果满意。

### 2.2 明暗处理

经叠加及消隐处理后得到的是一幅具有深度的可见面的投影图。为使这幅投影图更加生动、更有立体感, 对投影图进行明暗处理, 使投影图上每个点都赋予一相应的灰度值。光线照射在物体上, 可被吸收、反射或透射, 而只有反射和透射的光使物体可见。光照模型就是物体亮度与光源关系的数学描述。复杂的光照模型可以得到较好的处理效果, 但同时也大大地增加了处理时间的开销。综合考虑各因素, 本文采用简单、快速、实用的点光源光照模型。其计算公式为:

$$I = I_e K_e + \frac{I_i}{r + k} (K_d \cos \theta + K_s \cos^2 \alpha)$$

式中  $I$  为反射光光强,  $I_e$  和  $I_i$  分别为环境光光强及入射光光强,  $K_e$ ,  $K_d$  和  $K_s$  分别为环境光反射系数、漫反射系数及镜面反射系数,  $\theta$  和  $\alpha$  分别为光线入射角及视角,  $r$  为视点与物体表面的距离,  $k$  为一常数,  $n$  一般取值为 1 至 200。

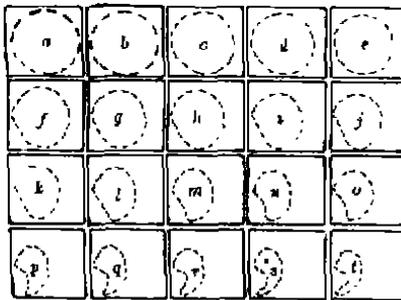


图5 不带分枝轮廓插补图

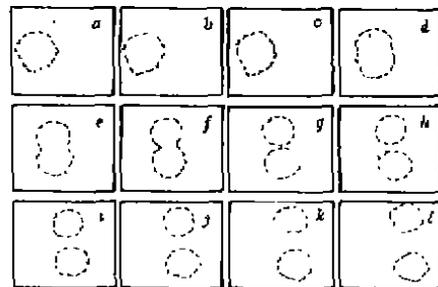


图6 带分枝轮廓插补图

### 3 实验结果与讨论

#### 3.1 实验结果

以本文提出的算法为基础,我们编制了一套完整的三维重建软件对算法进行了验证,取得了较满意的结果。实验结果如图5~7所示。图5及图6分别显不带分枝和带分枝的轮廓插补图,图7为一段带分枝血管重建图。



图7 带分枝血管重建图

#### 3.2 讨论

本文提出的连续断层图象三维重建轮廓点匹配线性插补算法,克服了文献[2]~[12]的局限性,对各种生物学连续断层图象三维重建具有较强的通用性。能处理简单及多分枝物体,并能同时对多个物体进行重建。算法经简单扩充,即能等效地处理带孔物体。经实验验证,该算法还有简单、快速等特点。

### 参 考 文 献

- 1 keppel E. Approximation complex surfaces by triangulation of contour lines. *IBM J. Res. Development*, 1975, 19,2~11
- 2 Fuchs H, Keden Z M, Uselton S P. Optimal surface Reconstruction from Planar Contour. *Commun. ACM.*, 1977,20;693~702
- 3 Wang Y F, Aggarwal J K. Surface Reconstruction and Representation of 3-D Scenes. *Pattern Recognition*, 1986, 19(3);197~207
- 4 Boissonnat J D. Shape Reconstruction from Planar Cross Sections. *Comput. Graphics Image Processing*, 1988, 44;1~29
- 5 Cook L T, Dwyer S J, Batmitsky S, Lee K R. A three-dimensional Display System for Diagnostic Imaging Application. *Comput. Graphics Appl.* 3,1983,(5);13~19
- 6 Christiansen H N, Sederberg T W. Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics. *Comput. Graphics*, 1978,12;187~192
- 7 Ganapathy S, Dennehy T G. A New General Triangulation Method for Planar Contour. *Proc. SIGGRAPHu82*, 1982,16(3);69~75
- 8 Wu S C, Abel J F, Grenberg D P. An Interactive Computer Graphics Approach to Surface Representation. *Commun. ACM*,1977,20;703~712
- 9 Lin W C, Liang C C, Chen C T. Dynamic Elastic Interpolation for 3-D Objects Reconstruction from Serial Cross-Sectional Images. *IEEE Trans. Med. Imaging*, 1988,3;225~232
- 10 Lin W C, Chen S Y, Chen C T. A New Surface Interpolation Technique for Reconstructing 3D Objects from Serial Cross-Sections. *Comput. Graphics Image Processing*, 1989,48;124~143
- 11 Chen S Y, Lin W C, Liang C C, Chen C T. Improvement on Dynamic Elastic Interpolation Technique for Reconstructing 3D Objects from Serial Cross-sections. *IEEE Tran. Med. Imaging*, 1990,9(1);
- 12 Chang L W, Chen H W, Ho J R. Reconstruction for 3D medical Images, A Nonlinear Interpolation Technique for Reconstruction of 3D Medical Images. *Comput. Graphics Image Processing*, 1991,53((4);382~391