

基于直线段对应的相机位姿估计直接最小二乘法

李 鑫^{1,2} 张跃强^{1,2} 刘进博^{1,2} 张小虎^{1,2} 于起峰^{1,2}

¹国防科学技术大学航天科学与工程学院, 湖南 长沙 410073

²国防科学技术大学图像测量与视觉导航湖南省重点实验室, 湖南 长沙 410073

摘要 为了实现高精度的基于直线段对应的相机位姿估计, 提出一种直接最小二乘法。通过提出并利用一种直线段之间的距离测度, 将原问题转化为最小化一个姿态旋转矩阵的二次目标函数, 该距离测度综合考虑了线段的端点距离、中点距离、夹角和线段的长度, 通过旋转矩阵的CGR参数表达获得一个修正的目标函数, 此修正目标函数的最优解条件组成了一个三元三次方程组, 利用代数多项式方法在不需要迭代的情况下直接求解这个方程组, 从而得到了旋转矩阵的全局最优解。该算法的计算复杂度为 $O(n)$ 。仿真和真实实验验证了该方法的有效性和高精度。

关键词 机器视觉; 相机位姿估计; 直线段对应; 多项式方程组; 全局最优解

中图分类号 O436

文献标识码 A

doi: 10.3788/AOS201535.0615003

A Direct Least Squares Method for Camera Pose Estimation Based on Straight Line Segment Correspondences

Li Xin^{1,2} Zhang Yueqiang^{1,2} Liu Jinbo^{1,2} Zhang Xiaohu^{1,2} Yu Qifeng^{1,2}

¹College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, Hunan 410073, China

²Hunan Key Laboratory of Videometrics and Vision Navigation, National University of Defense Technology, Changsha, Hunan 410073, China

Abstract In order to estimate the camera pose from straight line segment correspondences in high precision, a direct least squares solution is proposed. A straight line segment distance measurement which integrates distance between endpoints, midpoints, angle and lengths of two line segments is proposed and used to transform the original problem into minimize a quadratic cost function of pose rotation matrix. A modified cost function based on the CGR parameters of rotation matrix is minimized, which the optimality conditions form a system of three third-order polynomial equations. Algebraic method is used to solve this system directly without requiring iterations, and the globally optimal solution of rotation matrix is obtained. The complexity of proposed method is $O(n)$. The efficiency and high precision are demonstrated by synthesized and real data experiments.

Key words machine vision; camera pose estimation; straight line segment correspondence; polynomial equations; globally optimal solution

OCIS codes 150.1488; 150.6910; 120.4640

1 引 言

相机位姿估计问题是指在给定相机内参数, 已知3D模型特征和其在像面上的投影, 求解相机位置和姿态的问题^[1]。若采用点特征, 则称为n点透视(PnP)问题; 若为直线特征, 则称为n线透视(PnL)问题。PnP问题已经在近些年得到了显著的发展^[2-6]。而对PnL问题, 由于直线特征能够更加稳健的提取, 并且更多的反映参考模型的结构信息, 能够较少的受噪声和遮挡影响, 因此, 近些年也得到了广泛的研究。

收稿日期: 2014-12-31; 收到修改稿日期: 2015-03-03

基金项目: 国家973计划(2013CB733100)、国家自然科学基金(11272347、11472302)

作者简介: 李 鑫(1985—), 男, 博士研究生, 主要从事大型结构变形测量及机器视觉等方面的研究。E-mail: lixin_nudt@163.com

导师简介: 于起峰(1958—), 男, 教授, 中国科学院院士, 主要从事空天图像测量与视觉导航方面的研究。

E-mail: yuqifeng@vip.sina.com

通常,求解 PnL 问题的算法分为线性算法和迭代算法。线性算法速度较快,但精度不高。而迭代算法对某个目标函数进行迭代优化,一般精度较高,但计算量大。

PnL 问题所需要的最少线数为 3(P3L)。文献[7-10]均得到了基于直线方向约束的 8 次多项式方程,也就是 P3L 问题最多有 8 组解。当 $n > 3$ 时,Ansar 等^[11]通过线性求解旋转矩阵的各个元素,其计算复杂度为 $O(n^2)$,且其精度对图像噪声很敏感。Mirzaei 等^[12]提出了利用代数多项式方程方法求解一个修正的方向误差函数,从而确定了旋转矩阵的全局最优解,之后线性求解平移向量。Zhang 等^[10]参考了 $RPnP^{[2]}$ 的方法,提出一种稳健高效的方法,选择一个旋转轴,把问题转化为 3 条直线一组的情况。每一组都存在一个 8 次多项式,然后组合起来求解一个 15 阶次的方程来求解这个旋转轴,建立线性方程组求解剩余的旋转参数和平移向量,采用绝对定向规范相机的位姿。Liu 等^[13]利用了一种角度深度信息求解相机姿态。张跃强等^[14]提出了结合粒子滤波跟踪和 E-估计优化的位姿跟踪估计算法。

迭代算法通常要最小化某种代价函数。通常代价函数可以选择在像方,也可以选择在物方。针对此,Kumar 等^[15]提出了多种不同的代价函数,并提出一种能够同时求解旋转和平移的迭代算法。Taylor 等^[16]定义了一种二维(2D)线段和 2D 直线的距离后,首次给出了 PnL 问题的重投影像方误差目标函数,并且给出了一种优化策略。Christy 等^[17]提出一种针对仿射(弱透视或参数弱透视)相机模型的迭代方法,该方法需要较好的初值,否则计算量很大。

针对未知 2D 直线与三维(3D)直线对应的情况,很多文章都采取了类似文献[18]方法的思想,依次迭代求解位姿关系和对应关系。David 等^[19]根据弱透视相机模型提出了 SoftPOSIT 算法。Zhang 等^[20]也提出了一种同时求解位姿和对应关系的算法,针对位姿求解的部分,其最小化直线方向误差与线上某个点的共面误差的加权和,迭代方法参考了文献[5],提出直线对应的正交迭代方法。

本文针对在对应关系已知的情况下如何求解相机位姿的问题,提出一种直线段之间的距离测度,该距离测度综合考虑了线段的端点距离、中点距离、夹角和线段的长度。利用此将原问题转化为最小化一个旋转矩阵的非线性目标函数,利用旋转矩阵的 CGR 参数表达修正目标函数,用代数多项式方法线性求出其全局最优解。虽然与文献[12]使用了相同的代数多项式方法,但提出了更加合理的距离测度和目标函数。文献[12]先求旋转后求平移,势必使得旋转矩阵的求解误差传递到平移的误差,而本文方法由于定义了合理的距离测度,并推导出旋转矩阵已经确定的情况下平移向量存在最优线性解,从而将原问题转化为最小化一个旋转矩阵的二次目标函数。本文算法具有:1)可以保证获得全局最优解且精度高;2)不需要初值;3)计算复杂度为 $O(n)$ 。

2 观测模型和目标函数

2.1 观测模型

位姿估计的目标就是为了获得相机坐标系相对于世界坐标系的变换关系。观测量为已知模型 3D 直线在图像上的 2D 投影直线。由于模型一般为有限大,且图像像面也有限,所以,此处 3D 和 2D 直线实际上是有限的 3D 和 2D 直线段。

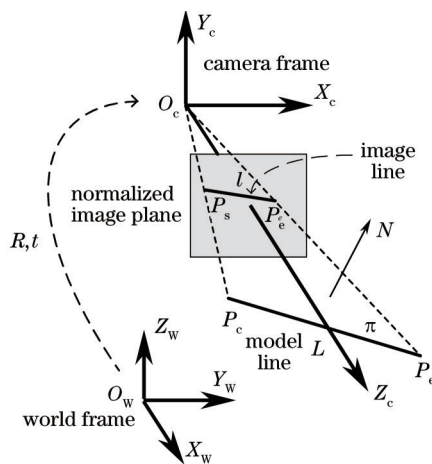


图 1 模型 3D 直线透视投影

Fig.1 Perspective projection of 3D model line

如图 1 所示,世界坐标系 $O_w - X_w Y_w Z_w$ 建立在模型上,且已知模型直线在世界坐标系下的坐标。 L 为 3D 模型直线段, P_s 、 P_e 为两个端点坐标。相机坐标系 $O_c - X_c Y_c Z_c$ 则建立在相机上,原点在相机光心, Z_c 轴沿光轴指向相机前方, R, t 为相机坐标系的旋转矩阵和平移向量。由于相机内参已知,则对线段的提取结果可以认为是在归一化像面上,归一化像面指相机坐标系 $Z_c = 1$ 的平面。 l 为对应的投影在归一化像面上的 2D 直线段, p_s 、 p_e 为两个端点坐标。 π 表示 l 与光心确定的平面, N 表示该平面法向单位矢量。

在没有噪声的情况下,两组对应端点应该满足共线方程,但是通常直线段提取时,端点沿着直线方向的定位误差较大^[15],因此实际中使用的是共面方程,即相机光心、图像 2D 直线和模型 3D 直线三者在同一平面内。可以表示为

$$N^T(RP + t) = 0, \quad (1)$$

式中 P 为 3D 直线上的任意一点在世界坐标系下的坐标。由于 2D 直线段两个端点在相机坐标系下的坐标分别为 $[p_s^T \ 1]^T$ 、 $[p_e^T \ 1]^T$, 则

$$N = \frac{[p_s^T \ 1]^T \times [p_e^T \ 1]^T}{\|[p_s^T \ 1]^T \times [p_e^T \ 1]^T\|}. \quad (2)$$

两点确定一条直线,因此只需要把 3D 直线的两个端点代入(1)式中即可对共面性完全描述。由于直线也可以使用直线上的任意一点加直线的方向进行描述,因此共面描述还有其他形式,如关于 3D 直线的方向的观测方程为

$$N^T Rn = 0, \quad (3)$$

式中 $n = (P_e - P_s) / \|P_e - P_s\|$ 为 3D 直线在世界坐标系下的方向矢量。

2.2 直线段之间距离测度

为了建立直线段测量位姿估计的目标函数,需要定义一种直线段之间的距离测度。

每一条线段存在两个端点,设其是有方向的,也就是定义了一个端点为起点,另一个端点为终点。设线段 LS_1 表示为 $x = (x_s, x_e)$, 其中 x_s 为起点, x_e 为终点。线段 LS_2 表示为 $y = (y_s, y_e)$, 其中 y_s 为起点, y_e 为终点。

则线段 LS_1 、 LS_2 上的参数化的一般点为

$$x(t) = x_s + t(x_e - x_s) = tx_e + (1-t)x_s, t \in [0, 1], \quad (4)$$

$$y(t) = y_s + t(y_e - y_s) = ty_e + (1-t)y_s, t \in [0, 1], \quad (5)$$

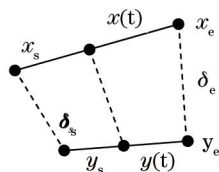


图 2 两条直线段上的对应点

Fig.2 Point correspondence between two straight line segments

则两个有向直线段之间的距离定义为

$$d(LS_1, LS_2) = \sqrt{\int_0^1 \|y(t) - x(t)\|^2 dt}. \quad (6)$$

假设两条有向线段内部的点存在对应关系,并且是按照其距离两个端点的长度比例对应的,也就是说 $x(t)$ 与 $y(t)$ 对应,若 $t=0$,则表示起点对应起点,若 $t=1$,则终点对应终点,然后取对应点欧式距离平方的平均值的 2 次根。将(4)式和(5)式代入(6)式,得到

$$d^2(LS_1, LS_2) = \int_0^1 \|(y(t) - x(t))\|^2 dt = \int_0^1 (at^2 + bt + c) dt = \frac{1}{3}a + \frac{1}{2}b + c, \quad (7)$$

式中 $a = \|[y_e - x_e - (y_s - x_s)]\|^2$, $b = 2[y_e - x_e - (y_s - x_s)]^T (y_s - x_s)$, $c = \|(y_s - x_s)\|^2$ 。若设 $\delta_s = y_s - x_s$, $\delta_e = y_e - x_e$, $\delta = [\delta_s^T \ \delta_e^T]^T$, 则容易推导出

$$d^2(LS_1, LS_2) = \frac{1}{3}(\|\delta_s\|^2 + \delta_s^T \delta_e + \|\delta_e\|^2) = \delta^T B \delta = \|A \delta\|^2, \quad (8)$$

式中 $B = \begin{bmatrix} \frac{1}{3}I & \frac{1}{6}I \\ \frac{1}{6}I & \frac{1}{3}I \end{bmatrix}$, $A = \begin{bmatrix} \frac{\sqrt{3}}{6}I & -\frac{\sqrt{3}}{6}I \\ \frac{1}{2}I & \frac{1}{2}I \end{bmatrix}$, I 为线段所在维度的单位矩阵。从(8)式可以看出,此处定义的线段

距离实际上是一种马氏距离。

若考虑两条线段的中点, $x_m = (x_s + x_e)/2$, $y_m = (y_s + y_e)/2$, 且 $\delta_m = y_m - x_m$, 则也容易得到

$$d^2(LS_1, LS_2) = \frac{1}{6}(\|\delta_s\|^2 + 4\|\delta_m\|^2 + \|\delta_e\|^2). \quad (9)$$

可见,两条线段之间的距离为起点、中点和终点三个距离加权和,且中点的权重最大。这与传统的只考虑两个端点的距离^[15,21]的情况不同,且直观理解,考虑中点更符合实际。

若设 $S_x = \|x_s - x_e\|$ 、 $S_y = \|y_s - y_e\|$ 分别表示两条线段的长度, $n_x = (x_e - x_s)/S_x$ 、 $n_y = (y_e - y_s)/S_y$ 分别表示两条线段的指向, θ 表示夹角。则

$$d^2(LS_1, LS_2) = \frac{1}{6}(\|\delta_s\|^2 + 4\|\delta_m\|^2 + \|\delta_e\|^2) = \frac{1}{6} \left[4\|\delta_m\|^2 + \left\| x_m - \frac{S_x}{2}n_x - (y_m - \frac{S_y}{2}n_y) \right\|^2 + \left\| x_m + \frac{S_x}{2}n_x - (y_m + \frac{S_y}{2}n_y) \right\|^2 \right] = \|\delta_m\|^2 + \frac{1}{12}(S_x^2 + S_y^2) - \frac{1}{6}S_x S_y \cos \theta \quad (10)$$

从(10)式可以得到,两条线段中点距离越小、长度越小、夹角越小,则距离越小。为了更好的理解此处线段的距离的定义,比较图3三种情况下的两条线段的距离。



图3 两条线段距离的三个例子

Fig.3 Three examples of distance of two straight line segments

图3(a)中,两个线段长度相同且平行,且 $\delta_s = \delta_e$, 得到 $d_a^2 = \|\delta_s\|^2$ 。图3(b)中,两个线段长度相同且中点交叉,则 $\delta_s = -\delta_e$, 得到 $d_b^2 = \frac{1}{3}\|\delta_s\|^2$ 。图(c)中, $\delta_e = 0$, 则 $d_c^2 = \frac{1}{3}\|\delta_s\|^2$ 。因此有 $d_a^2 = 3d_b^2 = 3d_c^2$ 。针对图3(a),即使这两条线段平行,但是其沿垂直线段方向的距离却很大。直观上,把直线段当作点集,那么此处两个直线段之间的距离就是两个点集之间对应点的距离的一种平均,因此认为图3(a)的距离相对较大。

若使用传统的仅用两个端点的距离^[15]表示,则图3(a)与图3(b)的两条线段之间的距离相同,显然不合理,并没有考虑两条线段之间的夹角情况。另外,若使用指向误差与某一点(一般取中点)误差的和^[20],显然其并没有考虑线段的长度的影响。

从上述分析可以看出,此距离定义本身综合考虑了两条线段的端点距离、中点距离、夹角和长度,更加符合实际,并且形式上也较为简洁,方便后续列写统一的目标函数。

2.3 目标函数

在定义了直线段距离的情况下,只需要最小化模型线段到其在相机坐标系下的估计线段之间的距离,就可以对相机进行位姿估计。

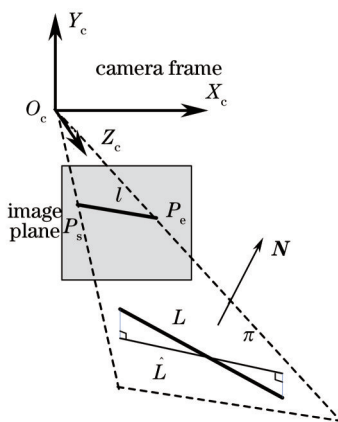


图4 空间直线段在平面上的投影

Fig.4 Projection from line segment to plane

如图4所示,定义 \hat{L} 为直线段 L 在平面 π 上的估计。满足 \hat{L} 的两个端点是 L 的两个端点在平面 π 上的投影,此时也称 \hat{L} 为 L 在平面 π 上的投影。很容易得到,在上一节线段距离的定义下, \hat{L} 是在平面 π 上所有线段中与 L 距离最小的线段。

容易证明,任意一个点 P 在平面 π 上的投影点为 $(I - NN^T)P$, 从而得到这个点与其投影点的距离为 $N^T P$ 。因此,在上一节的距离定义下,参考(9)式,原位姿估计问题可以转化为最小化下面的目标函数:

$$J(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^n d^2(L_i, \hat{L}_i) = \frac{1}{6} \sum_{i=1}^n \left\{ [N_i^T (RP_{si} + \mathbf{t})]^2 + 4[N_i^T (RP_{mi} + \mathbf{t})]^2 + [N_i^T (RP_{ei} + \mathbf{t})]^2 \right\}, \quad (11)$$

式中 P_{si} 、 P_{ei} 表示第 i 条 3D 模型线段的两个端点, $P_{mi} = (P_{si} + P_{ei})/2$ 为中点,且共有 n 条模型线段。

显然,目标函数(11)式是 \mathbf{t} 的二次函数。若 \mathbf{R} 已知,则可以得到最优 \mathbf{t} 为

$$\mathbf{t}(\mathbf{R}) = -\left(\sum_{i=1}^n N_i N_i^T\right)^{-1} \sum_{i=1}^n N_i N_i^T R P_{mi}. \quad (12)$$

注意,此处最优 \mathbf{t} 的表达式只与中点相关,可以参考(10)式,因为在 \mathbf{R} 确定的情况下,夹角是固定的,且线段的长度也是不变的,所以线段之间的距离就只与中点距离相关。

将(12)式代入目标函数(11)式,可以得到目标函数实质是旋转矩阵 9 个元素的二次函数。这样,目标函数(11)式转化为

$$J(\mathbf{R}) = \mathbf{r}^T \mathbf{C}_{9 \times 9} \mathbf{r}, \quad (13)$$

式中 $\mathbf{r} = \text{vec}(\mathbf{R})$ 为将 \mathbf{R} 矩阵按列堆栈成一个列向量。 \mathbf{C} 矩阵的具体计算可以参考文献[6,22]对 PnP 问题的处理方法。

2.4 修正目标函数

旋转矩阵 \mathbf{R} 用其 CGR 参数 $\mathbf{s} = [s_1 \quad s_2 \quad s_3]^T$ 表达为

$$\mathbf{R} = \frac{\mathbf{R}}{1 + \mathbf{s}^T \mathbf{s}}, \bar{\mathbf{R}} = (1 - \mathbf{s}^T \mathbf{s})\mathbf{I}_3 + 2[\mathbf{s}]_x + 2\mathbf{s}\mathbf{s}^T, \quad (14)$$

式中 \mathbf{I}_3 为单位矩阵, $[\mathbf{s}]_x$ 为反对称矩阵。将(14)式代入目标函数(13)式,得到

$$J(\mathbf{s}) = \frac{\bar{\mathbf{r}}^T \mathbf{C} \bar{\mathbf{r}}}{(1 + \mathbf{s}^T \mathbf{s})^2}, \quad (15)$$

式中 $\bar{\mathbf{r}} = \text{vec}(\bar{\mathbf{R}})$ 。设其分子部分为修正的目标函数

$$J'(\mathbf{s}) = \bar{\mathbf{r}}^T \mathbf{C} \bar{\mathbf{r}}. \quad (16)$$

从(14)式可以看出 $\bar{\mathbf{R}}_x$ 的每一个元素均为 \mathbf{s} 的二次多项式。修正的目标函数 J' 为 s_1, s_2, s_3 的四次多项式,且阶次与线段数目无关。为了对其进行最小化,只需要取其导数为零,也就是最优条件为

$$\frac{\partial J'}{\partial s_1} = 0, \frac{\partial J'}{\partial s_2} = 0, \frac{\partial J'}{\partial s_3} = 0. \quad (17)$$

由于 J' 的最高阶次为四次,因此,最优条件(17)式的最高阶次为 3 次。也就是原问题转化为求解三元三次方程组的问题。一旦获得了此方程组的根,也就得到了最小化修正目标函数的全局最优解,代入(14)式获得旋转矩阵的解,再根据(12)式得到平移向量的解。

此处,之所以选择修正的目标函数(16)式,而不是原目标函数(15)式,因为最小化(15)式的最优条件阶次为 7 次,会使得后续的计算复杂度很大,同时,根据文献[4, 12]以及后续实验,最小化修正目标函数求解的相机位姿精度也很高。

3 多项式方程组求解

用结式方法求得多项式方程组的解,具体可以参考文献[4,12,23,24]。

3.1 构造麦考利矩阵

定义 n 变量单项式为 $x^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n}$, $\gamma_i \in Z_{\geq 0}$, 和一个包括 n 个变量的多项式为 $f = \sum_i c_i x^\gamma$, $c_i \in C$ 。单

项式的阶次为 $\sum_{i=1}^n \gamma_i$, 而多项式的阶次则是其包含的单项式中的最高阶次。

假设一个方程组有 n 个多项式, 表示为 $f_i = 0, i = 1, \dots, n$ 。其中对应每个方程的阶次为 d_i 。再定义一个辅助多项式: u -多项式, 为 $f_0 = u_0 + u_1 x_1 + \dots + u_n x_n$, 其中 u_i 为任意随机数。那么, 一般情况下, 方程组的解会使得 $f_0 \neq 0$ 。

定义包含辅助多项式的整个方程组的阶次为

$$d = \sum_{i=0}^n d_i - n = 1 + \sum_{i=1}^n d_i - n. \quad (18)$$

定义包含所有阶次等于或低于 d 阶的单项式集合为 $S = \{x^\gamma: \sum \gamma_i \leq d\}$ 。可以得到 S 的元素个数为 C_{n+d}^n 。

为了更好的描述, 举一个 $n = 2$ 的例子:

$$f_1 = x_1 + 2x_2 + 5, \quad f_2 = x_1^2 + 2x_2^2 - 10. \quad (19)$$

在这个例子中, $d_1 = 1, d_2 = 2$, 加入辅助多项式 $f_0 = u_0 + u_1 x_1 + u_2 x_2$ 后的总的阶次为 $d = 2$ 。阶次小于或等于 2 的单项式集合为 $S = \{1, x_1, x_2, x_1 x_2, x_1^2, x_2^2\}$ 。

将 S 分解为 $n+1$ 个子集:

$$\begin{cases} S_n = \{x^\gamma: x^\gamma \in S; x_n^d | x^\gamma\} \\ S_{n-1} = \{x^\gamma: x^\gamma \in S, \notin S_n; x_n^{d-1} | x^\gamma\} \\ \vdots \\ S_0 = \{x^\gamma: x^\gamma \in S, \notin S_n, \dots \notin S_1\} \end{cases}, \quad (20)$$

式中 $|$ 表示整除。针对上面的例子, 则 $S_2 = \{x_2^2\}, S_1 = \{x_1^2, x_1 x_2, x_1\}, S_0 = \{1, x_2\}$ 。

在对集合的分解中, $|S_0| = d_1 d_2 \dots d_n$, 其中 $| \cdot |$ 表示取集合的基数, 可以认为是集合元素的个数。 $x_i^{d_i}, i = 1, \dots, n$ 不能整除 $x^\gamma \in S_0$, 因此, 在 $x^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \dots x_n^{\gamma_n}$ 中的每一个 γ_i 都必须满足 $0 \leq \gamma_i < d_i$ 。所以, 在这种条件下, 会有 $d_1 d_2 \dots d_n$ 种可能性, 也就是 S_0 中不同单项式的个数。另外, 当 $|S_0| > n$, 集合 S_0 中将包含 1 和所有的一阶单项式 x_1, x_2, \dots, x_n , 后面将利用特征向量求解方程组。

基于 S_0, \dots, S_n , 定义单项式集合:

$$S'_i = \left\{ \frac{x^\gamma}{x_i^{d_i}}: x^\gamma \in S_i \right\}, \quad i = 1, \dots, n, \quad S'_0 = S_0. \quad (21)$$

用 S'_i 中的每一个单项式乘以多项式 f_i , 得到扩展的多项式集合:

$$\begin{cases} g_{0,j} = x^{y_j} f_0, [j = 1, \dots, |S'_0|], [x^{y_j} \in S'_0] \\ \vdots \\ g_{n,j} = x^{y_j} f_n, [j = 1, \dots, |S'_n|], [x^{y_j} \in S'_n] \end{cases}. \quad (22)$$

从集合构造关系可以看出, 所有的 S_i 之间不相交, 且 $|S_0 \cup \dots \cup S_n| = |S| = C_{n+d}^n, |S'_i| = |S_i|$, 所以有

$$\sum_{i=1}^n |S'_i| = C_{n+d}^n. \quad (23)$$

也就是扩展的多项式将会有 C_{n+d}^n 个。针对例子(19)式, 有 $S'_2 = \{1\}, S'_1 = \{1, x_1, x_2\}, S'_0 = \{1, x_2\}$, 从而得到扩展的多项式集合:

$$\begin{cases} g_{0,1} = u_0 + u_1 x_1 + u_2 x_2 \\ g_{0,2} = u_0 x_2 + u_1 x_1 x_2 + u_2 x_2^2 \\ g_{1,1} = x_1 + 2x_2 + 5 \\ g_{1,2} = x_1^2 + 2x_2 x_1 + 5x_1 \\ g_{1,3} = x_1 x_2 + 2x_2^2 + 5x_2 \\ g_{2,1} = x_1^2 + 2x_2^2 - 10 \end{cases}. \quad (24)$$

因为集合 S_i 中单项式的阶次都小于等于 d , 所以 S'_i 中单项式的阶次都小于等于 $d - d_i$, 这样, 所有的 $g_{i,j}$ 的阶次都小于等于 d 。而集合 S 包含了所有阶次小于等于 d 的单项式, 因此 $g_{i,j}$ 可以写成这些单项式的线性组

合:

$$g_{ij} = x^{\gamma_j} f_i = c_{ij}^T \underline{x}^\gamma, \quad i = 1, \dots, n, \quad j = 1, \dots, |S_i|, \quad (25)$$

式中 c_{ij} 为对应的系数向量, $\underline{x}^\gamma = [x^{\gamma_1} \quad x^{\gamma_2} \quad \dots \quad x^{\gamma_l}]^T$, $x^{\gamma_i} \in S$, $l = |S| = C_{n+d}^n$ 。将所有的 g_{ij} 列写在一起,并用矩阵表示:

$$\begin{bmatrix} g_{0,1} \\ g_{0,2} \\ \vdots \\ g_{1,1} \\ \vdots \end{bmatrix} = \begin{bmatrix} c_{0,1}^T \\ c_{0,2}^T \\ \vdots \\ c_{1,1}^T \\ \vdots \end{bmatrix} \underline{x}^\gamma = \mathbf{M} \underline{x}^\gamma = \mathbf{M} \begin{bmatrix} \underline{x}^\alpha \\ \underline{x}^\beta \end{bmatrix}, \quad (26)$$

式中 \underline{x}^α 表示属于 S_0 中的单项式, \underline{x}^β 表示剩下的其他单项式。 \mathbf{M} 称为麦考利矩阵,其是具有 $|S| = C_{n+d}^n$ 阶次的方阵。这个矩阵在多项式代数中具有重要的作用。针对(19)式,则(26)式具体表达为

$$\begin{bmatrix} g_{0,1} \\ g_{0,2} \\ g_{1,1} \\ g_{1,2} \\ g_{1,3} \\ g_{2,1} \end{bmatrix} = \begin{bmatrix} u_0 & u_2 & 0 & 0 & u_1 & 0 \\ 0 & u_0 & 0 & u_1 & 0 & u_2 \\ 5 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 & 5 & 0 \\ 0 & 5 & 0 & 1 & 0 & 2 \\ -10 & 0 & 1 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_1 \\ x_2^2 \end{bmatrix} \begin{matrix} \underline{x}^\alpha \\ \underline{x}^\beta \end{matrix}. \quad (27)$$

3.2 求解方程组

假设存在一个解 $\underline{p} = [p_1 \quad \dots \quad p_n]^T$, 那么其满足 $f_1(\underline{p}) = \dots = f_n(\underline{p}) = 0$ 。则有 $g_{1,1}(\underline{p}) = \dots = g_{n,|S_i|}(\underline{p}) = 0$ 。由于 u 为随机数,所以 $f_0(\underline{p})$ 和 $g_{0,j}(\underline{p})$ 一般不等于 0。用 \underline{p}^γ 表示 \underline{x}^γ 在解 \underline{p} 处的表达。将其代入(26)式得到

$$\begin{bmatrix} g_{0,1}(\underline{p}) \\ \vdots \\ g_{0,|S_0|}(\underline{p}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \underline{p}^\alpha \\ \underline{p}^\beta \end{bmatrix}. \quad (28)$$

根据 g_{ij} 的定义,同时对 \mathbf{M} 矩阵进行分块,可以将(28)式变为

$$\begin{bmatrix} f_0(\underline{p}) \underline{p}^\alpha \\ 0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \underline{p}^\alpha \\ \underline{p}^\beta \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{00} & \mathbf{M}_{01} \\ \mathbf{M}_{10} & \mathbf{M}_{11} \end{bmatrix} \begin{bmatrix} \underline{p}^\alpha \\ \underline{p}^\beta \end{bmatrix}, \quad (29)$$

式中 \mathbf{M}_{00} 为 $|S_0| \times |S_0|$ 的方阵。利用 \mathbf{M} 矩阵的 Schur 补,可以得到

$$f_0(\underline{p}) \underline{p}^\alpha = \tilde{\mathbf{M}} \underline{p}^\alpha, \quad (30)$$

式中 $\tilde{\mathbf{M}} = \mathbf{M}_{00} - \mathbf{M}_{01} \mathbf{M}_{11}^{-1} \mathbf{M}_{10}$, 为 $|S_0| \times |S_0|$ 的方阵。

从(30)式可以看出, \underline{p}^α 为 $\tilde{\mathbf{M}}$ 矩阵的特征向量, $f_0(\underline{p})$ 为对应的特征值。因此 \underline{p}^α 可以通过特征分解再加上一个尺度缩放(因为 S_0 中总存在单项式 1)求解出来,其最多有 $|S_0|$ 个解。

针对例子(19)式, $S_0 = \{1, x_2\}$, $|S_0| = 2$ 。设此时 $\tilde{\mathbf{M}}$ 的两个特征向量为 $\mathbf{v}_1, \mathbf{v}_2$, 则可以的到 x_2 的两个解分别为

$$x_2^{*1} = \frac{\mathbf{v}_1(2)}{\mathbf{v}_1(1)}, x_2^{*2} = \frac{\mathbf{v}_2(2)}{\mathbf{v}_2(1)}, \quad (31)$$

式中 \mathbf{v}_j 表示取 \mathbf{v} 向量的第 j 个元素。一旦 x_2 求得,则例子(19)式很容易得解。

3.3 具体实现

针对原问题(17)式,其为三个三次方程。容易得到 $d = 7$ 。则

$$S = \{s^\gamma = s_1^{\gamma_1} s_2^{\gamma_2} s_3^{\gamma_3} : \gamma_1 + \gamma_2 + \gamma_3 \leq 7\}, \quad (32)$$

可以得到 S 集合中元素的个数为 $C_{3+7}^3 = 120$ 。且 $|S_0| = 27 > n = 3$, 则集合 S_0 中将包含 1 和所有的一阶单项式 s_1, s_2, s_3 。具体的有

$$S_0 = \{1, s_1, s_1^2, s_2, s_1 s_2, s_2^2, s_1 s_2^2, s_2^2 s_1, s_1^2 s_2^2, s_3, s_1 s_3, s_1^2 s_3, s_2 s_3, s_1 s_2 s_3, s_1^2 s_2 s_3, s_2^2 s_3, s_1 s_2^2 s_3, s_1^2 s_2^2 s_3, s_3^2, s_1 s_3^2, s_1^2 s_3^2, s_2 s_3^2, s_1 s_2 s_3^2, s_1^2 s_2 s_3^2, s_2^2 s_3^2, s_1 s_2^2 s_3^2, s_1^2 s_2^2 s_3^2\}, \quad (32)$$

注意到, S_0 中第 2, 4, 10 个元素分别对应着 s_1, s_2, s_3 。

根据(30)式,通过构造 \tilde{M} ,求其特征向量,得到了 S_0 集合的 27 个解,也就得到了最优条件(17)式中未知数 s_1, s_2, s_3 的 27 个解。

实际具体实现中需要注意的有:

1) (26) 式中麦考利矩阵 M 的构造,可以采用参数形式构造,再将具体数字代入得到矩阵的具体值。这样,整个问题只需要构造一次 M 矩阵,以后再使用时,直接代入具体数值即可。

2) 27 个候选解中,虚数解可以直接舍掉。因为 CGR 参数一定是实数。实验中,发现最多只有 4 组解为实数。

3) 可以从候选解中求出使得代价函数(16)式最小的全局最优解。

4) 当 u -多项式 f_0 在最优解处接近零时,则求解时数值不稳定,此时,只需要再重新随机生成一个 u -多项式即可。

5) 当旋转矩阵的旋转角接近 180° 的时候, CGR 参数将会非常大,导致求解的数值不稳定,此时,只需要将原旋转矩阵随机旋转为另一个新的旋转矩阵,当新的旋转矩阵求解结束时再旋转回来即可。

4 实 验

4.1 数字仿真实验

默认的仿真条件为:相机的等效焦距为 1024,半视场角为 25° , 2D 直线的两个端点在图像上随机生成,对应的 3D 直线的端点的深度为 [10,20] 区间均匀随机数。由于直线的端点沿直线方向定位误差较大,因此,参考文献[15]中直线段噪声的添加方法,沿直线方向,加入 [0,20%] 均匀分布的长度比例噪声,沿直线垂直方向,两个端点分别加入独立的方差为 1 pixel 的高斯噪声。相机的姿态旋转矩阵 R 随机生成,平移 t 的 3 个分量分别按照 $N(0,100)$ 随机生成。进行位姿估计的线段对数为 10。每一种配置仿真 1000 次,统计旋转误差及平移误差的平均值。旋转矩阵的误差用两个旋转矩阵之间的夹角表示,平移向量的误差用距离表示:

$$E_R = \cos^{-1} \left[\frac{\text{tr}(R^T R_g) - 1}{2} \right], \quad (34)$$

$$E_t = \|t - t_g\|, \quad (35)$$

式中 R_g, t_g 为对应的真值。

在计算精度和计算时间上,比较下列六种算法:

- 1) RPNL, 文献[10]提出的利用 P3L 的组合,将旋转的求解转化为求解一个 15 次方程的稳健高效线性算法。
- 2) Ansar, 文献[11]提出的用新参数表达使得多项式方程组变为线性方程组的线性算法,其计算复杂度为 $O(n^2)$ 。
- 3) Mirzaei, 文献[12]提出的使用直接最小二乘法先求旋转再求平移的线性算法。
- 4) RpnL+Kumar, 以 RPNL 为初值的、文献[15]中的 R_and_T 迭代算法。
- 5) RPNL+LOI, 以 RPNL 为初值的、文献[20]中的直线对应正交迭代算法。
- 6) DLSPnL, 本文提出的直接最小二乘法。

图 5、6 分别给出了噪声水平从 0.5 pixel 逐渐增加到 5 pixel、直线段数目从 4 增加到 20 得到的相机位姿的误差统计结果。可以看出, DLSPnL 算法精度不仅是线性算法中精度最高的,还略高于两种迭代方法。

图 7 给出了在目前常用配置 PC 机上使用 Matlab 软件实现,获得的六种算法随直线段对数增加所需要的计算时间,其中图 7(a)是较少直线段数的情况,图 7(b)为大量数目的情况。注意, Ansar 算法的复杂度为 $O(n^2)$, 因此图 7(b)并没有画出,再者,两种迭代算法的时间需要加上求解初值的时间。可以看出,在直线段数较少时,本文算法计算时间并不占优势,但仍然很快。本文算法与 Mirzaei 算法都需使用代数多项式方法,因此

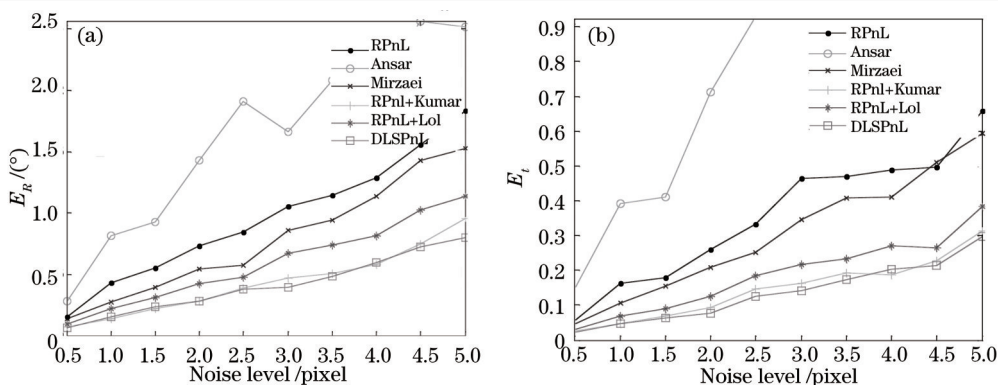


图 5 六种算法随着噪声增加的平均位姿误差

Fig.5 Mean pose errors with respect to varying noise levels of six algorithms

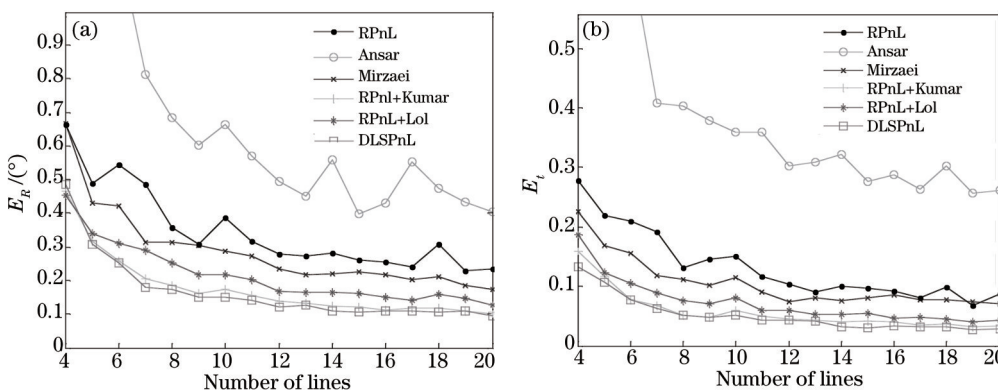


图 6 六种算法随着线数增加的平均位姿误差

Fig.6 Mean pose errors with respect to varying number of lines of six algorithms

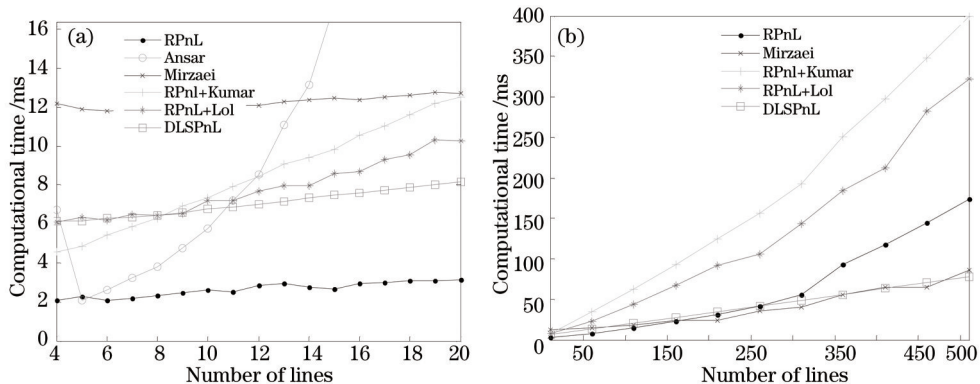


图 7 六种算法随着线数增加的平均计算时间

Fig.7 Mean computational time with respect to varying number of lines of six algorithms

两者计算时间接近。由于求解多项式方程组的时间可以认为是固定的,而系数矩阵的构造随着直线段数的增加所增加的计算量是缓慢的,因此,在直线段数较大时,本文算法相对于除 Mirzaei 算法外的其他 4 种算法,在计算速度上具有明显的优势。

4.2 真实图像实验

为了验证提出的直接最小二乘法的有效性,分别对一组仿真生成的卫星图像序列和一组真实卫星模型图像序列进行处理。直线特征采用文献[25]的方法检测。首帧手动确定 2D 图像线段与 3D 模型线段之间的对应关系,之后则采用最近邻方法确定对应关系。真实图像相机的内参提前使用 Zhang^[26]的方法标定。利用直接最小二乘法对相机进行位姿估计,并针对仿真生成的图像,将模型重投影到图像上,针对真实图像,将可见的模型线段重投影到图像上。如图 8 所示,红线表示重投影线,第一行为 4 幅仿真生成的图像重投影模型的结果,第二行为 4 幅真实图像重投影结果,可以看出,重投影结果与图像中的模型符合的很好,表明本算法是有效的。

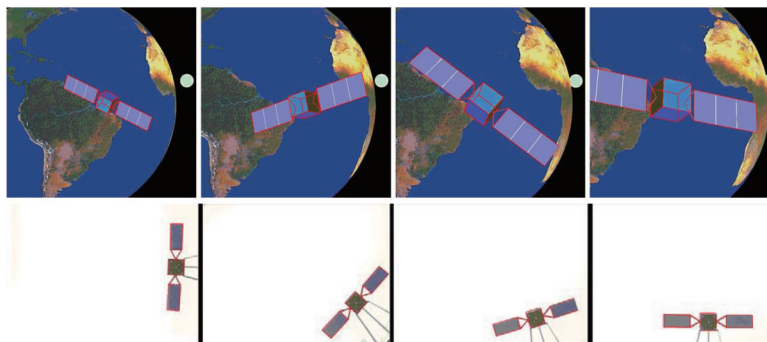


图 8 真实图像实验结果

Fig.8 Experimental results in real images

5 结 论

为了高精度的利用直线段对应进行相机位姿估计,提出一种直接最小二乘法。提出了一种直线段距离测度,该距离测度综合考虑了线段的端点距离、中点距离、夹角和长度,利用此距离测度将原问题转化为最小化一个姿态旋转矩阵的二次目标函数,通过旋转矩阵的CGR参数表达,获得一个修正的目标函数,此修正目标函数的最优解条件组成了一个三元三次方程组,利用代数多项式方法直接求解这个方程组,从而得到了旋转矩阵的全局最优解。仿真和真实实验验证了该方法的有效性和高精度。

后续的工作有:1)将直线段之间的距离测度推广到一般曲线之间的距离的情况;2)利用代数方法求解位姿的方法可以推广到别的需要求解位姿的情况。

参 考 文 献

- 1 R Hartley, A Zisserman. Multiple View Geometry in Computer Vision[M]. England: Cambridge University Press, Second Edition, 2004.
- 2 S Li, C Xu, M Xie. A robust $O(n)$ solution to the perspective- n -point problem[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1444-1450.
- 3 Y Zheng, Y Kuang, S Sugimoto, *et al.*. Revisiting the PnP problem: A fast, general and optimal solution[C]. ICCV, 2013: 2344-2351.
- 4 J A Hesch, S I Roumeliotis. A direct least-squares (DLS) method for PnP[C]. ICCV, 2011: 383-390.
- 5 C P Lu, G D Hager, E Mjolsness. Fast and globally convergent pose estimation from video images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(6): 610-622.
- 6 G Schweighofer, A Pinz. Globally optimal $O(n)$ solution to the PnP problem for general camera models[C]. BMVC, 2008: 1-10.
- 7 M Dhome, M Richetin, J T Lapreste, *et al.*. Determination of the attitude of 3D objects from a single perspective view[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11(12): 1265-1278.
- 8 H H Chen. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions[C]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(6): 530-541.
- 9 L J Qin, F Zhu. A new method for pose estimation from line correspondences[J]. Acta Automatica Sinica, 2008, 34(2): 130-134.
- 10 L Zhang, C Xu, K M Lee, *et al.*. Robust and efficient pose estimation from line correspondences[C]. ACCV 2012: 217-230.
- 11 A Ansar, K Daniilidis. Linear pose estimation from points or lines[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(5): 578-589.
- 12 F M Mirzaei, S I Roumeliotis. Globally optimal pose estimation from line correspondences[C]. IEEE International Conference on Robotics and Automation (ICRA), 2011, 5581-5588.
- 13 J Liu, X Zhang, H Liu, *et al.*. New method for camera pose estimation based on line correspondence[J]. Science China Technological Sciences, 2013, 56(11): 2787-2797.
- 14 Zhang Yueqiang, Su Ang, Zhu Zunshang, *et al.*. Three dimensional rigid objects pose tracking and optimization based on multilevel line representation and M-eEstimation[J]. Acta Optica Sinica, 2015, 35(1): 0115003.
张跃强, 苏 昂, 朱遵尚, 等. 基于多级直线表述和E-估计的位姿跟踪优化算法[J]. 光学学报, 2015, 35(1): 0115003.
- 15 R Kumar, A R Hanson. Robust methods for estimating pose and a sensitivity analysis[J]. CVGIP: Image Understanding, 1994, 60(3): 313-342.

- 16 C J Taylor, D J Kriegman. Structure and motion from line segments in multiple images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(11): 1021-1032.
- 17 S Christy, R Horaud. Iterative pose computation from line correspondences[J]. Computer Vision and Image Understanding, 1999, 73(1): 137-144.
- 18 P Besl, N D Mckay. A method for registration 3-D shapes[J]. IEEE Transactions on. Pattern Analysis and Machine Intelligence, 1992, 14(2): 232-256.
- 19 P David, D Dementhon, R Duraiswami, *et al.*. Simultaneous pose and correspondence determination using line features[C]. CVPR, 2003: 424-431.
- 20 X Zhang, Z Zhang, Y Li, *et al.*. Robust camera pose estimation from unknown or known line correspondences[J]. Appl Opt, 2012, 51(7): 936-948.
- 21 Xu Yunxi, Jiang Yunliang, Chen Fang. An extended orthogonal iteration algorithm for pose estimation from point and line segment correspondences[J]. Acta Optica Sinica, 2009, 29(11): 3129-3135.
许允喜, 蒋云良, 陈 方. 基于点和直线段对应的扩展正交迭代位姿估计算法[J]. 光学学报, 2009, 29(11): 3129-3135.
- 22 Li Xin, Long Gucan, Liu Jinbo, *et al.*. Accelerative orthogonal iteration algorithm for camera pose estimation[J]. Acta Optica Sinica, 2015, 35(1): 0115004.
李 鑫, 龙古灿, 刘进博, 等. 相机位姿估计的加速正交迭代算法[J]. 光学学报, 2015, 35(1): 0115004.
- 23 D A Cox, J B Little, D O'shea. Using Algebraic Geometry[M]. Berlin: Springer, 2004: 77-128.
- 24 Wang Mingdong, Mou Chenqi, Li Xiaoliang, *et al.*. Polynomial Algebra[M]. Beijing: Higher Education Press, 2011. 107-135.
王明东, 牟晨琪, 李晓亮 等. 多项式代数[M]. 北京: 高等教育出版社, 2011: 107-135.
- 25 Von Gioi Rafael G, J Jérémie, J M Morel, *et al.*. LSD: A Fast Line Segment Detector with a False Detection Control[J]. IEEE Transactions. Pattern Analysis and Machine Intelligence, 2010, 4(32): 722-732.
- 26 Z Y Zhang. A flexible new technique for camera calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11): 1330-1334.

栏目编辑: 苏 岑