

基于傅立叶变换的三维轮廓线快速匹配算法

吕科, 耿国华, 周明全, 赵宏安

(西北大学 计算机科学系, 陕西 西安 710069)

摘要: 给出从提取文物碎片轮廓线出发, 解决空间曲线匹配来达到文物碎片复原的方法。给出了用于查找三维轮廓线匹配的哈希矢量, 以及基于傅立叶变换的轮廓线子段匹配算法, 通过比较两条轮廓线的哈希矢量来分析曲线段的相似性。从理论上给出了判断曲线匹配的性质, 如果曲线段之间的距离越小则哈希矢量之间的距离也越小。经实验证明, 方法计算复杂性低, 预测质量好, 运行高效、稳定、纠错能力强。

关键词: 曲线匹配; 哈希函数; 傅立叶变换

中图分类号: TP391 **文献标识码:** A **文章编号:** 1000-274 X (2003)02-0151-04

在许许多多出土文物碎片中借助计算机辅助碎片复原, 迄今是一个尚未完全解决的问题。本文从提取文物碎片轮廓线出发, 解决空间曲线匹配来达到文物碎片复原的方法。空间曲线的匹配技术是计算机视觉领域内的一项重要研究内容, 也是一个难点问题。针对空间曲线匹配, 曾进行了大量研究, 文献[1]使用半微分不变量的方法对空间进行曲线匹配, 方法主要采用曲线上两点之间的距离以及在每一点的切向量来进行匹配。其不足之处在于不能充分利用曲线的几何特征, 而且两点之间的距离跟曲线所在的坐标系有关。Eyal Kishon 文献[2]采用样条进行三维曲线的匹配, 主要是采用样条来逼近一条曲线, 然后将三维空间曲线转换为一维的字符串进行匹配, 用哈希技术能找到最长的匹配子串。由于该方法是采用样条来逼近曲线, 因此决定了它的适用范围。最小二乘哈希方法^[3], 是在最小二乘哈希方法里, 与一段轮廓线相对应的是一个实数向量, 而不是一个整数。对于这个哈希向量, 应该满足下面的性质^[4]: 如果两个子段的 RMS 距很小, 那么与之相对应的向量之间的距也应该很小。最小二乘法不满足这个性质, 实际上, 最小二乘哈希方法时常会在搜索相似轮廓线时失效, 这是最小二乘哈希方法的一个致命的弱点^[5,6]。

本文所给方法不要求一段轮廓线与另外一段轮

廓线完全匹配, 而是把它们分成很多子段(子轮廓线), 再用这些子段进行匹配。子轮廓线的匹配过程可以看作是个过滤器, 先选定一条给定轮廓线的若干子段, 然后对每一小段, 找到有子段与之相似的所有轮廓线, 进而对找到的轮廓线, 再进行比较。

1 轮廓线描述

图1为通过三维激光扫描仪得到文物碎片曲面, 获取的数据来提取曲面边缘轮廓线。



图1 文物碎片边缘轮廓线提取

Fig.1 3-D object pieces contour

1.1 三维轮廓线的表示

我们认为一条轮廓线由一个点序列组成, 一条空间轮廓线表示为: $P = (p_1, p_2, \dots, p_n)$, p_i 表示第 i 个点与该轮廓线的质心所构成的向量。

1.2 曲线段之间距离

定义两条轮廓线的距离如下:

定义1 用 $P = (p_1, p_2, \dots, p_n)$, $Q = (q_1, q_2,$

收稿日期: 2002-04-12

基金项目: 国家自然科学基金资助项目(60271032); 国家 863 高科技发展计划资助项目(863-502-34-10, 2001AA114182)

作者简介: 吕科(1972-), 男, 宁夏银川人, 西北大学博士生, 从事图形图像处理、可视化技术方面的研究。

..., q_n) 来表示两条空间的轮廓线, 假设 P, Q 经过平移之后, 使其质心 ($\frac{1}{n} \sum_{i=1}^n p_i$) 与原点重合。我们定义 $d(P, Q, R, a)$ 为

$$d(P, Q, R, a) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|Rp_i + a - q_i\|^2},$$

其中 R 为旋转矩阵, a 为平移向量, $\|x\|$ 表示向量 x 的长度。那么 P 和 Q 之间的距离 $d_{rms}(P, Q)$ 就定义为

$$d_{rms}(P, Q) = \min_{R, a} d(P, Q, R, a),$$

$d_{rms}(P, Q, R, a)$ 当 a 为零向量并且 $R = (A^t A)^{1/2} A^{-1}$ 时最小。其中矩阵

$$A = (A_{ij}) (i, j = 1, 1, 2, 3), A_{ij} = \sum_{k=1}^n (p_k)_i (q_k)_j, A^{1/2}$$

表示 $BB = A$ 。很显然, 计算 $d_{rms}(P, Q), R$ 和 a 的时间复杂度为 $O(n)$ 。

1.3 轮廓线的搜索

根据上面对两条曲线距离的定义, 我们这样来定义轮廓线的搜索问题:

输入 一条模板轮廓线 $P = (p_1, p_2, \dots, p_m)$, 一个实数 $\delta > 0$, 一组轮廓线

$$QS = \{Q^1, Q^2, \dots, Q^N\},$$

输出 所有的这些 Q^j , 对每一个 Q^j , 至少包含一个子段, 使得

$$d_{rms}(P, Q_{i,j+m-1}^j) \leq \delta,$$

其中, $Q_{i,j}$ 表示轮廓线 $Q = (q_1, q_2, \dots, q_n)$ 中的一个子段 $(q_i, q_{i+1}, \dots, q_j)$ 。子段的搜索可以用原始的方法来计算所有 $Q_{i,j+m-1}^j$ 的 RMS 距。但是, 它的时间复杂度为 $O(Nmn)$, 其中 n 为轮廓线 Q_j 的长度。

1.4 哈希向量的条件

在描述新的哈希向量之前, 我们先来看一下任何哈希向量都应该满足的条件。在传统的哈希方法里, 一个整数对应一个物体。但是, 对于轮廓线, 我们要用一个实数向量来对应一个固定长度的子段。对每一个子段长度为 H 的子段 $P = (p_1, \dots, p_H)$, 有一个哈希向量 $hs(P)$ 与之对应。 $hs(p)$ 应该满足下面两个条件:

1) $hs(P)$ 对 P 来说应该是旋转、平移不变。

2) 如果 $d_{rms}(P, Q)$ 很小, 那么 $hs(P)$ 应该接近 $hs(Q)$ 。

注意到一旦一个向量给定, 只有当 $hs(P)$ 应该接近 $hs(Q)$ 时, 才需要计算 $d_{rms}(P, Q)$ 。最小二乘哈希方法只满足条件 1), 而不满足条件 2)。

定义 2 给出一个新的哈希向量, 用 HASH(A)

来表示。

HASH(A):

$$hs(P) = (c_1(p), s_1(p), \dots, c_d(p), s_d(p)),$$

其中 $c_k(P) = \alpha \sum_{i=1}^H \|p_i - c\| (\beta + \cos \frac{2\pi k(i-1)}{H})$,

$$s_k(P) = \alpha \sum_{i=1}^H \|p_i - c\| (\beta + \sin \frac{2\pi k(i-1)}{H}),$$

c 是轮廓线 p 的质心 ($\frac{1}{n} \sum_{i=1}^n p_i$), $\|p_i - c\|$ 表示第 i 个点到质心的距离, $\alpha (\alpha > 0), \beta (\beta \geq 0)$ 是固定的实数, d 是固定的整数, 这在后面给出。

$hs(P)$ 对应与点到质心的傅立叶变换的低频成分, 图 2 表示空间轮廓曲线的哈希矢量值。后面的定理 1 可以证明 $hs(P)$ 满足条件 1) 和条件 2)。

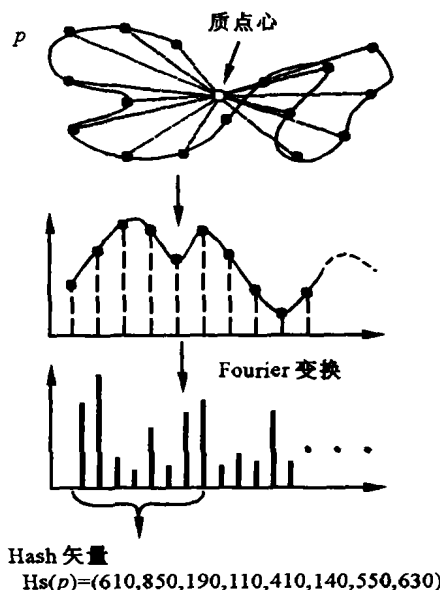


图 2 轮廓曲线的哈希矢量表示
Fig. 2 Computation of a Hash vector

2 算法实现

引理 1 设 $P = (p_1, p_2, \dots, p_H)$ 和 $Q = (q_1, q_2, \dots, q_H)$ 是经过变换后质心重合的轮廓线序列, 那么有

$$|\sum_{i=1}^H \|p_i\| - \sum_{i=1}^H \|q_i\|| \leq H d_{rms}(P, Q).$$

证明 假设 $\hat{P} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_H)$ 是对 P 经过旋转后的序列, 有 $d(\hat{P}, Q, I, 0) = d_{rms}(P, Q)$, 其中 0 表示 0 矢量, I 表示单位矩阵, 那么下面的不等式成立

$$\begin{aligned} & \left| \sum_{i=1}^H \|p_i\| - \sum_{i=1}^H \|q_i\| \right| = \\ & \left| \sum_{i=1}^H \|\hat{p}_i\| - \sum_{i=1}^H \|q_i\| \right| \leq \\ & \sum_{i=1}^H \left| \|\hat{p}_i\| - \|q_i\| \right| \leq \sum_{i=1}^H \|\hat{p}_i - q_i\| \end{aligned}$$

后面的不等式由三角不等式得到。又由于

$$t_1 + t_2 + \dots + t_H \leq \sqrt{H} \sqrt{t_1^2 + t_2^2 + \dots + t_H^2},$$

其中 $t_1 \geq 0, t_2 \geq 0, \dots, t_H \geq 0$ 成立。

$$\text{而 } \sum_{i=1}^H \|\hat{p}_i - q_i\| \leq \sqrt{H} \sqrt{\sum_{i=1}^H \|\hat{p}_i - q_i\|^2} =$$

$$\sqrt{H} \cdot \sqrt{H} \sqrt{\frac{1}{H} \sum_{i=1}^H \|\hat{p}_i - q_i\|^2} =$$

$$\sqrt{H} \cdot \sqrt{H} \cdot d_{rms}(P, Q) = Hd_{rms}(P, Q)$$

成立,下面给出定理。

定理 1 对于所有的 k 来说有

$$|C_k(P) - C_k(Q)| \leq H\alpha(1 + \beta)d_{rms}(P, Q),$$

$$|S_k(P) - S_k(Q)| \leq H\alpha(1 + \beta)d_{rms}(P, Q).$$

成立。

证 明 不失一般性,假定 P 和 Q 都是经过变化后质心重合的曲线,那么利用引理 1 有下面的不等式

$$\begin{aligned} |C_k(P) - C_k(Q)| & \leq \alpha \sum_{i=1}^H (\beta + \\ & \cos \frac{2\pi k(i-1)}{H}) \cdot \left| \|p_i\| - \|q_i\| \right| \leq \end{aligned}$$

$$\alpha(1 + \beta) \sum_{i=1}^H \left| \|p_i\| - \|q_i\| \right| \leq$$

$$H\alpha(1 + \beta)d_{rms}(P, Q)$$

成立。因此,对于 $C_k(P), C_k(Q)$ 定理证明成立,同理也可以证明对于 $s_k(P), s_k(Q)$ 也是成立的。对于所有的 $k(1 \leq k \leq d)$ 来说,如果 $|C_k(P) - C_k(Q)| \leq \gamma$ 和 $|s_k(P) - s_k(Q)| \leq \gamma$ 同时成立,我们记做 $\text{COND}(P, Q, \gamma)$ 成立。从上面的定理可以得到有下面的性质成立:

性质 1 对于 $\gamma = H\alpha(1 + \beta)\delta$ 来说,如果 $\text{COND}(P, Q, \gamma)$ 不成立,那么有

$$d_{rms}(P, Q) > \delta.$$

因此说,如果 $\text{COND}(P, Q, \gamma)$ 不成立,就有 $d_{rms}(P, Q) > \delta$,进而 RMS 距离的计算是没有必要的。这一性质对于子段的快速搜索来说是非常有用的。

对于 $hs(P)$ 的计算来说需要 $O(H)$ 次,由于我

们假定 d 是一个固定小的值,所以对于 $\text{COND}(P, Q, \gamma)$ 的判断是有限次的。

3 利用哈希矢量对子段的查询

我们利用 $hs(P)$ 对匹配子段的查询是十分快捷的,查询主要可以分两个阶段进行:预处理阶段和查询阶段。

3.1 预处理阶段

对于任意的测试轮廓线 Q^j ,假设有 n 个点构成,那么对于它的哈希矢量的计算如下:

1) for $i = 1$ to $n - H + 1$, repeat 2);

2) 计算 $hs(Q_{i,i+H-1}^j)$ 。

其中 H 为一常量可以选定。

3.2 查询阶段

对于给定的基轮廓线 P ,假设有 m 个点构成,其中要求 $m \geq H$ 。那么对于基轮廓线上其中一个子段的匹配查询可以通过下面的途径。首先计算 $hs(P_{1,H})$ 对于所有的子段 $Q_{i,i+H-1}^j$,比较 $hs(P_{1,H})$ 和 $hs(Q_{i,i+H-1}^j)$ 。如果条件 $\text{COND}(P_{1,H}, Q_{i,i+H-1}^j, \gamma)$ 满足,计算 $d_{rms}(P, Q_{i,i+m-1}^j)$ 并判断。否则, $d_{rms}(P, Q_{i,i+m-1}^j)$ 不再计算,进而测试下一轮廓线段。下面我们归纳查询阶段的过程。用 $|Q^j|$ 表示轮廓线 Q^j 的点数,假设共有 N 段测试曲线。

1) 计算 $hs(p_{1,H})$;

2) for $j = 1$ to N , repeat 3) ~ 5);

3) for $i = 1$ to $|Q^j|$, repeat 4, 5);

4) if $\text{COND}(p_{1,H}, Q_{i,i+H-1}^j, \gamma)$ 成立, then do 5);

5) if $d_{rms}(P, Q_{i,i+H-1}^j) \leq \delta$, then output Q^j ,

$$j = j + 1.$$

对于上面的过程来说,它的运算时间中,由于 1) 的时间很小,对于 5) 的判断最多可以执行最大 i 次。因此我们只考虑 2) ~ 5), 对于每一条轮廓 Q^j 线来说时间最多为 $O(n)$ 。因此对于 N 条测试轮廓线来说运算时间最多为 $O(Nn)$ 。

4 实验与结论

在我们完成计算机辅助文物复原系统中,利用上面算法得到文物碎片匹配的结果,如图 3 所示。

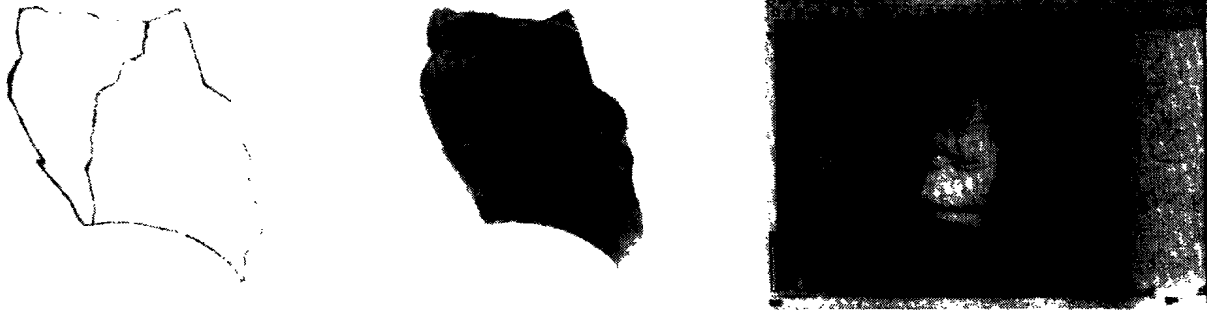


图 3 轮廓曲线匹配结果

Fig. 3 Result of curve matching

在本研究中,证明了两条曲线的距离越小,所得到的哈希矢量的距离值也越小,在对离散轮廓线顶点作 Fourier 变换的基础上,采用新的哈希技术来查找曲线段之间的匹配段,不仅在理论上证明了算法的科学性,从实验中也证明了所采用方法运行高效,快速,稳定。本文所给方法不仅可以用于空间两条曲线的匹配,还可应用于曲线相似度比较、几何特征提取、物体识别、数字文档等多个领域。

参考文献:

- [1] SEBASTIAN T B, KLEIN P N. Alignment-based recognition of shape outlines[J]. *J of Computer Vision*, 1992, 10(1): 79-104.
- [2] KISHON E, HASITE T, WOLFLSON H. 3D curve matching using splines [J]. *J of Robotic Systems*, 1991, 8(6): 723-743.
- [3] ANDRE G. Smoothing and matching of 3-D Space Curves [J]. *J of Computer Vision*, 1994, 12(1): 79-104.
- [4] UMEYAMA S. Least-squares estimation of transformation parameters between two point patterns [J]. *IEEE PAMI*, 1991, 13(4): 376-380.
- [5] PAUL J B. A method for registration of 3-D shapes [J]. *IEEE PAMI*, 1992, 14(2): 64-71.
- [6] BHANU B. Shape matching of two-dimensional objects [J]. *IEEE PAMI*, 1984, 6(2): 36-45.

(编辑 曹大刚)

Rapid 3D contour matching based on the Fourier transform

LÜ Ke, GENG Guo-hua, ZHOU Ming-quan, ZHAO Hong-an

(Department of Computer Science, Northwest University, Xi'an 710069, China)

Abstract: A novel method searching for similar fragments of 3D curves is presented. In this method, a hash vector (a vector of real numbers) is associated with each fixed-length fragments of 3D sherds. Each vector consists of low frequency components of Fourier-Like spectrum for the distance between profile curve and the centroid. Then the similarity between fragments by evaluating the difference between Hash vectors can be analyzed. The novel aspect of the method is that the following property is proved theoretically; if the distance between two fragments of curve is small, then the distance between the hash vectors is small.

Key words: contour matching; Hash function; Fourier transform.