

基于质点系运动原理的像机位姿估计算法

陈 鹏¹ 胡广大¹ 闫桂荣²

(¹ 北京科技大学自动化学院, 北京 100083
² 美国德州大学艾尔帕索分校土木工程系, 美国 艾尔帕索 TX 79968)

摘要 大多数基于单幅二维(2D)图像的像机位姿估计算法都是在已知全部或者部分三维(3D)/2D 特征点对应关系的基础上设计的,而对于 3D/2D 特征点对应关系完全未知的情况则很少涉及。利用经典力学中的质点系运动原理设计了一种像机位姿估计算法,该算法能够在未知特征点对应关系的情况下,同时确定 3D/2D 特征点对应关系与像机位姿。此外,利用空间共线性误差和匹配矩阵,所提出的算法不仅能够处理 3D/2D 特征点一一对应的情况,而且能够处理部分 3D 特征点被遮挡,以及图像中存在 2D 伪特征点的情况。通过实验,并与其他相关算法进行比较,结果表明所提出的算法能够在不增加计算复杂度的条件下有效地匹配特征点并估计像机位姿,而且受图像噪声和 2D 伪特征点的影响较小。

关键词 测量;位姿估计;质点系;无对应

中图分类号 TP391.41 **文献标识码** A **doi**: 10.3788/AOS201434.0112002

Camera Pose Estimation Algorithm Based on Particle System Kinematics

Chen Peng¹ Hu Guangda¹ Yan Guirong²

(¹ School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
² Department of Civil Engineering, University of Texas at El Paso, El Paso, TX 79968, USA)

Abstract Most pose estimation algorithms using a single two-dimensional (2D) image are designed based on totally or partially knowing the correspondence between three-dimensional (3D) and 2D feature points, but few have involved in a correspondenceless case. A camera pose estimation algorithm is proposed by using particle system kinematics in classical mechanics. It can simultaneously decide the correspondence and the camera pose. Besides, by introducing object space collinear error and matching matrix, the proposed algorithm can be used not only in the case when the correspondences of 3D/2D feature points are one to one, but also in the case when the 3D feature points are partially occluded and when there are 2D false feature points in an image. Through experiments and comparison with other algorithms, the result shows that the proposed algorithm can effectively find the correct correspondence and estimate camera pose without increasing computational complexity, and the impact of image noises and 2D false feature points on the algorithm is little.

Key words measurement; pose estimation; particle system; correspondenceless

OCIS codes 120.0280; 110.3925; 100.4995; 150.1135

1 引 言

像机位姿估计问题是机器视觉领域的一个经典问题,在机器人手眼定标、虚拟现实、物体识别乃至空间站交会对接等应用中均有涉及。

在像机内参数已知的情况下,位姿估计问题又被抽象为透视 n 点投影问题,简称 PnP 问题。求解 PnP 问题的方法主要有三类:解析算法^[1-3],迭代算法^[4-7]和全局优化算法^[8-10]。这些算法使用的前提

收稿日期: 2013-06-18; **收到修改稿日期**: 2013-07-18

作者简介: 陈 鹏(1983—),男,博士研究生,主要从事机器视觉、图像处理等方面的研究。E-mail: holmes83@163.com

导师简介: 胡广大(1962—),男,博士,教授,主要从事飞行器导航控制与微分方程数值方法方面的研究。

E-mail: ghu@ies.ustb.edu.cn

本文电子版彩色效果请详见中国光学期刊网 www.opticsjournal.net

都是需要已知三维(3D)物体特征点与图像中二维(2D)特征点的对应关系。于是,在实际应用中为了尽可能简单高效地获得这种对应关系,常常要将物体特征点设计为不同的形状或颜色。这样做的缺点是:特征点的数量越多,为每个特征点分配不同的形状或者颜色的问题就越复杂;当像机的视角或者光照发生变化时,所成图像中特征点的形状与颜色也会发生变化以致无法有效地与其他特征点区分;物体运动过程中出现特征点被遮挡的问题或者图像特征提取过程中提取到伪特征点的问题往往在实际应用中无法避免。所以,研究如何在未知 3D/2D 特征点对应关系的情况下匹配特征点并确定像机位姿十分必要。

目前应用最多的方法是假设检验法,即假设一小部分特征点的匹配关系是正确的,利用这部分特征点的对应关系求解像机位姿,将解出的位姿应用于所有特征点,如果这些特征点在图像上的投影与 2D 特征点间的距离足够小,就认为当前的位姿是正确的,否则就产生新的假设并重复上述过程。这类方法的典型代表是 RANSAC 算法^[11],然而在文献[12]中已经说明当 3D/2D 特征点的对应关系完全未知时,RANSAC 算法并不高效。

SoftPOSIT 算法^[12]是一种同时确定 3D/2D 特征点对应关系与像机位姿的算法。这种方法将 SoftAssign 方法^[13]与 POSIT 算法^[14]相结合,并在算法执行过程中用确定性退火算法来维护一个匹配矩阵。为了避免陷入局部极小,算法需要从多个初始姿态开始执行。基于力学原理同时确定特征点匹配与像机位姿的方法^[15]的优点是无论何种初始条件出发,当执行次数足够多时,均能够在 3D/2D 特征点一一对应或者仅存在 3D 特征点被遮挡的情况下找到正确的位姿,相比于 SoftPOSIT 算法具有较强的稳健性,因此,它可以用来初始化 SoftPOSIT 算法^[16]。

本文在文献[15-16]的基础上,将空间共线性误差^[17]和匹配矩阵^[12]的概念引入到基于力学原理的特征点对应关系未知的像机位姿估计算法中来,利用质心运动定理与质点系相对于质心的动量矩定理^[18]改进了原算法,使得新算法不仅能够应用于 3D/2D 特征点一一对应的情况,而且能够应用于 3D 特征点存在遮挡,以及图像特征提取后出现 2D 伪特征点的情况。

2 算法描述

假设物体由 L 个特征点组成,并且第 k 个特征

点 P_k^w 在世界坐标系中的坐标为 $P_k^w = [X_k^w, Y_k^w, Z_k^w]^T$ 。根据透视投影模型, P_k^w 将被投影为图像坐标下的点 $p_i, p_i = [x_i, y_i]^T, p_i$ 和 P_k^w 的关系如下:

$$\tilde{p}_i = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} [R \quad t] \tilde{P}_k^w, \quad (1)$$

式中 \tilde{p}_i 和 \tilde{P}_k^w 分别为 p_i 和 P_k^w 的齐次坐标^[19], f 为镜头焦距,矩阵 R 为旋转矩阵,满足 $R^T R = I$,如果将 R 表示为 $R = [R_1, R_2, R_3]$,则 R_1, R_2, R_3 分别表示世界坐标系的 X^w, Y^w, Z^w 轴的方向向量在像机坐标系中的坐标。向量 t 为平移向量,同时也是世界坐标系的原点在像机坐标系中的坐标。

图 1 是透视投影模型的示意图。根据(1)式, P_k^w 应位于光心 O_c 与图像点 p_i 构成的视线 $O_c p_i$ 上。然而,由于噪声干扰等原因, p_i 点会发生偏移,形成点 $p_i^n, p_i^n = [x_i^n, y_i^n]^T$ 。于是, P_k^w 与视线 $O_c p_i^n$ 间的距离可以表示为^[17]

$$d_{ik} = \| (I - V_i)(R P_k^w + t) \|^2. \quad (2)$$

$P_k^i = R P_k^w + t$, 是 P_k^w 在像机坐标系下的坐标。 V_i 是视线 $O_c p_i^n$ 的投影矩阵,可以由下式求得:

$$V_i = \frac{v_i v_i^T}{v_i^T v_i}, \quad v_i = [x_i, y_i, f]^T. \quad (3)$$

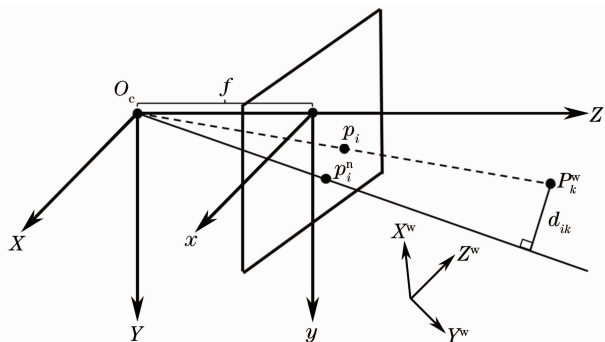


图 1 透视投影模型

Fig. 1 Perspective projection model

假设图像点共有 N 个,并且 N 可以不等于 L 。用 $m_{ik} = 1$ 表示 P_k^w 与 p_i^n 相匹配,用 $m_{ik} = 0$ 表示它们不匹配,那么,当 3D/2D 特征点对应关系与像机位姿都正确时,如下的能量函数将达到最小:

$$E = \sum_{i=1}^N \sum_{k=1}^L m_{ik} \| (I - V_i)(R P_k^w + t) \|^2. \quad (4)$$

当给定 3D 特征点集合 $P^w = \{P_k^w, k = 1, 2, \dots, L\}$ 和 2D 特征点集合 $p^n = \{p_i^n, i = 1, 2, \dots, N\}$ 后,根据(2)式就可以计算出物体点到各条视线的距离,得到距离矩阵 $D = [d_{ik}]_{N \times L}$ 。由矩阵 D 按照如下算法就能够获得由 m_{ik} 构成的匹配矩阵 $M = [m_{ik}]_{N \times L}$ 。

算法 1 此算法在给定矩阵 \mathbf{D} 后返回匹配矩阵 \mathbf{M} , 步骤如下。

- 1) 令 $\mathbf{M}=[0]_{N \times L}$, $\mathbf{D}_m=\mathbf{D}$ 。
- 2) 确定 (i, k) 使得 $d_{ik}=\min\{d|d \in \text{cs}(\mathbf{D}_m)\}$, 其中 $\text{cs}(\mathbf{D}_m)$ 表示将矩阵 \mathbf{D}_m 进行列展开。
- 3) 令 $m_{ik}=1$, 同时将 \mathbf{D}_m 的第 i 行和第 k 列置为 C , C 是一个非常大的常数。
- 4) 如果 \mathbf{D}_m 中的所有元素不都等于 C , 则转到步骤 2), 否则终止算法, 并返回匹配矩阵 \mathbf{M} 。

算法 1 中得出的匹配矩阵 \mathbf{M} 具有这样的特点: 当 3D 特征点存在遮挡时, \mathbf{M} 中对应的列全部为 0, 而当存在 2D 伪特征点时, 对应的行全部为零, 因此算法 1 能够突破文献[15-16]中“图像中不存在 2D 伪特征点”的限制, 从而可以应用于 3D/2D 特征点一一对应、3D 特征点存在遮挡以及图像中存在 2D 伪特征点的情况。需要注意的是算法 1 还不能处理 3D 特征点被遮挡与图像中 2D 伪特征点同时出现的情况, 所以, 在后面的论述中均假设不出现这种情况。

实际上, 由匹配矩阵 \mathbf{M} 和距离矩阵 \mathbf{D} 可以将(4)式所示的能量函数简单地写成两个矩阵列展开的点积形式, 即

$$E = \text{cs}(\mathbf{M}) \cdot \text{cs}(\mathbf{D}). \quad (5)$$

在未知 3D/2D 特征点对应的条件下, 同时确定 3D/2D 特征点匹配与像机位姿的过程可以描述为通过更新矩阵 \mathbf{M} 和 \mathbf{D} 使能量 E 达到最小的过程。根据(2)、(4)式可知要使能量 E 最小, 则应使距离 d_{ik} 最小, 即要使 P_k^c 向视线 $O_c p_i^n$ 靠近。假设视线 $O_c p_i^n$ 对点 P_k^c 存在引力作用, 在该力的作用下 P_k^c 向视线 $O_c p_i^n$ 运动, 随着距离 d_{ik} 的减小, 该引力作用也逐渐减小, 使得 P_k^c 最终停留在 d_{ik} 最小的位置。对于所有 L 个特征点, 可以将它们看作由质量为 1 的质点构成的质点系, 在引力作用下, 其运动规律服从质心运动定理与质点系相对于质心的动量矩定理。于是, 将未知特征点对应的像机位姿估计算法描述如下。

算法 2 给定 3D 特征点集合 $P^w = \{P_k^w, k = 1, 2, \dots, L\}$ 和 2D 特征点集合 $p^n = \{p_i^n, i = 1, 2, \dots, N\}$, 此算法返回能量 E_{est} , 旋转矩阵 \mathbf{R}_{est} 和平移向量 \mathbf{t}_{est} 。步骤如下。

- 1) 将世界坐标系的原点移动到特征点集的质心 P_c^w , 并得到 3D 特征点在新的世界坐标系下的坐标 $P_k^w, k = 1, 2, \dots, L$ 。
- 2) 给定初始旋转矩阵 \mathbf{R}_0 和平移向量 \mathbf{t}_0 , 计算

$P_k^c = \mathbf{R}P_k^w + \mathbf{t}, k = 1, 2, \dots, L$, 以及距离矩阵 \mathbf{D} 和匹配矩阵 \mathbf{M} 。

- 3) 根据(5)式计算能量 E_0 , 并令 $E_{\text{est}} = E_0$, $\mathbf{R}_{\text{est}} = \mathbf{R}_0, \mathbf{t}_{\text{est}} = \mathbf{t}_0$, 设定参数 ϵ, τ 和算法最大执行次数 η 。

- 4) 每个特征点 P_k^c 上的受力为

$$\mathbf{F}_k = \sum_{i=1}^N m_{ik} (\mathbf{I} - \mathbf{V}_i) \mathbf{P}_k^c. \quad (6)$$

- 5) 由于质点系中各个质点的质量为 1, 所以对于由 L 个质点构成的质点系, 其质量为 L , 根据质心运动定理, 质心的加速度为

$$\mathbf{a}_c = \sum_{k=1}^L \mathbf{F}_k / L. \quad (7)$$

- 6) 引力对质心的主矩为

$$\mathbf{T}_c = \sum_{k=1}^L \mathbf{r}_k \times \mathbf{F}_k, \quad (8)$$

式中 $\mathbf{r}_k = \mathbf{P}_k^c - \mathbf{t}$, 是 \mathbf{P}_k^c 相对于质心的矢径。

- 7) 根据质点系相对于质心的动量矩定理, 质点系的角加速度为

$$\boldsymbol{\alpha} = (\mathbf{J}_c)^{-1} \mathbf{T}_c, \quad (9)$$

式中 \mathbf{J}_c 是质点系相对于质心的转动惯量矩阵, 如果将 \mathbf{r}_k 表示为 $\mathbf{r}_k = [r_{xk}, r_{yk}, r_{zk}]^T$, 则 \mathbf{J}_c 的形式为

$$\mathbf{J}_c = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{yx} & J_{yy} & -J_{yz} \\ -J_{zx} & -J_{zy} & J_{zz} \end{bmatrix}, \quad (10)$$

式中

$$\begin{aligned} J_{xx} &= \sum_{k=1}^L (r_{yk}^2 + r_{zk}^2), & J_{yz} &= J_{zy} = \sum_{k=1}^L r_{yk} r_{zk}, \\ J_{yy} &= \sum_{k=1}^L (r_{xk}^2 + r_{zk}^2), & J_{xz} &= J_{zx} = \sum_{k=1}^L r_{xk} r_{zk}, \\ J_{zz} &= \sum_{k=1}^L (r_{xk}^2 + r_{yk}^2), & J_{xy} &= J_{yx} = \sum_{k=1}^L r_{xk} r_{yk}. \end{aligned} \quad (11)$$

- 8) 由于算法是逐次执行的, 所以相邻两次执行之间位移的改变量可以写成 $\Delta \mathbf{t} = \mathbf{a}_c \cdot \Delta t^2$, 故可令新的 $\mathbf{t} = \mathbf{t} + \mathbf{a}_c$ 。

- 9) 将 \mathbf{R} 表示为 $\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3]$, 并使 $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ 分别绕 $\boldsymbol{\alpha}$ 转过 $|\boldsymbol{\alpha}|$ 弧度, 从而得到新的旋转矩阵 \mathbf{R} 。

- 10) 计算 $P_k^c = \mathbf{R}P_k^w + \mathbf{t}$, 根据(2)式和算法 1 更新距离矩阵 \mathbf{D} 和匹配矩阵 \mathbf{M} , 并计算能量 E 。

- 11) 如果 $E < E_{\text{est}}$, 则令 $E_{\text{est}} = E, \mathbf{R}_{\text{est}} = \mathbf{R}, \mathbf{t}_{\text{est}} = \mathbf{t}$, 否则直接转到步骤 12)。

12) 如果 $E < \epsilon$, 或者算法已经执行了 η 次, 则返回 $E_{\text{est}}, \mathbf{R}_{\text{est}}$, 以及 $\mathbf{t}_{\text{est}} = \mathbf{t}_{\text{est}} - \mathbf{R}\mathbf{P}_c^w$, 并终止算法; 否则计算 $\Delta E = E - E_0$. 如果 $\Delta E > 0$, 则将 \mathbf{R} 取为一个随机旋转矩阵, 转到步骤 4); 否则令 $E_0 = E$, 转到步骤 4); 此外, 为了避免算法陷入局部极小, 当连续 τ 次出现 $-\epsilon < \Delta E < 0$ 时, 将 \mathbf{R} 置为一个随机旋转矩

阵, 并转到步骤 4)。

随机旋转矩阵的产生, 可以遵照这样的过程: 假设世界坐标系的三个坐标轴 X^w, Y^w, Z^w 相对于像机坐标系转过的角度为 ϑ, ψ, φ , 则旋转矩阵 \mathbf{R} 可以表示为

$$\mathbf{R} = \begin{bmatrix} \cos \psi \cos \varphi & \cos \psi \sin \varphi & -\sin \psi \\ \cos \varphi \sin \vartheta \sin \psi - \cos \vartheta \sin \varphi & \cos \vartheta \cos \varphi + \sin \vartheta \sin \psi \sin \varphi & \cos \psi \sin \vartheta \\ \sin \vartheta \sin \psi + \cos \vartheta \cos \varphi \sin \psi & \cos \vartheta \sin \psi \sin \varphi - \cos \varphi \sin \vartheta & \cos \vartheta \sin \psi \end{bmatrix}. \quad (12)$$

如果已知物体发生旋转的取值范围, 则可令 ϑ, ψ, φ 取这一范围内的随机数, 并由 (12) 式产生随机旋转矩阵 \mathbf{R} . 由于在未知特征点对应的像机位姿估计问题中, (4) 式具有许多的局部极小值, 因此合理地确定物体的旋转范围, 并使得该范围尽可能小, 会有助于算法在较少的执行次数下得到正确的结果. 对比文献 [15-16], 算法 2 在能量函数的计算、特征点 P_i^c 的受力计算、角加速度的计算以及随机旋转矩阵的产生等方面做出了改进.

3 实验结果

分别对提出的算法进行了合成图像和真实图像实验, 其中, 合成图像实验通过大量重复性实验来对算法的性能进行统计分析, 而真实图像则用来评价算法在实际应用中的效果.

3.1 合成图像实验

在这部分实验中, 考察图像噪声增加、被遮挡的 3D 特征点数量增加以及图像中 2D 伪特征点增加时对算法性能的影响. 比较关注的算法性能包括: 算法是否能正确找到 3D/2D 特征点对应, 算法在正确匹配了 3D/2D 特征点后的位姿估计误差. 其中, 位置估计误差和旋转估计误差表示为^[10]

$$E_p = \|\mathbf{R}_{\text{est}}^T \mathbf{t}_{\text{est}} - \mathbf{R}_{\text{true}}^T \mathbf{t}_{\text{true}}\|, \quad (13)$$

$$E_r = 2 \arccos(0.5 \sqrt{1 + E_{11} + E_{22} + E_{33}}), \quad (14)$$

$$\mathbf{E} = \mathbf{R}_{\text{est}} \mathbf{R}_{\text{true}}^T.$$

在一台配有 Intel®i3 处理器, 主频为 2.5 GHz, 内存为 4 GB 的计算机上创建了一台焦距为 35 mm 的虚拟像机, 成像尺寸为 1200 pixel × 1600 pixel, 主点位于图像的中心, 单位像元的大小为 12 μm × 12 μm. 每次实验中令 3D 特征点随机分布在像机坐标系中 $[-2 \text{ m}, 2 \text{ m}] \times [-2 \text{ m}, 2 \text{ m}] \times [4 \text{ m}, 9 \text{ m}]$ 的区域内, 之后使三个旋转角在 $[0^\circ, 45^\circ]$ 内随机产生. 另外, 设定每次实验中算法的最高执行次数为 $\eta = 20000$, 参数 $\epsilon = 10^{-5}$, $\tau = 20$, 随机旋转矩阵的三个旋转角均产生于 $[-90^\circ, 90^\circ]$ 内, 初始平移向量产生于像机坐标系中 $[-5 \text{ m}, 5 \text{ m}] \times [-5 \text{ m}, 5 \text{ m}] \times [0, 10 \text{ m}]$ 的区域内. 针对 $L = 8, 10, 12, 15$ 的情况分别进行 100 次实验, 对得到的结果进行统计和分析.

图 2 为假设图像噪声服从零均值高斯分布, 取标准差 $\sigma = 0.5, 5, 10$ pixel 时得到的图像噪声对算法性能的影响结果. 从旋转估计误差和位置估计误差能够看出实验结果和预期的结果是相吻合的: 当 σ 变大时, 估计误差随之增大; 当 L 增加时, 估计误差随之减小. 从 3D/2D 特征点的匹配成功率来看,

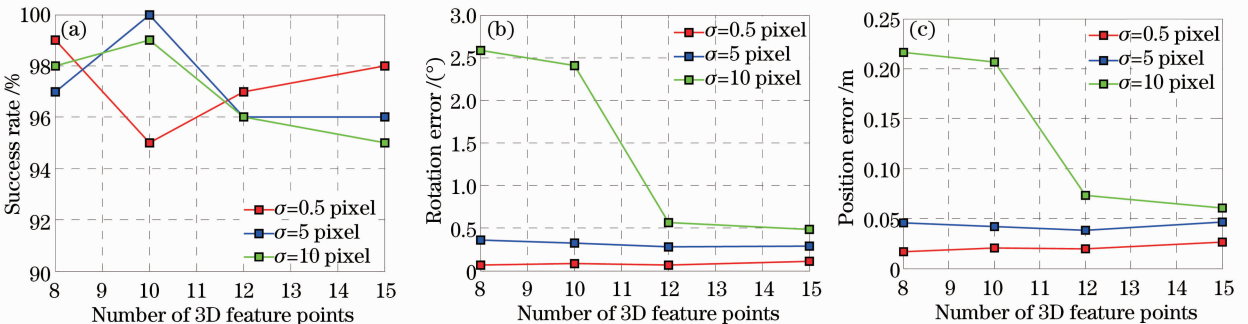


图 2 图像噪声对算法性能的影响. (a) 匹配成功率; (b) 旋转误差; (c) 位置误差

Fig. 2 Impact of image noise on the algorithm. (a) Matching success rate; (b) rotation error; (c) position error

随着 σ 的增加, 匹配成功率都在 95% 以上, 并没有出现明显的下降, 这说明图像噪声对算法匹配结果的影响较小。由于当 3D/2D 特征点匹配成功以后, 可以采用基于特征点对应的位姿估计算法对相机位姿进行更精确的估计, 所以较高的匹配成功率更具有实际意义。

图 3 为 $\sigma=0.5$ pixel, 将 3D 特征点被遮挡数量 a 分别设置为 1、2、3 时的实验结果。图中当被遮挡点的数量小于 3 时算法仍然保持了接近 90% 的匹配成功率, 而当被遮挡点的数量达到 3 以后, 如果 L 较小, 例如 $L=8$ 时, 匹配成功率会出现明显的下降。这是由于被遮挡点的引入使得 3D/2D 特征点

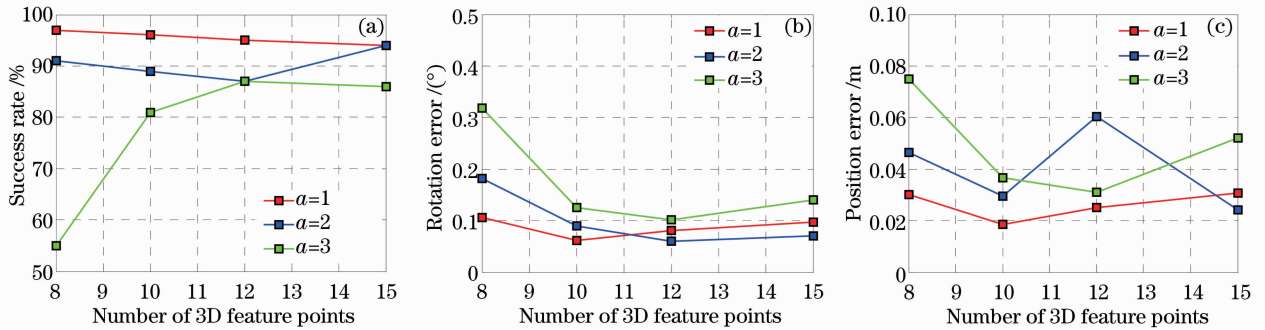


图 3 3D 特征点被遮挡对算法的影响。(a) 匹配成功率; (b) 旋转误差; (c) 位置误差

Fig. 3 Impact of occluded 3D feature points on the algorithm. (a) Matching success rate; (b) rotation error; (c) position error

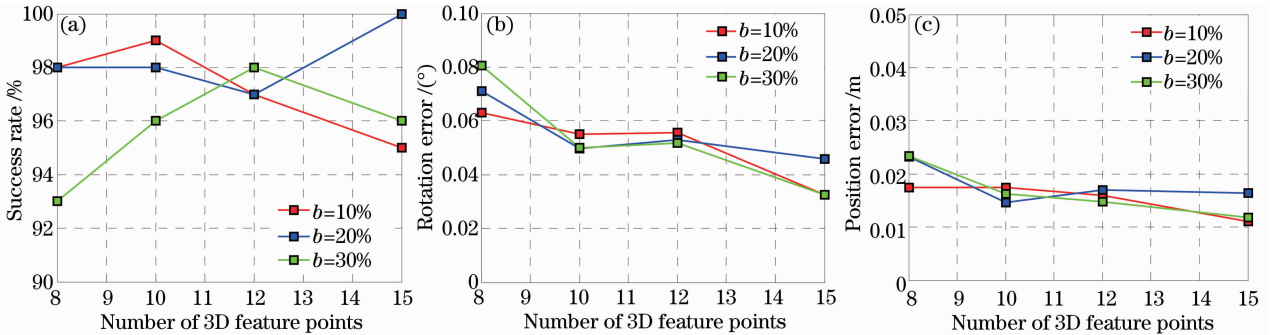


图 4 2D 伪特征点对算法的影响。(a) 匹配成功率; (b) 旋转误差; (c) 位置误差

Fig. 4 Impact of 2D false feature points on the algorithm. (a) Matching success rate; (b) rotation error; (c) position error

图 4 显示, 通过引入匹配矩阵 \mathbf{M} , 本文所提出的算法能够很好地处理图像中存在 2D 伪特征点的情况, 而这时文献[15-16]中的算法已经无法使用。为了与文献[15-16]进行更全面的对比, 将它们中的算法标记为 GPE, 将本文提出的算法标记为 PSKPE, 令 $\sigma=0.5$ pixel, 3D 特征点数量 $L=10$, 限制算法最大执行次数分别为 $\eta=500, 1000, 2000, 5000, 10000$, 不存在遮挡与伪特征点, 其他参数配置与之前的实验相同。图 5 是对 PSKPE 与 GPE 的

的匹配组合关系变得更加复杂, 算法的搜索空间变大, 从而无法在 η 次执行后得出正确的结果。当匹配成功以后, 旋转估计误差不超过 0.5° 、位置估计误差不超过 0.08 m, 与图 2 所示的结果相符。

图 4 为 $\sigma=0.5$ pixel, 将 2D 伪特征点的数量比 b (占 2D 特征点数量 N 的百分比) 分别设置为 10%、20%、30% 时的实验结果。从图中可以看出算法保持了 92% 以上的匹配成功率, 也说明伪特征点对于算法匹配成功率的影响较小。而旋转估计误差小于 0.08° , 位置估计误差小于 0.03 m 的结果说明当图像噪声很小, 或者特征点提取比较精确时, 算法能够很好地估计相机的位姿。

匹配成功率、旋转误差和位置误差进行对比的结果。从图中能够看出, 当算法的执行次数较少时, PSKPE 能够以更大的概率找到正确的特征点匹配。这主要是由于两种算法为了避免陷入局部极小所采用的随机旋转矩阵产生策略有所不同: 在 GPE 算法中先是随机产生角加速度 α , 然后令当前的 \mathbf{R}_1 、 \mathbf{R}_2 和 \mathbf{R}_3 分别绕 α 转过 $|\alpha|$ 弧度得到随机旋转矩阵, 而在 PSKPE 算法中则是在物体允许的旋转范围内随机产生三个旋转角 ϑ , ψ 和 φ , 并由 (12) 式得到随机

旋转矩阵。显然 PSKPE 能够以更高的概率摆脱局部极小,因此当算法执行次数较少时,PSKPE 也能够以更高的概率得出正确的特征点匹配。在图 5 中随着算法执行次数的增加,当两种算法找到正确匹配的概率趋于相同时,PSKPE 所得出的 R_{est} 和 t_{est} 更接近真实的 R 和 t 。这是由于:PSKPE 在算法开

始时将世界坐标系的原点移动到了质点系的质心位置,从而使得算法之后的计算能够完全符合质点系运动原理的要求;空间共线性误差的引入使得特征点 P_k^* 上的受力计算更加准确;相比于文献[15-16]中的转动惯量矩阵的计算方式,(10)、(11)式的计算过程更简明,误差也更小。

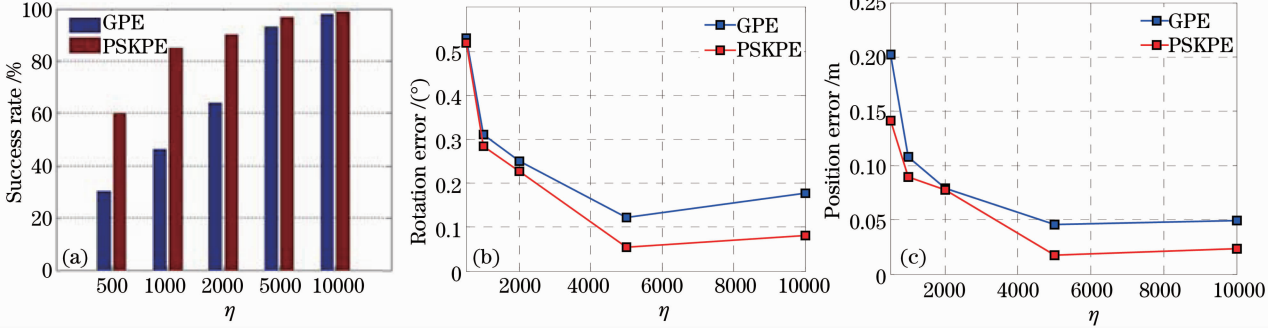


图 5 PSKPE 与 GPE 对比结果。(a)匹配成功率;(b)旋转误差;(c)位置误差

Fig. 5 Comparing results of PSKPE and GPE. (a) Matching success rate; (b) rotation error; (c) position error

表 1 给出了在图 2~4 所对应的实验条件下每次实验中 PSKPE 算法的耗时。表中,算法耗时约在 30~60 s 之间,最大耗时不超过 90 s。耗时大小主要取决于算法执行的次数与特征点的数量。由于在绝大多数实验中算法都是执行至设定的 20000 次后退出的,由此也可以算出算法单次执行的耗时在 1.5~3 ms 之间,最大不超过 5 ms。表 2 给出了令 $\sigma = 0.5$ pixel,3D 特征点数量分别为 $L=8,10,12,15$,算法最大执行次数 $\eta=20000$,不存在遮挡与伪特征点的情况下,PSKPE 算法与 GPE 算法的耗时对比。从表中能够看出两种算法耗时的最小差异为 0.7 s,约为两种算法平均耗时的 2.7%,最大耗时差异为 3.1 s,约为两种算法平均耗时的 5.6%。这说明两种算法的耗时差异不大,计算复杂度相似。综合前面的实验结果,可知本文提出的算法能够在不增加计算复杂度的情况下得到更优的结果。

表 1 实验中算法耗时

Table 1 Time consumptions in the tests unit: s

	$L=8$	$L=10$	$L=12$	$L=15$
$\sigma=0.5$ pixel	26.4	34.6	40.3	54.1
$\sigma=5$ pixel	24.6	31.4	39.8	54.0
$\sigma=10$ pixel	46.0	32.4	41.0	56.8
$a=1$	24.0	31.4	37.9	52.0
$a=2$	26.0	29.3	36.8	50.3
$a=3$	21.4	26.7	35.3	48.1
$b=10\%$	27.6	33.1	44.3	58.2
$b=20\%$	27.2	34.2	47.5	76.1
$b=30\%$	34.8	42.3	51.7	80.2
Average	28.6	32.8	41.6	62.1

表 2 PSKPE 与 GPE 算法的耗时对比

Table 2 Comparison of time consumptions between PSKPE and GPE unit: s

	$L=8$	$L=10$	$L=12$	$L=15$
PSKPE	26.4	34.6	40.3	54.1
GPE	25.7	33.1	41.4	57.2
Difference	0.7	1.5	1.1	3.1

3.2 真实图像实验

实验采用焦距为 5.2 mm 的摄像头对真实物体进行成像,图像尺寸为 640 pixel \times 480 pixel,主点位于(328.368 pixel, 249.901 pixel),单位像元尺寸为 8.8 $\mu\text{m} \times 8.8 \mu\text{m}$ 。设定算法的最高执行次数为 50000,其他参数与合成图像实验相同。

对物体 1 采用的 3D 特征点为 $P_1^*(10, 145.8, 0)$, $P_2^*(0, 0, 0)$, $P_3^*(100, 0, 0)$, $P_4^*(0, 0, 90)$, $P_5^*(10, 0, 100)$, $P_6^*(90, 0, 100)$,坐标单位为毫米,如图 6(a)所示。图中,这些点被顺次连接了起来,当 3D/2D 特征点正确匹配后将能在图像中得到类似的连接结果。图 6(b)为算法运行到第 50 次时特征点的匹配结果,显示了图像特征提取的情况,除了与 3D 特征点对应的点外,还有 3 个伪 2D 特征点。图 6(c)为算法运行终止后 3D/2D 特征点的匹配结果,可见此时的连线方式与图 6(a)中设定的连线方式是相符合的。为了反映位姿估计误差,将世界坐标系中顶点为 $A(0, 0, 0)$, $B(0, 0, 50)$, $C(100, 0, 50)$, $D(100, 0, 0)$, $E(0, 50, 0)$, $F(0, 50, 50)$, $G(100, 50, 50)$, $H(100, 50, 0)$ 的虚拟长方体,如图 7(a)所示,利用 R_{est} 和 t_{est} 投影到图像中,图 7(b)

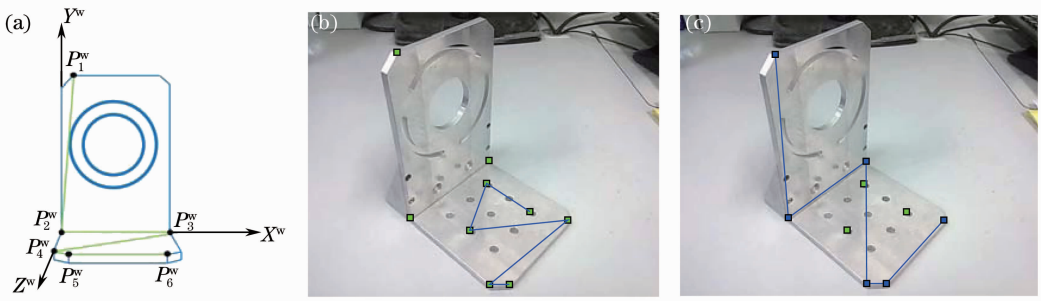


图 6 物体 1 的匹配结果。(a) 3D 特征点设置；(b) 算法运行至第 50 次时的结果；(c) 最终的特征点匹配结果
Fig. 6 Matching results of the 1st object. (a) Configuration of 3D feature points; (b) result after 50 runs of the algorithm; (c) final matching result

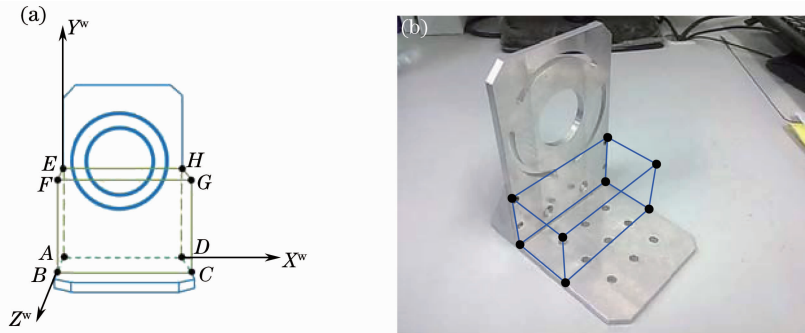


图 7 物体 1 的位姿估计。(a) 虚拟长方体的设置；(b) 图像中的投影结果

Fig. 7 Pose estimation of the 1st object. (a) Configuration of the virtual cuboid; (b) projected cuboid on the image

显示该虚拟长方体在图像中能够与周围的场景相符合,说明位姿估计误差很小。

上述实验重复进行 20 次,其中 19 次均实现了 3D/2D 特征点的正确匹配。在实现了正确匹配的实验中,算法最大执行次数为 45070,用时 49.65 s;最小执行次数 525,用时 0.59 s;平均执行次数 13857,用时 15.28 s。这些结果说明在真实图像实验中算法能够更快地得到匹配结果并估计位姿。

对物体 2 采用的 3D 特征点为 $P_1^w(0, 0, 0)$, $P_2^w(0, 56, 0)$, $P_3^w(0, 56, 110)$, $P_4^w(0, 0, 110)$, $P_5^w(110, 56, 110)$, $P_6^w(110, 56, 0)$, $P_7^w(110, 0, 0)$, $P_8^w(54, 38.4, 54)$, 坐标单位为毫米,如图 8(a)所

示。图中, P_7^w 由于遮挡在图像中无法显示,而 P_3^w 则由于所采用的特征提取算法不能够被提取出来,等效为被遮挡,所以顺次连接时忽略了这两个点。图 8(b)为图像特征提取的结果以及算法运行到第 50 次时特征点的匹配结果。图 8(c)为算法运行终止后 3D/2D 特征点的匹配结果。此外,将世界坐标系中端点为 $A(0, 56, 0)$, $B(0, 56, 110)$, $C(110, 56, 110)$, $D(110, 56, 0)$, $E(54, 70, 54)$ 的虚拟物体,如图 9(a)所示,利用 R_{est} 和 t_{est} 投影到图像中 [图 9(b)],图 9(b)显示该虚拟物体能够与周围场景符合。

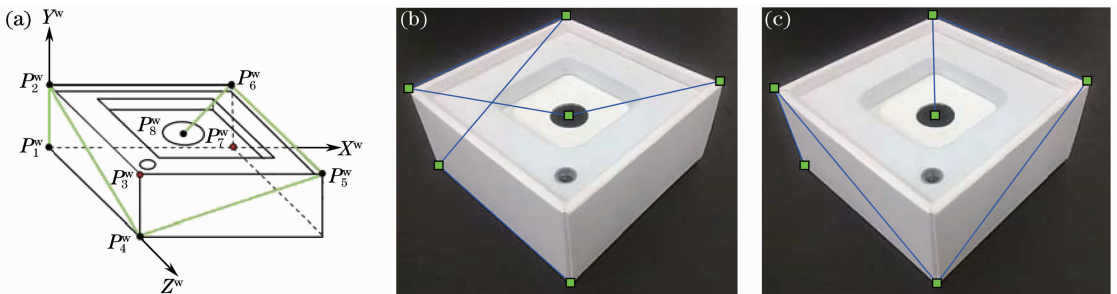


图 8 物体 2 的匹配结果。(a) 3D 特征点设置；(b) 算法运行至第 50 次时的结果；(c) 最终的特征点匹配结果
Fig. 8 Matching results of the 2nd object. (a) Configuration of 3D feature points; (b) result after 50 runs of the algorithm; (c) final matching result

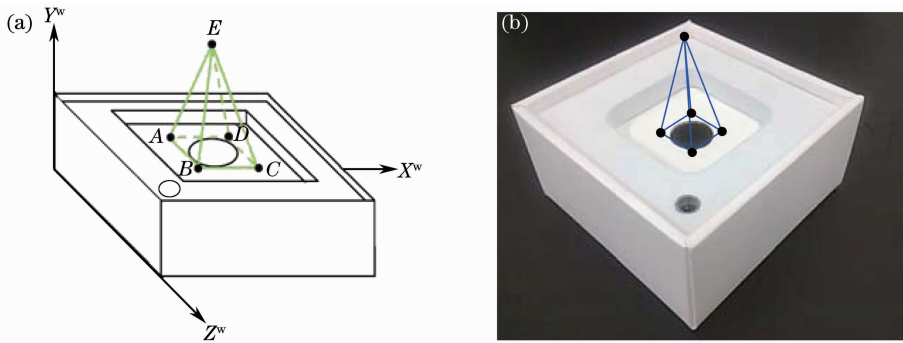


图9 物体2的位姿估计。(a)虚拟物体的设置；(b)图像中的投影结果

Fig. 9 Pose estimation of the 2nd object. (a) Configuration of the virtual object; (b) projected object on the image

将上述实验重复进行20次,20次实验均实现了3D/2D特征点的正确匹配。在这些实验中,算法最大执行次数为23461,用时27.76 s,最小执行次数为287,用时0.34 s,平均执行次数为6364,用时7.51 s。再次印证了算法能够在真实图像实验中更快地获得特征点对应和位姿估计结果。

4 结 论

针对3D/2D特征点对应关系未知的情况,提出了一种应用质点系运动原理同时确定特征点匹配与像机位姿的算法。通过引入空间共线性误差和匹配矩阵使得算法不仅能够应用于3D/2D特征点一一对应的情况,而且对于3D特征点存在遮挡,以及图像中存在2D伪特征点的情况也能够应用。通过实验,并与已有的算法进行比较,结果表明该算法能够在不增加计算复杂度的情况下以更高的概率得出特征点的正确匹配,并且受图像噪声和2D伪特征点的影响较小。未来将对同时存在3D特征点遮挡和2D伪特征点的情况进行研究,从而进一步扩展算法的应用范围。

参 考 文 献

- 1 L Quan, Z Lan. Linear N -point camera pose determination [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(8): 774–780.
- 2 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.
- 3 V Lepetit, F Moreno-Noguer, P Fua. EPnP: an accurate $O(n)$ solution to the PnP problem [J]. Int J Comput Vis, 2009, 81(2): 155–166.
- 4 M L Liu, K H Wong. Pose estimation using four corresponding points [J]. Pattern Recognition Letters, 1999, 20(1): 69–74.
- 5 Zhu Renzhang, Lin Yan, Zhang Lei. New algorithm of solving for ranges during final approach of spacecraft rendezvous [J]. J Beijing University of Aeronautics and Astronautics, 2006, 32(7): 764–768.
朱仁章, 林彦, 张磊. 航天器交会计算机视觉系统测距求解新算法[J]. 北京航空航天大学学报, 2006, 32(7): 764–768.
- 6 Zhu Renzhang, Lin Yan. Relative attitude estimation and control

- schemes for the final approach phase of spacecraft rendezvous [J]. J Beijing University of Aeronautics and Astronautics, 2007, 33(5): 544–548.
- 朱仁章, 林彦. 航天器交会最终逼近段相对姿态估计与控制 [J]. 北京航空航天大学学报, 2007, 33(5): 544–548.
- 7 Dai Dongkai, Wang Xingshu, Hu Chunsheng, *et al.*. Camera calibration and attitude measurement technology based on astronomical observation [J]. Acta Optica Sinica, 2012, 32(3): 0312005.
- 戴东凯, 王省书, 胡春生, 等. 基于天文观测的相机标定及姿态测量技术研究 [J]. 光学学报, 2012, 32(3): 0312005.
- 8 F Kahl, D Henrion. Globally optimal estimates for geometric reconstruction problems [J]. Int J Comput Vis, 2007, 74(1): 3–15.
- 9 G Schweighofer, A Pinz. Globally optimal $O(n)$ solution to the PnP problem for general camera models [C]. Proceedings of the 19th British Machine Vision Conference, 2008.
- 10 H Hmam, J Kim. Optimal non-iterative pose estimation via convex relaxation [J]. Image and Vision Computing, 2010, 28(11): 1515–1523.
- 11 M A Fishler, R C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography [J]. Communications of the ACM, 1981, 24(6): 381–395.
- 12 P David, D Dementhon, R Duraiswami, *et al.*. SoftPOSIT: simultaneous pose and correspondence determination [J]. Int J Comput Vis, 2004, 59(3): 259–284.
- 13 S Gold, A Rangarajan, C P Lu, *et al.*. New algorithms for 2D and 3D point matching: pose estimation and correspondence [J]. Pattern Recognition, 1998, 31(8): 1019–1031.
- 14 D Dementhon, L Davis. Model-based object pose in 25 lines of code [J]. Int J Comput Vis, 1995, 15(1–2): 123–141.
- 15 H F Ugurdag, S Goren, F Canbay. Correspondenceless pose estimation from a single 2D image using classical mechanics [C]. Proceedings of IEEE International Symposium on Computer and Information Sciences, 2008.
- 16 H F Ugurdag, S Goren, F Canbay. Gravitational pose estimation [J]. Computers and Electrical Engineering, 2010, 36(6): 1165–1180.
- 17 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.
- 18 Mechanical Department of Harbin Institute of Technology. Theoretical Mechanics(I) [M]. Beijing: Higher Education Press, 2009. 250–275.
哈尔滨工业大学理论力学教研室. 理论力学(I) [M]. 北京: 高等教育出版社, 2009. 250–275.
- 19 R Hartley, A Zisserman. Multiple View Geometry in Computer Vision [M]. Cambridge: Cambridge University Press, 2003. 87–127.