

# 用于快速特征点配准的聚类凸集投影算法

连 玮 梁 彦 程咏梅 潘 泉 张洪才

(西北工业大学自动化学院, 西安 710072)

**摘 要** 为降低特征点配准的计算量,提出了一种聚类凸集投影算法。该算法首先通过聚类将模板点集和目标点集的点配准问题转化为相应的类配准问题,然后将序贯凸集投影算法用于求解该问题,从而得到一种聚类的凸集投影算法。它可以看作是序贯凸集投影算法结合聚类思想而得到的推广。由于该算法的误差和计算量取决于类半径的大小,因此在点密度较大的情况下,通过适当选择类半径,可明显降低计算量,而精度只有少许降低。仿真结果表明,该算法是有效的。

**关键词** 聚类 配准 凸集投影

中图法分类号: TP391.4 文献标识码: A 文章编号: 1006-8961(2007)03-0505-06

## Clustering Successive POCS for Fast Point Matching

LIAN Wei, LIANG Yan, CHENG Yong-mei, PAN Quan, ZHANG Hong-cai

(School of Automation, Northwestern Polytechnical University, Xi'an 710072)

**Abstract** A clustering successive projection onto convex sets algorithm is presented for fast point matching. Via feature point clustering, the problem of matching two point sets is converted to that of matching corresponding clusters, which is then solved by a tailored successive projection onto convex sets (SPOCS) algorithm. The resulting algorithm can be viewed as an extension of SPOCS by combining with clustering. Its precision and computational complexity are decided by the clustering radius. Under the condition that the point sets' density is high, by choosing a proper radius, the computational burden can be reduced with only negligible deterioration of precision. Experimental results demonstrate the effectiveness of the algorithm.

**Keywords** clustering, registration, projection onto convex sets

## 1 引言

点配准是指要找到两幅图像中的特征点之间的对应关系。它在机器视觉、对地遥感以及医学图像处理等领域均有重要的应用。现有的主流点配准算法是基于能量函数最小化的原理,其中,迭代最近点(iterative closest point, ICP)算法<sup>[1]</sup>属于比较简单的一种。该算法在迭代的每一步,先由最近邻来确定离散的对应关系,再由对应关系来更新空间变换。该算法简单、计算量小,但它粗糙的对应关系求解方

法使得能量函数极易陷入局部极小。针对ICP算法的缺点,出现了各种改进策略,其中比较重要的一种是将对应关系松弛成允许取连续值。文献[2~4]用一个概率密度估计模型来描述点配准,EM(expectation-maximization)算法被用于求解配准问题,其中E步根据当前的空间变换来估计对应关系;M步则根据当前的对应关系来更新空间变换。该算法存在的问题是不能保证一对一的对应关系。鲁棒点匹配(robust point matching, RPM)算法<sup>[5~8]</sup>则将点先配准建模成一个线性指派——最小二乘优化问题,然后采用确定性退火来进行寻优,并采用交

基金项目:国家自然科学基金项目(60404011、60372085)

收稿日期:2005-11-07;改回日期:2006-04-06

第一作者简介:连玮(1978~),男,2003年获西北工业大学硕士学位,现为西北工业大学博士研究生。主要从事图像处理、机器视觉等方面的研究。E-mail: lianwei3@gmail.com

替行列归一化来使对应关系满足一一对应的约束。针对确定性退火和交替行列归一化方法比较复杂的缺点,文献[9]提出一种梯度下降结合凸集投影(projection onto convex sets, POCS)的优化算法。序贯凸集投影(successive POCS, SPOCS)被用于使对应关系满足一一对应的约束。SPOCS 算法首先将约束凸集分解为行约束凸集和列约束凸集的交,而对应关系矩阵向整个约束凸集的投影则通过它交替地向行约束凸集和列约束凸集投影来逼近得到。

本文在 SPOCS 算法的基础上,提出了一种快速的聚类凸集投影算法。该算法首先通过聚类,将模板点集和目标点集的点配准问题转化为相应的类配准问题,然后将 SPOCS 算法用于求解该问题,从而得到一种聚类的凸集投影算法。它可以看作是引入聚类思想的 SPOCS 算法。该算法的配准精度和计算量取决于类半径的大小。仿真结果表明,该算法是有效的。

## 2 点配准问题的优化模型

一般地,在  $d$  维空间中讨论问题。模板点集定义为  $X = \{x_i, i = 1, \dots, n\}$ , 目标点集定义为  $Y = \{y_j, j = 1, \dots, m\}$ 。并且假设  $X$  中的任一点在  $Y$  中有且只有一个对应点,则有  $n \leq m$ 。

这里采用  $n \times m$  大小的点的对应关系矩阵  $P = [p_{i,j}]$  来描述点的对应关系,其中元素  $p_{i,j}$  表示  $X$  中的点  $x_i$  和  $Y$  中的点  $y_j$  之间的对应程度。矩阵  $P$  的约束集  $C_P$  定义为

$$C_P = \left\{ \begin{array}{l} \sum_{j=1}^m p_{i,j} = 1, \quad i = 1, \dots, n \\ p_{i,j}: \sum_{i=1}^n p_{i,j} \leq 1, \quad j = 1, \dots, m \\ p_{i,j} \geq 0, \quad \forall i, j \end{array} \right\} \quad (1)$$

采用文献[8]的模型,点配准问题可以表述为求使能量函数

$$J(\Omega, P) = \sum_{i,j} p_{i,j} \|y_j - \Omega(x_i)\|^2 + \Phi(\Omega)$$

最小的空间映射  $\Omega$  和对应关系矩阵  $P \in C_P$ 。其中  $\|\cdot\|$  表示矩阵 2 范数,函数  $\Phi(\Omega)$  表示对  $\Omega$  取值的限制。若固定  $\Omega$ ,则该问题等价为一个线性指派问题;若固定  $P$ ,则该问题等价为一个最小二乘问题。若  $\Omega$  是参数映射,  $J$  是  $\Omega$  的凸函数,且  $\frac{\partial J}{\partial \Omega} = 0$  有显式解  $\Omega_0$ ,则将  $\Omega = \Omega_0$  带入到函数  $J$  中就得到以下

只含  $P$  的函数:

$$\tilde{J}(P) = J(\Omega_0, P) \quad (2)$$

然后就可以用文献[9]的梯度下降结合凸集投影的算法对  $\tilde{J}(P)$  进行寻优:

$$\text{初始化: } P^{(0)} = \frac{\mathbf{1}_{n \times m}}{m}$$

while( $w < N$  且  $\delta > \xi$ ) 或 ( $w < 3$ )

$$\tilde{P}^{(w+1)} = P^{(w)} - \eta \left[ \frac{\nabla \tilde{J}(P^{(w)})}{\|\nabla \tilde{J}(P^{(w)})\|} \right] \quad // \text{梯度}$$

下降

$$P^{(w+1)} = T_{C_P}(\tilde{P}^{(w+1)}) \quad // \text{凸集投影}$$

$$\delta = \text{tr}((P^{(w+1)} - P^{(w)})^T (P^{(w+1)} - P^{(w)}))$$

$$w \leftarrow w + 1$$

end while

其中,  $\mathbf{1}_{n \times m}$  为元素全为 1 的  $n \times m$  大小的矩阵,  $w$  ( $w < N$ ) 为迭代次数,  $\xi$  为收敛尺度,  $\eta$  为步长。  $\text{tr}$  表示对方阵求迹。  $\nabla \tilde{J}(P^{(w)})$  表示函数(式(2))在迭代  $w$  次矩阵  $P^{(w)}$  处的梯度矩阵,  $T_{C_P}(P^{(w)})$  表示矩阵  $P^{(w)}$  到约束凸集  $C_P$  的投影,其定义为

$$T_{C_P}(P^{(w)}) = \arg \min_{P \in C_P} \|P - P^{(w)}\|$$

以下定理描述了  $T_{C_P}(P^{(w)})$  的存在唯一性:

**定理 1** 若  $C$  为 Hilbert 空间  $H$  的任一非空凸集,则任给  $P^{(w)} \in H$ ,必存在唯一的  $P_0 \in C$ ,使得

$$\|P_0 - P^{(w)}\| = \min_{Z \in C} \|Z - P^{(w)}\|$$

## 3 算法描述

凸集投影是上述寻优算法的关键,这是因为  $T_{C_P}(P^{(w)})$  没有闭形式解。文献[9]的 SPOCS 算法首先将  $C_P$  分解为  $C_P = C_P^r \cap C_P^c$ , 其中,

$$C_P^r = \left\{ \begin{array}{l} \sum_{j=1}^m p_{i,j} = 1 \quad i = 1, \dots, n \\ p_{i,j} \geq 0 \quad \forall i, j \end{array} \right\} \text{为行约束凸}$$

$$\text{集}; C_P^c = \left\{ \begin{array}{l} \sum_{i=1}^n p_{i,j} \leq 1 \quad j = 1, \dots, m \\ p_{i,j} \geq 0 \quad \forall i, j \end{array} \right\} \text{为列约束}$$

凸集,由于  $P^{(w)}$  向  $C_P^r$  (上角  $r$  代表 row,下同)和  $C_P^c$  (上角  $c$  代表 column,下同)的投影都有闭形式解,因此  $T_{C_P}(P^{(w)})$  可通过  $P^{(w)}$  向  $C_P^r$  和  $C_P^c$  交替投影来逼近得到。该算法的成立是基于以下定理及推论:

**定理 2** 任给凸集  $C_i, i = 1, \dots, m$ , 如果

$C_0 = \bigcap_{i=1}^m C_i$  非空, 且定义算子  $T = T_{C_m} T_{C_{m-1}} \cdots T_{C_1}$ , 其中  $T_{C_i}$  为到凸集  $C_i$  的投影算子, 则任给  $P^{(w)} \in H$ , 序列  $\{T^n(P^{(w)})\}$  弱收敛于  $C_0$  中的一点。

**推论 1** 在上述定理的条件下, 如果  $C_i, i=1, \dots, m$  是闭子空间或线性簇, 则任给  $P^{(w)} \in H$ , 序列  $\{T^n(P^{(w)})\}$  强收敛于  $T_{C_0}(P^{(w)})$ , 即  $P^{(w)}$  到  $C_0$  的投影。

SPOCS 算法的计算复杂度取决于待配准的两点集的势, 当该值比较大时, 计算量将比较大。对此, 本文提出将聚类的思想引入到凸集投影中, 用来降低计算复杂度。

首先对点集  $X$  进行聚类得到类集合  $\Gamma^X = \{\Gamma_k^X, k=1, \dots, \hat{n}\}$ , 其中  $\Gamma_k^X$  为由第  $k$  类中的点的标号构成的集合; 同样地, 对点集  $Y$  进行聚类得到类集合  $\Gamma^Y = \{\Gamma_l^Y, l=1, \dots, \hat{m}\}$ 。

继而采用  $\hat{n} \times \hat{m}$  大小的类对应关系矩阵  $Q = [q_{k,l}]$  来描述类对应关系, 其中元素  $q_{k,l}$  表示类  $\Gamma_k^X$  和类  $\Gamma_l^Y$  的对应程度。类对应关系的约束集  $C_Q$  定义为

$$C_Q = \left\{ q_{k,l} : \begin{cases} \sum_{l=1}^{\hat{m}} | \Gamma_l^Y | q_{k,l} = 1 & k = 1, \dots, \hat{n} \\ \sum_{k=1}^{\hat{n}} | \Gamma_k^X | q_{k,l} \leq 1 & l = 1, \dots, \hat{m} \\ q_{k,l} \geq 0 & \forall k, l \end{cases} \right. \quad (3)$$

其中,  $|\cdot|$  表示集合的势。  $C_Q$  显然是凸的。集合  $C_p$  与  $C_Q$  存在以下关系:

$$P \in C_p \Rightarrow \left[ \frac{\sum_{i \in \Gamma_k^X, j \in \Gamma_l^Y} p_{i,j}}{|\Gamma_k^X| |\Gamma_l^Y|} \right] \in C_Q$$

$n \times m$  大小的对应关系矩阵  $P^{(w)}$  到约束凸集  $C_Q$  的投影定义为

$$T_{C_Q}(P^{(w)}) = \arg \min_{Q \in C_Q} \sum_{k,l} \sum_{i \in \Gamma_k^X, j \in \Gamma_l^Y} (q_{k,l} - p_{i,j}^{(w)})^2 \quad (4)$$

这样, 通过该定义, 对应关系矩阵  $P^{(w)}$  到约束凸集  $C_p$  的投影问题就转化为  $P^{(w)}$  到约束凸集  $C_Q$  的投影问题。

解出  $T_{C_Q}(P^{(w)})$  后,  $T_{C_p}(P^{(w)})$  可通过以下步骤来近似得到

如果  $\exists k, l$ , 使  $i \in \Gamma_k^X, j \in \Gamma_l^Y$ , 则  $p_{i,j} \leftarrow q_{k,l}$ 。

### 3.1 聚类凸集投影的求解

可参考 SPOCS 算法思想来求解  $T_{C_Q}(P^{(w)})$ , 类似地, 可将  $C_Q$  分解为  $C_Q = C_Q^r \cap C_Q^c$ , 其中,

$$C_Q^r = \left\{ q_{k,l} : \begin{cases} \sum_{l=1}^{\hat{m}} | \Gamma_l^Y | q_{k,l} = 1 & k = 1, \dots, \hat{n} \\ q_{k,l} \geq 0 & \forall k, l \end{cases} \right\} \text{ 为行约束}$$

$$\text{凸集; } C_Q^c = \left\{ q_{k,l} : \begin{cases} \sum_{k=1}^{\hat{n}} | \Gamma_k^X | q_{k,l} \leq 1 & l = 1, \dots, \hat{m} \\ q_{k,l} \geq 0 & \forall k, l \end{cases} \right\} \text{ 为}$$

列约束凸集。仿照式(4)定义, 可将  $P^{(w)}$  向  $C_Q^r$  和  $C_Q^c$  的投影定义为  $T_{C_Q^r}(P^{(w)})$  和  $T_{C_Q^c}(P^{(w)})$ 。于是  $T_{C_Q}(P^{(w)})$  可通过  $P^{(w)}$  向  $C_Q^r$  和  $C_Q^c$  交替投影来逼近得到。

$\hat{n} \times \hat{m}$  大小的类对应关系矩阵  $Q^{(w)} = [q_{k,l}^{(w)}]$  到约束凸集  $C_Q^r$  的投影定义为

$$T_{C_Q^r}(Q^{(w)}) = \arg \min_{Q \in C_Q^r} \sum_{k,l} | \Gamma_k^X | | \Gamma_l^Y | (q_{k,l} - q_{k,l}^{(w)})^2 \quad (5)$$

这里采用相同的符号  $T_{C_Q}$  作为  $P^{(w)}$  或  $Q^{(w)}$  到约束凸集  $C_Q^r$  的投影算子, 其含义可根据自变量的不同而加以区分。同样地可定义  $Q^{(w)}$  到约束凸集  $C_Q^c$  的投影  $T_{C_Q^c}(Q^{(w)})$ 。可以证明下述定理是成立的:

#### 定理 3

$$\begin{aligned} T_{C_Q^r}(P^{(w)}) &= T_{C_Q^r} \left( \left[ \frac{\sum_{i \in \Gamma_k^X, j \in \Gamma_l^Y} p_{i,j}^{(w)}}{|\Gamma_k^X| |\Gamma_l^Y|} \right] \right) \\ T_{C_Q^c}(P^{(w)}) &= T_{C_Q^c} \left( \left[ \frac{\sum_{i \in \Gamma_k^X, j \in \Gamma_l^Y} p_{i,j}^{(w)}}{|\Gamma_k^X| |\Gamma_l^Y|} \right] \right) \end{aligned} \quad (6)$$

证明思路可参考文献[10]。

因此  $T_{C_Q}(P^{(w)})$  可通过  $\left[ \frac{\sum_{i \in \Gamma_k^X, j \in \Gamma_l^Y} p_{i,j}^{(w)}}{|\Gamma_k^X| |\Gamma_l^Y|} \right]$  向  $C_Q^r$  和

$C_Q^c$  交替投影来逼近得到, 计算步骤描述如下:

$$\text{初始化: } \hat{k} = 1, \hat{\delta} > 0.001, Q = \left[ \frac{\sum_{i \in \Gamma_k^X, j \in \Gamma_l^Y} p_{i,j}^{(w)}}{|\Gamma_k^X| |\Gamma_l^Y|} \right]$$

while ( $\hat{\delta} > 0.001$  且  $\hat{k} < \hat{m}$ )

$$\hat{Q} = Q$$

$$Q \leftarrow T_{C_Q^r}(Q)$$

$$Q \leftarrow T_{C_Q^c}(Q)$$

$$\hat{\delta} = \text{tr}((\hat{Q} - Q)^T (\hat{Q} - Q))$$

$$\hat{k} \leftarrow \hat{k} + 1$$

end while

其中,  $T_{C_Q^r}(Q)$  和  $T_{C_Q^c}(Q)$  的计算方法可参考文献[9]的推证过程类似地得到, 附录列出了它们的

Matlab 伪代码。

对比本文算法和 SPOCS 算法可知,本文算法可以看作是引入聚类思想的 SPOCS 算法,其处理的对象就变成了类,而不再是点。

### 3.2 计算复杂度分析

不妨假设  $n = m$  及  $\hat{n} = \hat{m}$ , 则 SPOCS 算法在最坏情况下,计算复杂度为  $O(n^4)$ 。本文算法通过引入聚类可使计算复杂度降为  $O(\hat{n}^4)$ 。

## 4 实验

取  $\Omega$  为仿射变换,  $\Phi(\Omega)$  为  $\Omega$  的线性部分与单位阵之间距离的平方。实验参数的选择同文献[9]。聚类算法采用文献[11]的最远点聚类法。类半径  $R$  分别取成模板点集宽度的 0.1、0.15 和 0.2 倍。

实验采用文献[8]的实验方法来测试本文算法和 SPOCS 算法的性能。图 1(a) 的点集被作为模板点集。算法分以下 3 项:(1)模板点集通过不同程度的非刚性形变(通过高斯基函数产生,其系数服从期望为 0 的高斯分布,标准方差取 0.01 到 0.06 的数)来生成目标点集,目的是测试算法抗形变的能力;(2)不同标准方差的高斯白噪声(标准方差取 0.01 到 0.06 的数)被加给经过中等形变的模板点集,用来生成目

标点集,目的是测试算法抗噪声的能力;(3)不同数量的随机出格点(出格点个数同模板点个数的比率取 0.5 到 2.5 的数)被加给经过中等形变的模板点集,用来生成目标点集,目的是测试算法抗出格点的能力。误差定义为应用算法得到的变形模板与目标点集之间的均方距离。算法采用 Matlab 语言,在 Pentium 4 2.6GHz CPU,256MB RAM 的 PC 机上进行实验。每项实验进行 100 次,得到的平均误差和标准方差如图 1(b)、图 1(c)和图 1(d)所示。

由图 1 可见,对应不同  $R$  值的本文算法和 SPOCS 算法的误差之间的差别相对恒定,且误差随着类半径  $R$  的增大而增大。在测试形变和噪声的实验中,所有算法的误差随形变程度或噪声强度的增加都增长不大,这表现出对形变和噪声影响的鲁棒性。而在测试出格点的实验中,所有算法的误差则随出格点比率的增加而增长比较快,可见本文算法和 SPOCS 算法都较易受出格点的影响。

从计算量来看,上述实验中,本文算法在  $R = 0.1$  时,平均计算时间为 9.0232s,在  $R = 0.15$  时,为 3.7185s,在  $R = 0.2$  时,为 1.6391s。可见计算量随着类半径的增大和类的个数的减少而减少。

图 2 给出上述测试的一个例子, $R$  取成模板点集宽度的 0.1 倍。图 2(a)和图 2(b)为对模板点集

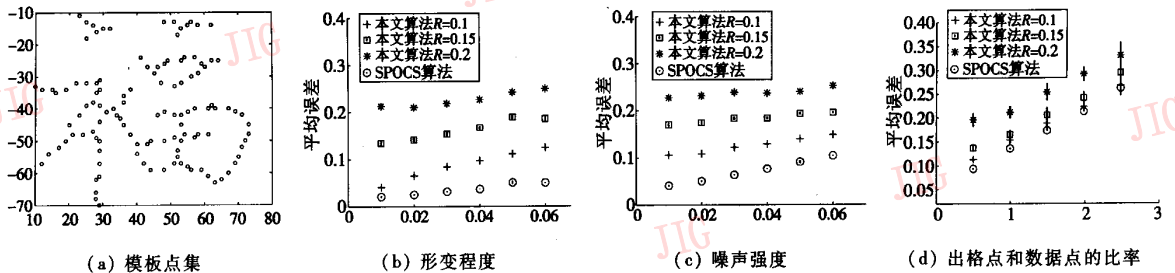


图 1 本文算法与 SPOCS 算法的平均误差及方差(由误差条的长度表示)

Fig.1 Statistics of the synthetic experiments

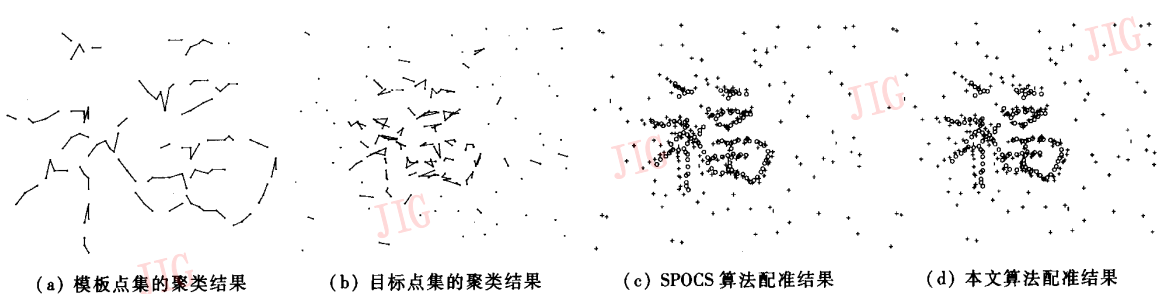


图 2 模板点集和目标点集的聚类和配准结果

Fig.2 Clustering and matching results of the template and target

和目标点集进行聚类的结果,该结果被本文算法所利用。类内点由线段进行了连接。图 2(c)为 SPOCS 算法的配准结果,图 2(d)为本文算法的配准

结果。由此可见,聚类只导致算法精度的少许降低。图 3 给出了一个配准图像特征点的例子。 $R$  的选择同上。同样地,聚类只导致了算法精度的少许降低。

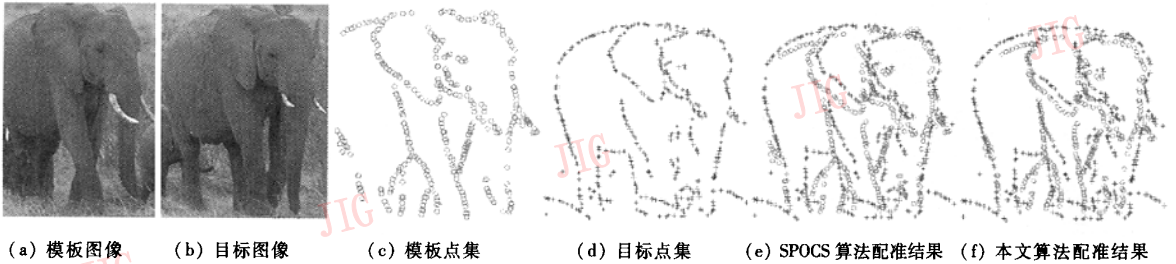


图 3 图像提取特征点再配准的结果

Fig. 3 Matching of feature points extracted from images

## 5 结 论

本文提出一种用于特征点配准的快速聚类凸集投影算法。该算法先将模板点集和目标点集的配准问题通过聚类转化为相应类集合之间的配准问题,然后将序贯凸集投影算法应用于该问题而得到一种聚类的凸集投影算法。它可以看作是引入聚类思想的序贯凸集投影算法。由于该算法的误差和计算量取决于类半径的大小,因此在点密度较大的情况下,通过选择适当的类半径,可获得计算量的明显改善,而精度只有少许降低。仿真结果表明,该算法是有效的。

### 参考文献 (References)

- 1 Besl P J, McKay N D. A method for registration of 3-D shapes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239 ~ 256.
- 2 Cross A D J, Hancock E R. Graph matching with a dual-step EM algorithm[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(11): 1236 ~ 1253.
- 3 Hinton G E, Williams C K I, Revow M D. Adaptive elastic models for hand-printed character recognition [A]. In: Moody John E, Hanson Steve J, Lippmann Richard P, eds. Advances in Neural Information Processing Systems(NIPS)[C], San Mateo, CA, USA: Morgan Kaufmann, 1992: 512 ~ 519.
- 4 Wells W. Statistical approaches to feature-based object recognition [J]. International Journal of Computer Vision, 1997, 21(1): 63 ~ 98.
- 5 Chui H, Rambo J, Duncan J, et al. Registration of cortical anatomical structures via robust 3D point matching [A]. In: Proceedings of the International Conference on Information Processing

in Medical Imaging (IPMI) [C], London UK: Springer-verlag, 1999: 168 ~ 181.

- 6 Gold S, Rangarajan A, Lu C P, et al. New algorithms for 2-D and 3-D point matching; pose estimation and correspondence[J]. Pattern Recognition, 1998, 31(8): 1019 ~ 1031.
- 7 Rangarajan A, Chui H, Bookstein F L. The softassign procrustes matching algorithm [A]. In: Proceedings of the 15th International Conference on Information Processing in Medical Imaging (IPMI) [C], London, UK, Springer-verlag, 1997: 29 ~ 42.
- 8 Chui H, Rangarajan A. A new point matching algorithm for non-rigid registration[J]. Computer Vision and Image Understanding, 2003, 89(2-3): 114 ~ 141.
- 9 Van Wyk B J, Van Wyk M A. A POCS-based graph matching algorithm[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(11): 1526 ~ 1530.
- 10 Van Wyk B J. Kronecker product, successive projection, and related graph matching algorithms[D]. In: University of the Witwatersrand, Johannesburg, 2003[EB/OL], <http://www.ee.wits.ac.za/comms/output/theses.htm>.
- 11 Gonzalez T. Clustering to minimize the maximum intercluster distance [J]. Theoretical Computer Science, 1985, 38(2-3): 293 ~ 306.

## 附 录

计算  $T_{c_q}(Q)$  的伪代码如下:

for  $i = 1 : \hat{n}$

$\varphi = \text{dot}(Q(i, :), X_Y)$ ;

$\sigma = \text{sum}(X_Y)$ ;

$\bar{q} = Q(i, :)$ ;

$[s, d] = \text{sort}(\bar{q})$ ;

for  $j = 1 : \hat{m}$

$s(j) = s(j) + (1 - \varphi) / \sigma$ ;

if  $s(j) < 0$

```

s(j) = 0;
 $\varphi = \varphi - \bar{q}(d(j)) * |\Gamma_{d(j)}^x|;$ 
 $\sigma = \sigma - |\Gamma_{d(j)}^x|;$ 
end
 $\bar{q}(d(j)) = s(j);$ 
end
 $Q(i, :) = \bar{q};$ 
end
计算  $T_{c^0}(Q)$  的伪代码如下:
for j = 1:  $\hat{m}$ 
 $\varphi = \text{dot}(Q(:, j), \chi_x);$ 
if  $\varphi > 1$ 
 $\sigma = \text{sum}(\chi_x);$ 
 $\bar{q} = Q(:, j);$ 
[ $s, d$ ] = sort( $\bar{q}$ );

```

```

for i = 1:  $\hat{n}$ 
 $s(i) = s(i) + (1 - \varphi) / \sigma;$ 
if  $s(i) < 0$ 
 $s(i) = 0;$ 
 $\varphi = \varphi - \bar{q}(d(i)) * |\Gamma_{d(i)}^x|;$ 
 $\sigma = \sigma - |\Gamma_{d(i)}^x|;$ 
end
 $\bar{q}(d(i)) = s(i);$ 
end
 $Q(:, j) = \bar{q};$ 
end
end
其中向量  $\chi_y = [|\Gamma_1^y|, \dots, |\Gamma_m^y|]$ ,  $\chi_x = [|\Gamma_1^x|, \dots, |\Gamma_n^x|]$ 。

```