

RANSAC 算法的自适应 $T_{c,d}$ 预检验

田 文 王宏远 徐 帆 方 磊

(华中科技大学电子与信息工程系, 武汉 430074)

摘 要 随机抽样一致性算法是计算机视觉领域应用最广泛的鲁棒性算法。为了进一步提高 RANSAC 算法的运算速度, 首先在介绍 RANSAC 算法的 $T_{c,d}$ 预检验加速模型的基础上, 提出了一种两步法用来实现优化的预检验参数选择; 然后基于这种优化选择方法提出了自适应 $T_{c,d}$ 预检验的新算法, 从而实现了不依赖用户选择的 RANSAC 算法的自适应加速。基于窄基线和宽基线图像对的极线几何计算的实验表明, 该新算法相对于标准 RANSAC 算法的运算速度平均提升超过了 400%。

关键词 随机抽样一致性算法 预检验 鲁棒性估计 基础矩阵 局部优化

中图法分类号: TP391.4 **文献标识码**: A **文章编号**: 1006-8961(2009)05-973-05

Enhanced RANSAC with Adaptive Pre-verification

TIAN Wen, WANG Hong-yuan, XU Fan, FANG Lei

(Department of Electronics and Information Engineering Huazhong University of Science and Technology, Wuhan 430074)

Abstract RANSAC is the most widely used robust regression algorithm in computer vision. Starting from the $T_{c,d}$ pre-evaluation model of RANSAC algorithm, a two-step method is presented for optimal (c, d) selection. Based on this method, the adaptive $T_{c,d}$ test extension is proposed to achieve user independent RANSAC acceleration. We show experimentally that using both short-baseline and wide-baseline epipolar geometry estimation, the proposed method is up to 400% faster than the standard RANSAC.

Keywords random sample consensus (RANSAC), pre-evaluation, robust estimation, fundamental matrix, local optimization

1 引 言

Fischler 和 Bolles 于 1981 年提出的随机抽样一致性算法 (RANSAC)^[1] 在计算机视觉领域有着十分广泛的应用。RANSAC 算法的最大优点是能够处理传统鲁棒性方法所不能处理的误匹配点 (outlier) 比率大于 50% 的情况, 因此其在很多计算机视觉应用中已经成为标准配置; 应用范围包括基础矩阵估计^[2]、三焦张量估计^[3]、运动分割^[4]、图像拼接^[5]以及特征图像匹配^[6]等。

RANSAC (random sample consensus) 算法的运

行时间依赖于求解模型参数的维度和数据的 outlier 比率, 随着这两者的增大, 为了保证算法的置信度, 致使 RANSAC 算法所需的循环次数快速上升。RANSAC 算法的主要性能瓶颈在于对模型的评估, 因为每一次随机抽样计算模型参数后, 需要计算每一个数据的余差。

近几年的很多研究都针对 RANSAC 算法的效率问题提出了加速算法。Chum 等人提出了基于局部优化 (local optimization) 的 LO-RANSAC 算法^[7-8], 并通过加速算法收敛来减少循环次数。Matas 和 Chum 等人在文献[9]中指出, 模型评估不需要对所

收稿日期: 2007-07-02; 改回日期: 2007-12-05

第一作者简介: 田 文 (1983 ~), 女。华中科技大学电子与信息工程系通信与信息系统专业博士研究生。主要从事图像 3 维重建研究。

E-mail: twbiggs@gmail.com

通讯作者: 王宏远。E-mail: wythylw@public.wh.hb.cn

有数据都计算代价函数,同时提出了 $T_{d,d}$ 预检验模型,并指出模型必须首先通过预检验才能继续完成全部的评估过程,由于错误模型通过预检验的几率很低,因此算法避免了大量不必要的评估过程,实验表明,算法效率提升了超过 100%。随后陈付幸和王润生以及 Nister 等人在文献[9]的 $T_{d,d}$ 预检验模型的基础上,提出了更加一般化的 $T_{c,d}$ 预检验^[10-11], Nister 还通过简单的实验指出, $T_{c,d}$ 预检验模型在大多数情况下能够取得比 $T_{d,d}$ 预检验模型更好的加速效果。Capel 通过大量的实验指出,在 RANSAC 算法中加入局部优化和预检验过程并不会影响 RANSAC 算法的置信度和精度^[12]。

本文主要针对 $T_{d,d}$ 预检验模型以及 $T_{c,d}$ 预检验模型目前存在的以下两个问题:

(1) 预检验参数的优化取值问题,如何选取 c 和 d ,以使算法达到最快的运行速度。

(2) 预检验算法是假设数据的 inlier 比率 ε 和错误模型的平均相容率 δ 已知,而如何在算法执行过程中,动态地估算 ε 和 δ 是保证预检验算法正确运行的关键问题。

本文通过分析首先指出, $T_{c,d}$ 预检验算法的极值特性,然后提出了一个近似优化的两步法用来估算预检验参数 c 和 d ,并提出了 ε 和 δ 的动态估算方法,最终给出了整套 RANSAC 算法的优化加速方案——自适应 $T_{c,d}$ 预检验 (adaptive $T_{c,d}$ test)。

2 RANSAC 算法的 $T_{c,d}$ 预检验模型

RANSAC 算法是随机算法,它是在全部 N 个数据中通过循环随机抽取最小数据集集中的 m 个数据来估计模型参数。

假设 N 个数据中共有 I 个正确的匹配点 (inlier), inlier 比率为 $\varepsilon = I/N$,在 $N \gg m$ 的情况下一次抽样为无污染抽样的概率为 ε^m 。RANSAC 算法在循环计算 l 次后就能够保证在置信度 p 下取得至少一次无污染抽样,从而可保证得到正确的模型参数。算法置信度 p 和所需循环次数 l 之间有如下关系:

$$l = \left\lceil \frac{\log(1-p)}{\log(1-\varepsilon^m)} \right\rceil \quad (1)$$

RANSAC 算法的每一次循环包括一次模型参数计算和 N 次代价函数计算,以一次代价函数计算为基本时间单位,记模型参数估计所需的时间为 t_M (下角 M 代表 model),则 RANSAC 算法所需的总时间为

$$t = l(N + t_M) \quad (2)$$

所谓预检验是指在模型评估之前,首先在全数据中随机抽取一小部分数据用来对模型进行预检验,如果预检验通过,则继续进行模型评估,如果没有通过预检验,则放弃后面的评估过程。加入预检验过程后,对一个模型的平均验证时间为

$$t_V = \varepsilon^m (\alpha N + (1-\alpha)\bar{t}_\alpha) + (1-\varepsilon^m) (\beta N + (1-\beta)\bar{t}_\beta) \quad (3)$$

其中, α 是正确模型通过预检验的概率, β 是错误模型通过预检验的概率,好的预检验模型应该让正确模型通过预检验的概率大,而错误模型通过预检验的概率小,因此一般情况下, $\alpha \gg \beta$ 。 $\bar{t}_\alpha$ 是正确模型预检验失败的平均时间,而 $\bar{t}_\beta$ 则是错误模型预检验失败的平均时间, RANSAC 算法所需的总时间为

$$t = l(\bar{m}_s t_V + t_M) \quad (4)$$

$\bar{m}_s$ 为求解模型参数所得解的平均个数。例如,基础矩阵正规化 8 点算法只有 1 个解,而 7 点算法有 1 个或 3 个解。 $\bar{m}_s$ 和 t_M 均为实验统计结果。

预检验 RANSAC 算法需要在置信度 p 下,能够保证 l 次抽样中至少有一次是无污染抽样,并能通过预检验

$$l = \left\lceil \frac{\log(1-p)}{\log(1-\varepsilon^m \alpha)} \right\rceil \quad (5)$$

下面给出 $T_{c,d}$ 预检验的定义:

定义 1 ($T_{c,d}$ 预检验) $T_{c,d}$ 预检验是指在数据中随机抽取 d 个数据用来对模型进行预检验,如果至少 c 个数据通过,则为通过预检验,否则为预检验失败。

正确模型和错误模型分别通过 $T_{c,d}$ 预检验的概率 α 和 β 遵从超几何分布,在假设 $N \gg d$ 的情况下可以用以下二项分布来近似:

$$\alpha = 1 - \sum_{i=0}^{c-1} \binom{d}{i} \varepsilon^i (1-\varepsilon)^{d-i} \quad (6)$$

$$\beta = 1 - \sum_{i=0}^{c-1} \binom{d}{i} \delta^i (1-\delta)^{d-i} \quad (7)$$

其中, δ 是错误模型的平均相容率,指的是单个数据与错误模型相容的平均概率。每次 $T_{c,d}$ 预检验不用检验全部 d 个数据,当不通过的数据达到 $d-c+1$ 时,则判定预检验失败,即可退出预检验,因此有

$$(1-\alpha)\bar{t}_\alpha = \sum_{i=d-c+1}^d \binom{i-1}{d-c} i (1-\varepsilon)^{d-c+1} \varepsilon^{i-(d-c+1)} \quad (8)$$

$$(1-\beta)\bar{t}_\beta = \sum_{i=d-c+1}^d \binom{i-1}{d-c} i (1-\delta)^{d-c+1} \delta^{i-(d-c+1)} \quad (9)$$

综合式 (5) ~ 式 (9),并代入式 (3),则整个算法的

执行时间就可以直接计算,若取 $N = 1\ 000$, $I = 500$, $m = 8$, $\delta = 0.05$, $t_M = 20$,则算法的总运行时间 t 与预检验参数 d 和 c 的关系如图 1 所示,由图 1 可见,算法的执行时间曲面呈盆地状,这说明取 $c = d$ 的 $T_{c,d}$ 预检验并不是最优化的选择。图 1 也验证了文献[11]的推断。

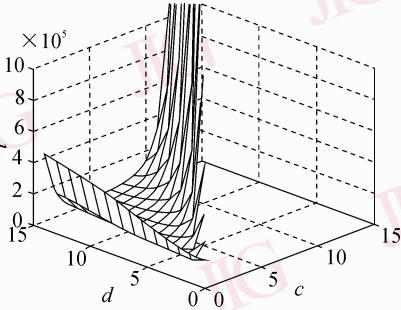


图 1 $T_{c,d}$ 预检验 RANSAC 算法的执行时间

Fig. 1 Total time spent by RANSAC with $T_{c,d}$ pre-verification

3 预检验参数 c, d 的优化估计

算法的总运行时间 t 虽然可以直接计算,但由于展开后是一个关于 c 和 d 的极其复杂的表达式,因此想要采用近似求导的常规方法或者是制表法来求解 c, d 都十分困难;况且 $T_{c,d}$ 预检验的目的是提高算法效率,因此采用任何影响效率的迭代算法来估计 c 和 d 都是不能接受的。本文提出了一种近似优化的两步法用来求解 c 和 d ,将这个复杂的 2 维优化问题转化为两个 1 维的优化问题,步骤如下:

- (1) 在 $c = d$ 的约束下求解 c 使 t 最小化。
- (2) 通过固定 c 求解 d 来使 t 最小化。

3.1 参数 c 的优化估计

将 $c = d$ 代入式(6)~式(9)有

$$\alpha = \varepsilon^c \text{ 和 } \beta = \delta^c$$

$$(1 - \alpha) \bar{l}_\alpha = \sum_{i=1}^d i(1 - \varepsilon) \varepsilon^{i-1}$$

$$(1 - \beta) \bar{l}_\beta = \sum_{i=1}^d i(1 - \delta) \delta^{i-1}$$

由于 α 值较大而 δ 值很小,因此有如下近似:

$$(1 - \alpha) \bar{l} \approx \sum_{i=1}^{\infty} i(1 - \varepsilon) \varepsilon^{i-1} = \frac{1}{1 - \varepsilon}$$

$$(1 - \beta) \bar{l}_\beta \approx 1$$

代入式(4)则总时间 t 有如下近似值:

$$t \approx l(\bar{m}_s \left(\varepsilon^m \left(\varepsilon^c N + \frac{1}{1 - \varepsilon} \right) + \right.$$

$$\left. (1 - \varepsilon^m)(\delta^c N + 1) \right) + t_M) \quad (10)$$

由于 $\varepsilon^c N \gg \delta^c N$,且文献[9]中已经证明,用 l 取平均化的 $1/\varepsilon^m \varepsilon^c$ 来求解 c 并不影响算法的置信度。因此有

$$t \approx \frac{1}{\varepsilon^m \varepsilon^c} \left(\bar{m}_s (\varepsilon^{m+c} N + \delta^c N + \frac{\varepsilon^{m+1}}{1 - \varepsilon} + 1) + t_M \right) \quad (11)$$

对上式求导,使 $\partial t / \partial c = 0$, c 的近似优化取值如下:

$$c^* = \frac{\log \left(\frac{\log \varepsilon (t_M + \bar{m}_s + \bar{m}_s \varepsilon^{m+1} / (1 - \varepsilon))}{\bar{m}_s N (\log \delta - \log \varepsilon)} \right)}{\log \delta} \quad (12)$$

由于 c 是大于等于 1 的整数,因此其最终取值为

$$c_{\text{opt}} = \max(1, \lfloor c^* \rfloor) \quad (13)$$

3.2 参数 d 的优化估计

好的预检验模型应该让正确模型通过的概率大,而尽量阻止错误模型,因此预检验模型的参数设计应该遵循如下的原则:

$$(1) c/d \rightarrow \varepsilon$$

(2) c/d 应该随 δ 的增大而减小,以减少错误模型的预检验通过率。

因此给出 d 的近似优化取值为

$$d^* = \frac{c^*}{\varepsilon} - \frac{\log \varepsilon}{\log \delta} \quad (14)$$

由于 d 是大于等于 c 的整数,因此其最终取值为

$$d_{\text{opt}} = \max(c_{\text{opt}}, \lfloor d^* \rfloor) \quad (15)$$

两步法当然有另一种求解顺序:先求解 d ,再求解 c 。即在 $c = d$ 约束下求出 d 的最优值,再沿 c 轴优化。实验证明,第 2 种方案中, c 的取值范围太小,优化的可行性不大。

T_{adaptive} 预检验 RANSAC 算法的详细步骤如下:

- (1) 从全部 N 个数据中随机抽取 m 个数据。
- (2) 用抽取的 m 个数据来估计模型参数 M 。
- (3) 从剩余的 $N - m$ 个数据中抽取 d 个数据对参数 M 做 $T_{c,d}$ 预检验,若通过,则跳至步骤 4,若不通过,则跳至步骤 6。

(4) 继续使用剩余的 $N - d$ 个数据来评估模型参数 M ,并计算代价函数 ρ ,若 $\rho < \rho_{\text{best}}$,则跳至步骤 5,否则跳至步骤 6。

(5) 保存最好的模型参数 $M_{\text{best}} = M$,及其代价函数 $\rho_{\text{best}} = \rho$,并统计估算 ε ,跳至步骤 7。

(6) 统计估算 δ 。

(7) 依式(13)、式(15)更新预检验参数。

(8) 依式(18) 累计算法的置信度,同时判断当前的累计置信度是否已经达到阈值 p ,若达到,则退出循环,算法输出 M_{best} ;若未达到,则跳至步骤 1 继续循环。

4 自适应 $T_{c,d}$ 预检验

4.1 δ 和 ε 的统计估算

由于实际应用中是无法预先得知 δ 和 ε 的取值的,为此 T_{adaptive} 预检验 RANSAC 算法是通过动态统计估算 δ 和 ε 的方法来实现自适应优化的 $T_{c,d}$ 预检验。 δ 是错误模型的平均相容率,错误模型包括算法步骤(3)中的预检验没有通过的模型和步骤(4)中被淘汰的模型。统计评估这些模型的总数据的个数 n_{eval} 和相容数据的个数 n_{pass} ,则 $\delta = n_{\text{pass}} / n_{\text{eval}}$ 。 ε 是数据中的 inlier 比率, $\varepsilon = I_{\text{best}} / N$,其为最优模型 M_{best} 相容数据的个数 I_{best} 和总数据的个数 N 的比值。随着 ε 和 δ 的收敛,最终预检验参数 (c,d) 也会收敛到一个优化的取值上,从第 5 节的实验结果中也可以看到这个优化的预检验取值确实取得了很好的效果。

4.2 算法停止条件

T_{adaptive} 预检验 RANSAC 算法在运行过程中,由于 c 和 d 的取值会发生变化,因此使用式(5)作为算

法的停止条件是不准确的。RANSAC 类算法的停止条件是要求累计的置信度达到阈值 p ,本文对停止条件做一下改进,预检验 RANSAC 算法改进后的停止条件如下:

1 - (1 - ε^m α)^l ≥ p (16)

并有

llog(1 - ε^m α) ≤ log(1 - p) (17)

由于 T_{adaptive} 预检验 RANSAC 算法在每次循环中, c 和 d 的取值可能会发生变化,即导致 α 会依式(6)改变,因此停止条件可有如下形式:

Σ log(1 - ε^m α_i) ≤ log(1 - p) (18)

在算法实现过程中,可通过当前的预检验参数 c,d 值来计算 α_i ,并求和累计就能准确地保证算法的置信度。

5 实验结果

为了检验基于 RANSAC 的 T_{adaptive} 预检验算法的加速效果,本文采用如图 2 所示的几幅图像进行了几种算法的匹配实验。通过基础矩阵的鲁棒性估计实验来验证 T_{adaptive} 预检验 RANSAC 算法的实际加速效果,实验对比了下列算法:

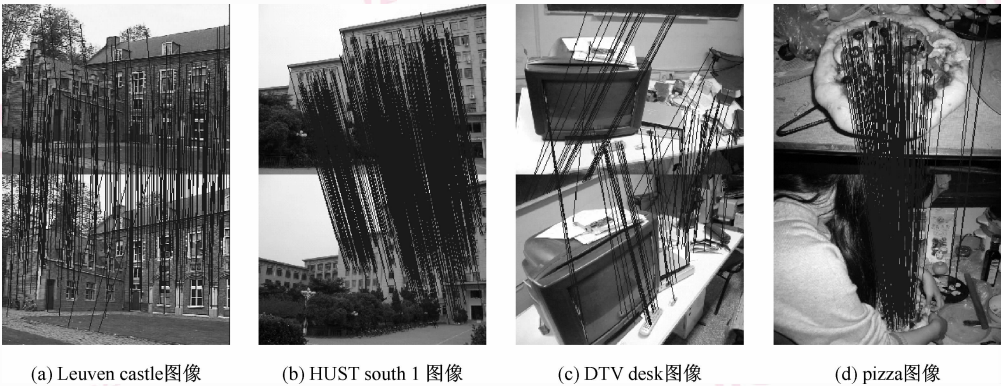


图 2 图像匹配
Fig. 2 Result of stereo matching

- (1) 标准 RANSAC 算法。
- (2) $T_{d,d}$ 预检验 RANSAC 算法,取 $d = 1$ 。
- (3) T_{adaptive} , 本文提出的自适应 $T_{c,d}$ 预检验 RANSAC 算法。

实验数据采用了窄基线的图像对 Leuven castle 和 HUST south1 以及宽基线的 DTVdesk 和 pizza。窄基线图像匹配采用子像素精度的 Harris 角点探测

器^[13]来输出特征点,并通过计算零均值正规化互相关(ZNCC)进行匹配,同时采用 7 点算法计算基础矩阵;宽基线图像匹配则采用 SIFT 算法^[14]输出匹配点,同时采用正规化 8 点算法^[15]计算基础矩阵,并将计算得到的 Sampson 误差^[16]作为余差;阈值 T 设为 1pixel,置信度 $p = 0.95$ 。实验重复 30 次取平均值。算法输出的各个图像对的正确匹配点如图 2

所示,各个图像对的总匹配点数和算法统计的 ε 、 δ 和 $\bar{m}_s$ 值见表 1,最终的实验结果如表 2 所示。从实验结果可以看出, T_{adaptive} 算法相对于标准 RANSAC 算法不仅大幅减少了模型评估的计算量,使计算余差的次数减少了 10 倍以上,并且没有像 $T_{d,d}$ 预检验一样出现模型计算次数大幅上升的情况。另外, inlier 数和平均余差都在正常波动范围以内,这说明预检验和局部优化的 RANSAC 增强技术并没有影响算法的置信度。

表 1 匹配点个数及 $\varepsilon, \delta, \bar{m}_s$ 统计值

Tab. 1 Number of correspondences and the converged $\varepsilon, \delta, \bar{m}_s$

图像	N	ε	δ	$\bar{m}_s$
Leuven castle	448	0.46	0.11	1.57
HUST South 1	1 622	0.46	0.28	1.63
DTV desk	232	0.39	0.06	1
pizza	381	0.44	0.18	1

表 2 实验结果

Tab. 2 Experiment result

图像	算法	模型 个数	评估 次数	运行时 间(s)	Inlier 数	平均 余差
Leuven castle	RANSAC	1 610	721 589	16.4	204	0.14
	$T_{1,1}$	3 068	41 814	3.5	204	0.13
	T_{adaptive}	1 923	22 593	2.4	204	0.13
HUST south1	RANSAC	989	1 604 158	34.6	766	0.20
	$T_{1,1}$	3 057	244 641	8.6	748	0.15
	T_{adaptive}	1 972	116 859	3.9	757	0.16
DTV desk	RANSAC	6 358	1 475 148	36.6	91	0.22
	$T_{1,1}$	14 313	94 399	23.1	92	0.20
	T_{adaptive}	5 547	139 063	13.6	92	0.19
pizza	RANSAC	2 287	871 270	19.8	167	0.19
	$T_{1,1}$	4 876	85 419	8.7	167	0.17
	T_{adaptive}	2 788	60 506	6.8	168	0.16

6 结 论

本文针对 RANSAC 算法的效率问题,提出了一种自适应 $T_{e,d}$ 预检验方法,即根据输入数据优化地调整预检验参数,进而实现了不依赖用户的自适应算法加速框架。通过实验证明,结合局部优化的自适应 $T_{e,d}$ 预检验 RANSAC 算法相对于标准 RANSAC

算法速度提升了 400%。

参考文献 (References)

1 Fischler M A, Bolles R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications ACM, 1981, **24**(6): 381-395.

2 Torr P H S, Murray D W. The development and comparison of robust methods for estimating the fundamental matrix [J]. International Journal of Computer Vision, 1997, **24**(3): 271-300.

3 Torr P H S, Zisserman A. Robust parameterization and computation of the trifocal tensor[J]. Image and Vision Computing, 1997, **15**(8): 591-607.

4 Torr P H S. Outlier Detection and Motion Segmentation [D]. University of Oxford, England, 1995.

5 McLauchlan P F, Jaenicke A. Image mosaicing using sequential bundle adjustment [J]. Image and Vision Computing, 2002, **20**(9-10): 751-759.

6 Leonardis A, Bischof H. Robust recognition using eigenimages[J]. Computer Vision and Image Understanding, 2000, **78**(1): 99-118.

7 Chum O, Matas J, Kittler J. Locally optimized RANSAC [A]. In: Michaelis B, Krell G. eds. Proceedings of the 25th DAGM Symposium. [C], Berlin, Germany: Springer-Verlag, 2003: 236-243.

8 Chum O, Matas J, Obdrzalek S. Enhancing RANSAC by generalized model optimization [A]. In: Hong K S, Zhang Z. eds. Proceedings of the Asian Conference on Computer Vision (ACCV) [C], Seoul, Korea South: Asian Federation of Computer Vision Societies, 2004, **2**: 812-817.

9 Matas J, Chum O. Randomized RANSAC with $T_{d,d}$ test [J]. Image and Vision Computing, 2004, **22**(10): 837-842.

10 Chen Fu-xin, Wang Run-sheng. Fast RANSAC with preview model parameters evaluation [J]. Journal of Software, 2005, **16**(08): 1431-1438. [陈付幸, 王润生. 基于预检验的快速随机抽样一致性算法[J]. 软件学报, 2005, **16**(08): 1431-1438.]

11 Nister D. Preemptive RANSAC for live structure and motion estimation [J]. Machine Vision and Applications, 2005, **16**(5): 321-329.

12 Capel D P. An effective bail-out test for RANSAC consensus scoring [A]. In: Clocksin W, Fitzgibbon A, Torr P, eds. Proceedings of Conference on British Machine Vision [C], Oxford, England: British Machine Vision Association, 2005: 629-638.

13 Harris C, Stephens M. A combined corner and edge detector [A]. In: Proceedings of 4th ALVEY Vision Conference [C], Manchester, UK: Blackwell Publishing, 1988: 147-151.

14 Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, **60**(2): 91-110.

15 Hartley R I. In defense of the eight-point algorithm [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, **19**(6): 580-593.

16 Zhang Z. On the optimization criteria used in two-view motion analysis [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, **20**(7): 717-729.