

## 基于加权加速正交迭代算法的相机位姿估计

熊 芝<sup>1,2</sup>, 许 航<sup>1,2</sup>, 张刘港<sup>3</sup>, 郭志豪<sup>1,2</sup>, 伍楚奇<sup>1,2</sup>, 冯 维<sup>1,2</sup>, 翟中生<sup>1,2</sup>, 周维虎<sup>1,4</sup>, 董登峰<sup>4</sup>

1. 湖北工业大学 机械工程学院, 湖北 武汉 430068;
2. 湖北省现代制造质量工程重点实验室, 湖北 武汉 430068;
3. 海宁集成电路与先进制造研究院, 浙江 海宁 314400;
4. 中国科学院微电子研究所, 北京 100029)

**摘要:** 单目视觉中的位姿估计是三维测量中的一个关键问题, 在机器视觉、精密测量等方面运用广泛。该问题可通过  $n$  点透视 (PnP) 算法求解, 正交迭代算法 (OI) 作为 PnP 算法的代表, 因其高精度的优点在实际中得到了广泛运用。为了进一步提高 OI 算法的稳健性和计算效率, 提出了一种加权加速正交迭代算法 (WAOI)。该方法首先根据经典正交迭代算法推导出加权正交迭代算法, 通过构建加权共线性误差函数, 利用物点重投影误差更新权值, 达到迭代优化位姿估算结果的目的; 在此基础上, 通过自适应权值, 整合每次迭代过程中平移向量以及目标函数的计算, 减少迭代过程中的计算量, 从而实现算法的加速。实验表明, 在 12 个参考点中存在两个粗差点的情况下, WAOI 的参考点重投影精度为 0.64 pixel, 运算时间为 8.02 ms, 精度高且运行速度快, 具有较强的工程实用价值。

**关键词:** 机器视觉; 位姿估计; 加权正交迭代; 自适应权值

中图分类号: TP391 文献标志码: A DOI: 10.3788/IRLA20220030

## Pose estimation of camera based on weighted accelerated orthogonal iterative algorithm

Xiong Zhi<sup>1,2</sup>, Xu Hang<sup>1,2</sup>, Zhang Liugang<sup>3</sup>, Guo Zhihao<sup>1,2</sup>, Wu Chuqi<sup>1,2</sup>,  
Feng Wei<sup>1,2</sup>, Zhai Zhongsheng<sup>1,2</sup>, Zhou Weihu<sup>1,4</sup>, Dong Dengfeng<sup>4</sup>

1. School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China;
2. Hubei Key Lab of Manufacture Quality Engineering, Wuhan 430068, China;
3. Haining Institute of Integrated Circuits and Advanced Manufacturing, Haining 314400, China;
4. Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

**Abstract:** Pose estimation in monocular vision is a key problem in three-dimensional measurement, which is widely used in machine vision, precision measurement and so on. This problem can be solved by  $n$ -point perspective (PnP) algorithm. Orthogonal iterative algorithm (OI), as the representative of PnP algorithm, has been widely used in practice because of its high precision. In order to further improve the robustness and computational efficiency of OI algorithm, a weighted accelerated orthogonal iterative algorithm (WAOI) is proposed in this paper. Firstly, the weighted orthogonal iterative algorithm is deduced according to the classical orthogonal iterative algorithm. The weighted collinearity error function is constructed and the weight is updated by using the object point reprojection error to achieve the purpose of iteratively optimizing the pose estimation results.

收稿日期: 2022-03-01; 修订日期: 2022-04-06

基金项目: 国家重点研发计划 (2019 YFB2006100)

作者简介: 熊芝, 女, 副教授, 博士, 主要从事大尺寸测量、激光及光电检测技术方面的研究。

Secondly on this basis, through adaptive weights, the calculation of translation vector and objective function in each iteration is integrated to reduce the amount of calculation in the iterative process, so as to accelerate the algorithm. The experimental results show that when there are two rough points in the 12 reference points, the reprojection accuracy of the reference point of WAOI is 0.64 pixel, the operation time is 8.02 ms, the accuracy is high and the running speed is fast, so it has strong engineering practical value.

**Key words:** machine vision; pose estimation; weighted orthogonal iterative; adaptive weights

## 0 引言

相机位姿估计在计算机视觉、机器人、摄影测量中有广泛的应用<sup>[1]</sup>,常用的方法是在相机内参已知的情况下,通过  $n$  个空间参考点与其所对应的图像点来估算相机的空间位置和姿态信息,这通常称为  $n$  点透视问题,即 PnP (perspective- $n$ -point) 问题<sup>[2]</sup>。

目前,求解 PnP 问题的算法主要分为非迭代算法和迭代算法<sup>[3]</sup>。非迭代算法计算速度快,但对噪声敏感,一般得不到较高的位姿精度。其中具有代表性的算法有 DLT (direct liner transformation) 算法<sup>[4]</sup>、EPnP (efficiency PnP) 算法<sup>[5]</sup>、RPnP (robust PnP) 算法<sup>[6]</sup>、OPnP (non-iterative PnP) 算法<sup>[7]</sup>、SRPnP (single gaussian-newton robust PnP) 算法<sup>[8]</sup> 等。迭代算法通过迭代计算位姿,精度较高,但也因此耗时较长,并且初始位姿的选取对最终的结果影响较大,容易收敛到局部最优值。迭代算法的代表有比例正交投影迭代变化 (pose from orthography and scaling with iterations, POSIT) 算法<sup>[9]</sup>,以及正交迭代 (orthography iterative, OI) 算法<sup>[10]</sup>。由于采用了奇异值分解 (Singular Value Decomposition, SVD), OI 算法可以得到正交的旋转矩阵,并且 OI 算法引入了空间共线性误差的概念,相较于 POSIT 算法收敛性更好,在初始值不精确的时候也能保证收敛到不错的位姿,因此正交迭代算法成为目前应用较为广泛的位姿估计方法。

OI 算法于 2000 年由 Lu 等<sup>[10]</sup> 提出的,后续经过了不断的发展完善。2009 年, Zhang 等<sup>[11]</sup> 提出了基于直线的正交迭代算法 (LBOI), 将 OI 算法中的点与点的对应推广到了直线形式; Didier 等<sup>[12]</sup> 提出了基于正交迭代算法的混合姿态估计方法 (COI), 通过正方形基准定位技术获取姿态初值带入到正交迭代算法中从而加快算法收敛速度并提高精度。2015 年,李鑫等<sup>[13]</sup> 提出了一种加速正交迭代算法 (OI+), 通过规整正交迭代算法中计算过程, 加速正交迭代算法从而减少算

法运行时间。2017 年, Huo 等<sup>[14]</sup> 提出了基于不确定加权测量误差的特征目标位姿估计的广义正交迭代算法 (GOI), 将正交迭代算法推广到立体视觉系统中进行位姿估计。2018 年, Sun 等<sup>[15]</sup> 提出了基于入射射线跟踪模型的正交迭代算法 (PROI), 用入射射线跟踪摄像机模型替代原有的针孔相机模型; 周润等<sup>[16]</sup> 提出了加权形式的正交迭代算法。2019 年, Dong 等<sup>[17]</sup> 提出了 soft+OI 的组合算法, 将 SoftAssign 算法与 OI 算法结合, 可以在不确定物点与像点的对应关系的情况下, 同时得到位姿和物点与像点的对应关系; 张雄峰等<sup>[18]</sup> 提出了基于 S 估计的加权正交迭代算法, 通过 S 估计法计算权值从而加强算法的鲁棒性。OI 算法经过推广扩大了其使用范围, 但由于迭代过程复杂, 耗时较多, 实际应用中受到一定制约。

为了提高 OI 算法的求解精度并且降低计算时间, 提出了一种加权加速正交迭代算法 (WAOI), 关键思想在于构建加权形式的目标函数, 通过物方重投影误差调整权重系数, 优化位姿估计结果提升解算精度; 通过自适应权值以及对迭代过程中冗余计算的整合, 减少迭代过程中的计算量, 实现算法的加速。

## 1 加权正交迭代算法

### 1.1 算法原理

经典正交迭代算法的基本原理是物点、像点、相机坐标系原点三点共线<sup>[10]</sup>, 即满足共线性方程:

$$RP_i + t = V_i(RP_i + t) \quad (1)$$

式中:  $P_i$  为第  $i$  个 ( $i = 1, \dots, n, n \geq 4$ ) 非共线参考点在目标坐标系下的三维坐标;  $V_i = \frac{v_i v_i^T}{v_i^T v_i}$  为视线投影矩阵,  $v_i$  为第  $i$  个参考点在归一化像平面上的投影坐标;  $R$  为旋转矩阵;  $t$  为平移向量。以空间共线性误差平方和作为目标函数, 通过最小化目标函数得到最优估计的  $R$  和  $t$ , 目标函数定义为:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^n \|\mathbf{e}_i\|^2 = \sum_{i=1}^n \|(I - \mathbf{V}_i)(\mathbf{R}\mathbf{P}_i + \mathbf{t})\|^2 \quad (2)$$

式中:  $\mathbf{I}$  为单位矩阵。经典正交迭代算法中赋予每个参考点的权值是相同的, 因此当测量数据中存在粗大误差时, 会对最终的姿态估计精度产生较大影响。为了提高正交迭代算法的精度, 考虑加权形式的正交迭代算法, 通过对误差较大的参考点赋予较小的权值, 来减小误差较大的点对最终位姿估计精度的影响<sup>[16]</sup>。加权形式的目标函数为:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^n \omega_i \|\mathbf{e}_i\|^2 = \sum_{i=1}^n \omega_i \|(I - \mathbf{V}_i)(\mathbf{R}\mathbf{P}_i + \mathbf{t})\|^2 \quad (3)$$

式中: 第  $i$  个参考点的权值为  $\omega_i$ 。假设  $\mathbf{R}$  的初始值为  $\mathbf{R}^{(0)}$ , 第  $k$  次迭代时的  $\mathbf{R}$  值为  $\mathbf{R}^{(k)}$ , 第  $i$  个参考点的权值为  $\omega_i^{(k)}$ 。由公式 (3) 可知, 目标函数为  $\mathbf{t}$  的二次函数, 因此在  $\mathbf{R}^{(k)}$  和  $\omega_i^{(k)}$  的值确定后, 可以得到第  $k$  次迭代的  $\mathbf{t}^{(k)}$  的线性最优解为:

$$\mathbf{t}^{(k)} = \left( \mathbf{I} - \sum_{i=1}^n \omega_i^{(k)} \mathbf{V}_i \right)^{-1} \sum_{i=1}^n \omega_i^{(k)} (\mathbf{V}_i - \mathbf{I}) \mathbf{R}^{(k)} \mathbf{P}_i \quad (4)$$

由  $\mathbf{R}^{(k)}$  和  $\mathbf{t}^{(k)}$  则可计算得到对应参考点的视线投影向量  $\mathbf{q}_i^{(k)}$ :

$$\mathbf{q}_i^{(k)} = \mathbf{V}_i (\mathbf{R}^{(k)} \mathbf{P}_i + \mathbf{t}^{(k)}) \quad (5)$$

$$\mathbf{R}^{(k+1)} = \arg \min_{\mathbf{R}} \sum_{i=1}^n \omega_i^{(k)} \|\mathbf{R}^{(k)} \mathbf{P}_i + \mathbf{t}^{(k)} - \mathbf{q}_i^{(k)}\|^2 \quad (6)$$

通过求解公式 (6) 的绝对定向问题得到  $\mathbf{R}^{(k+1)}$ 。  $\mathbf{R}^{(k+1)}$  的具体解算步骤如下: 定义  $\bar{\mathbf{P}} = \sum_{i=1}^n \omega_i^{(k)} \mathbf{P}_i$ ,  $\bar{\mathbf{q}} = \sum_{i=1}^n \omega_i^{(k)} \mathbf{q}_i^{(k)}$ , 则有:

$$\mathbf{q}_i^k = \mathbf{q}_i^{(k)} - \bar{\mathbf{q}} \quad (7)$$

$$\mathbf{P}_i^k = \mathbf{P}_i - \bar{\mathbf{P}} \quad (8)$$

参照经典正交迭代算法, 定义  $\mathbf{M} = \sum_{i=1}^n \omega_i^{(k)} \mathbf{q}_i^k \mathbf{P}_i^k$ , 对  $\mathbf{M}$  进行 SVD 分解, 则有:

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (9)$$

当  $\mathbf{R}^{(k+1)} = \mathbf{U}\mathbf{V}^T$  时, 公式 (6) 取得最小值, 此时物方残差和最小, 因此可以得到  $\mathbf{R}^{(k+1)}$ 。重复上述计算过程, 直至  $\mathbf{R}$  和  $\mathbf{t}$  满足收敛条件, 从而得到  $\mathbf{R}$  和  $\mathbf{t}$  的最优估计。

## 1.2 权值确定

参考点的权值, 采用物方重投影误差来确定。即

通过计算参考点在相机位姿估计结果下的物方投影误差, 利用收敛速度快的改进 *Huber* 函数<sup>[19]</sup> 确定权重系数; 对投影误差大的参考点, 认为其测量误差较大, 对其分配较小权重, 以提高位姿估计精度。具体步骤如下: 各参考点初始权值均设为  $1/n$ , 设第  $k$  次迭代时第  $i$  个参考点的权值为  $\omega_i^{(k)}$ , 利用 1.1 节的方法计算得到旋转矩阵  $\mathbf{R}^{(k)}$  和平移向量  $\mathbf{t}^{(k)}$ , 由公式 (5)、(7)、(8) 得到各参考点的物方投影误差:

$$r_i^{(k)} = \|\mathbf{R}^{(k)} \mathbf{P}_i - \mathbf{q}_i^k\| \quad (10)$$

利用改进型 *Huber* 函数更新各参考点权值系数, 改进型 *Huber* 函数的表达式为:

$$\omega_i^{(k+1)} = \begin{cases} \omega_i^{(k)}, & r_i^{(k)} \leq r \\ \left( r^2 / (r_i^{(k)})^2 \right) \omega_i^{(k)}, & r_i^{(k)} > r \end{cases} \quad (11)$$

式中:  $r = \frac{1}{n} \sum_{i=1}^n r_i^{(k)}$ 。重复上述迭代过程, 直到权重系数满足收敛条件为止。

## 2 加速加权正交迭代算法

### 2.1 加速原理

加权正交迭代算法由于每次迭代需要计算各参考点的物方投影误差并更新权值, 因此计算耗时较经典正交迭代算法较长。由仿真实验可知, 一般权值的调整在迭代一定次数便趋于稳定, 因此考虑在权值收敛后将权值设为定值, 从而减少算法迭代过程中的计算量; 进一步的, 通过整合固定权值的加权正交迭代算法中的冗余计算, 加快计算效率实现算法加速。具体加速步骤如下:

(1) 自适应权值。计算两次迭代的参考点权值之差的模, 若小于设定阈值则认为权值已收敛, 此时将权值设为最后一次更新的值, 后续计算将不对权值进行更新, 以减少计算量。

(2) 整合迭代过程中的冗余计算。当参考点权值为定值后, 由公式 (4) 可知,  $\mathbf{t}$  可由  $\mathbf{R}$  线性解出, 投影点和目标函数均为  $\mathbf{R}$  和  $\mathbf{t}$  的函数, 因此可以用  $\mathbf{R}$  表示。通过对迭代过程中计算的规整化, 将重复计算的部分用常数矩阵表示, 在迭代过程前计算, 减少迭代过程中的计算量进而加快迭代过程。

### 2.2 常数矩阵计算

为了计算方便, 引入矩阵计算公式:

$$vec(ABC) = (C^T \otimes A)vec(B) \quad (12)$$

式中:  $vec()$ 表示将矩阵按照列堆栈形成的向量;  $\otimes$ 表示 Kronecker 积。

首先对参考点进行零均值化,  $P_i' = P_i - \bar{P}$ ,  $\bar{P} = \sum_{i=1}^n \omega_i P_i$ , 根据最优  $t$  的计算以及公式 (12), 则有:

$$t^{(k)} = D_{3 \times 9} r^{(k)} \quad (13)$$

其中,  $r^{(k)} = vec(R^{(k)})$

$$D = \left( I - \sum_{i=1}^n \omega_i V_i \right)^{-1} \sum_{i=1}^n \omega_i P_i^T \otimes (V_i - I) = \left( I - \sum_{i=1}^n \omega_i V_i \right)^{-1} \sum_{i=1}^n \omega_i P_i^T \otimes V_i \quad (14)$$

则计算后的投影点坐标为:

$$q_i^k = V_i (R^{(k)} P_i + t^{(k)}) = (P_i^T \otimes V_i + V_i D) r^{(k)} = E_i r^{(k)} \quad (15)$$

在利用绝对定向问题求解最优姿态时需要计算矩阵  $M$ 。

$$M^{(k)} = \sum_{i=1}^n \omega_i (q_i^k - \bar{q}^k) P_i^T = \sum_{i=1}^n \omega_i E_i r^{(k)} P_i^T \quad (16)$$

式中:  $\bar{q}^k$ 为第  $k$ 次迭代时所有投影点的平均值, 令  $m^{(k)} = vec(M^{(k)})$ , 则有:

$$m^{(k)} = \sum_{i=1}^n \omega_i (P_i \otimes E_i) r^{(k)} = F_{9 \times 9} r^{(k)} \quad (17)$$

其中

$$F = \sum_{i=1}^n \omega_i (P_i \otimes E_i) = \sum_{i=1}^n \omega_i (P_i \otimes P_i^T \otimes V_i) + \sum_{i=1}^n \omega_i (P_i \otimes (V_i D)) = \sum_{i=1}^n \omega_i (P_i \otimes P_i^T \otimes V_i) + \sum_{i=1}^n \omega_i (P_i \otimes V_i) (D) \quad (18)$$

根据绝对定向最优解, 对其进行奇异值分解  $M^{(k)} = U \Sigma V^T$ , 则得到  $R$  的最优估计  $R^{(k+1)} = UV^T$ , 然后继续进行下一次迭代。

针对目标函数的优化, 则有:

$$E(R, t) = \sum_{i=1}^n \omega_i \|(I - V_i)(R P_i + t)\|^2 = \sum_{i=1}^n \omega_i \|(I - V_i)(P_i^T \otimes I + D) r\|^2 = r^T G_{9 \times 9} r \quad (19)$$

其中

$$G = \sum_{i=1}^n \omega_i (P_i \otimes I + D^T)(I - V_i)(P_i^T \otimes I + D) \quad (20)$$

参考点  $P_i$  均为零均值化后的参考点, 经过上述计算整合后, 每次迭代时可由矩阵  $F$  计算得到  $M$ , 再利用奇异值分解得到最优旋转矩阵  $R$ 。目标函数的计算可以通过  $G$  确定, 当权值为定值时,  $F$ 、 $G$  为常系数矩阵, 因此可以在迭代过程之前进行计算, 从而减少计算量, 达到加速的目的。

### 2.3 收敛性证明

对于迭代算法而言, 必须给出收敛性证明。根据全局收敛定理<sup>[20]</sup>, 算法收敛需要满足 3 个条件。

- (1) WAOI 计算结果是封闭的。
- (2) WAOI 每次迭代产生的  $R$  都是封闭的。
- (3) WAOI 误差函数在迭代中严格单调下降, 直至达到最终收敛条件。

对于第一个条件, 正交迭代算法可分为 3 个映射, 计算  $M$  矩阵,  $M$  矩阵的 SVD 分解,  $R$  的计算, 可以看出上述 3 个步骤的计算都是连续且封闭的, 封闭映射的组合依然是封闭的, 因此 WAOI 也是封闭的。

WAOI 每次迭代计算出的  $R$  都是正交矩阵, 封闭且有界, 因此第 2 个条件也是满足的。

对于 WAOI 误差函数严格单调性的证明, 首先给出误差函数的表达式:

$$E(R^{(k+1)}) = \sum_{i=1}^n \omega_i \|q_i^{(k+1)} - V_i q^{(k+1)}\|^2 = \sum_{i=1}^n \omega_i \|q_i^{(k+1)} - V_i q_i^{(k)} + V_i q_i^{(k)} - V_i q_i^{(k+1)}\|^2 = \sum_{i=1}^n \omega_i \|q_i^{(k+1)} - V_i q_i^{(k)}\|^2 - \sum_{i=1}^n \omega_i \|V_i q_i^{(k+1)} - V_i q_i^{(k)}\|^2 \quad (21)$$

通过  $R^{(k)}$  的计算公式 (9), 可知  $R$  为使得公式 (9) 取最小值时的值, 因此有:

$$\sum_{i=1}^n \omega_i \|q_i^{(k+1)} - V_i q_i^{(k)}\|^2 \leq \sum_{i=1}^n \omega_i \|q_i^{(k)} - V_i q_i^{(k)}\|^2 = E(\mathbf{R}^{(k)}) \quad (22)$$

由公式 (21)、(22) 可得:

$$E(\mathbf{R}^{(k+1)}) \leq E(\mathbf{R}^{(k)}) - \sum_{i=1}^n \omega_i \|V_i q_i^{(k+1)} - V_i q_i^{(k)}\|^2 \quad (23)$$

假设  $\mathbf{R}^{(k)}$  不是 WAOI 算法的收敛点, 则有  $\mathbf{R}^{(k+1)} \neq \mathbf{R}^{(k)}$ ,  $q_i^{(k+1)} \neq q_i^{(k)}$ , 进而可以得到  $\sum_{i=1}^n \|V_i q_i^{(k+1)} - V_i q_i^{(k)}\|^2 \neq 0$ ,  $E(\mathbf{R}^{(k+1)}) < E(\mathbf{R}^{(k)})$ , 从而得到 WAOI 误差函数是严格单调下降的, 因此可以证明 WAOI 是严格收敛的。即从任意数值的初始值, WAOI 都能收敛到一个固定值。

### 3 仿真分析

仿真实验中相机内参参照真实相机设置, 等效焦距为 2500, 图像分辨率为 2590 pixel×2 048 pixel。参考点在相机坐标系下的坐标等间距分布在 [-10 cm, 10 cm]×[-10 cm, 10 cm]×[300 cm, 310 cm] 的区域内。平移向量选择为参考点的中心。旋转矩阵的迭代初始值  $\mathbf{R}^{(0)}$ , 一般通过弱透视投影模型计算得出, 为了进一步增加 WAOI 算法的全局收敛性, 且加快收敛速度, 选取目前非迭代算法中精度较高且运行速度较快, 具有全局寻优的 SRPnP 算法, 将其位姿解算值作为旋转矩阵迭代初始值  $\mathbf{R}^{(0)}$ 。对参考点对应的像点添加均值为 0, 标准差为 0.1 pixel 的高斯噪声模拟图像处理误差。仿真平台为 Matlab R2017 a; CPU 型号为 Intel(R)Core(TM) i5-9400 F, 主频 2.90 GHz; 内存 16 GB。为便于评定位姿解算精度, 定义旋转矩阵误差  $e_R$  与平移向量  $e_t$  误差如下:

$$e_R = 2 \arccos \left[ 0.5 \sqrt{1 + \text{tr}(\mathbf{R}\mathbf{R}_{true}^T)} \right] \quad (24)$$

$$e_t = \frac{\|\mathbf{t}_{true} - \mathbf{t}\|}{\|\mathbf{t}_{true}\|}$$

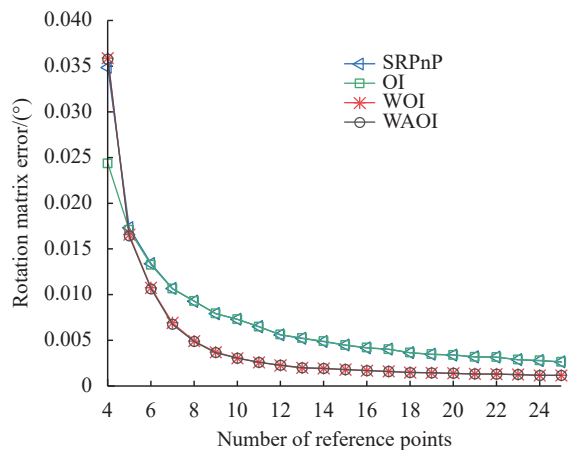
式中:  $\mathbf{R}$  和  $\mathbf{R}_{true}$  分别表示旋转矩阵解算值和真实值;  $\mathbf{t}$  和  $\mathbf{t}_{true}$  分别表示平移向量解算值和真实值<sup>[3]</sup>。对每种情况, 进行姿态解算独立重复试验 1000 次。

为比较不同算法之间的性能, 将文中的 WAOI 与提供迭代初值的 SRPnP<sup>[8]</sup>、经典正交迭代算法 OI<sup>[10]</sup>、加权正交迭代算法 WOI<sup>[16]</sup> 分别进行比较, 从参考点

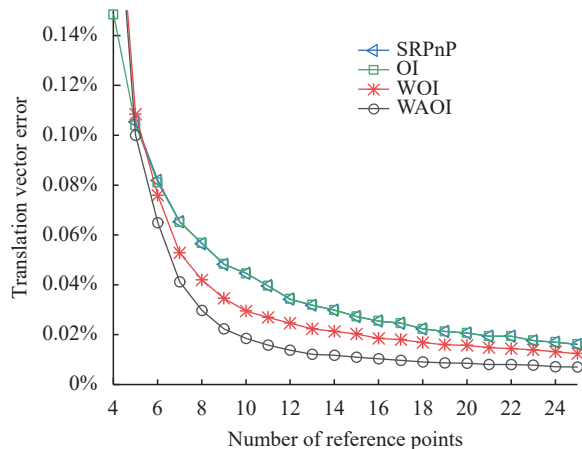
数目、噪声水平、粗差点数目以及运行时间 4 个方面进行对比。

#### (1) 参考点数目

取参考点数目范围 4~25, 在参考点中随机选取两个点, 对其图像坐标添加标准差为 1 pixel 的高斯噪声模拟粗差, 分别用上述 4 种算法进行姿态解算, 旋转矩阵与平移向量解算结果如图 1 所示。



(a) 旋转矩阵误差随参考点数目变化  
(a) Rotation matrix error varies with the number of reference points



(b) 平移向量误差随参考点数目变化  
(b) Translation vector error varies with the number of reference points

图 1 姿态解算误差随参考点数目变化图

Fig.1 Curve of attitude calculation error with the number of reference points

由图 1(a)可以看出, 随着参考点数的增加, 四种算法的旋转矩阵误差均在减小, 在参考点数目大于 4 时, 文中提出的 WAOI 在旋转矩阵的解算精度上高于 SRPnP 和 OI, 与 WOI 近似; 由图 1(b)可以看出, 在



平移向量的解算精度上, WAOI 的解算精度高于其他所有算法, 这是因为 WAOI 采用了物方投影误差作为权重系数的确定依据, 与 WOI 的像平面投影误差相比, 在平移向量的解算上会有较好的精度。WAOI 仅在参考点数目为 4 时姿态解算精度不及 SRPnP 和 OI, 这是因为 WAOI 通过给粗差点赋予较小的权值以减弱其对最终解算精度的影响, 这在参考点数目比较少的时候会造成参与计算的参考点数目不足从而造成比较大的解算误差。

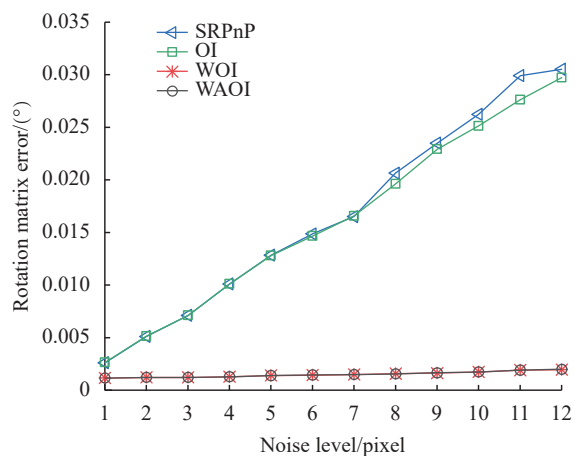
(2) 噪声水平

将参考点数目固定为 25 个, 粗差点数固定为两个, 改变粗差点噪声水平, 得到姿态解算误差随噪声变化曲线如图 2 所示。由图 2(a) 可以看出, 在旋转矩阵解算上 WAOI 和 WOI 均对噪声有较强的鲁棒性,

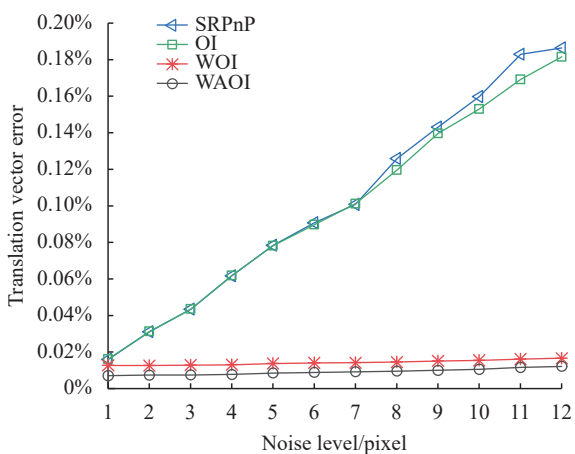
旋转矩阵解算误差基本不随噪声水平变化, 与 SRPnP 和 OI 对比具有较大的优势; 图 2(b) 中平移向量解算误差与图 2(a) 中旋转矩阵误差类似, WAOI 和 WOI 的平移向量解算误差基本不随噪声水平变化, 并且 WAOI 相较于 WOI 精度略高。

(3) 粗差点数目

固定粗差点的高斯噪声水平不变, 改变粗差点的数量, 得到姿态解算误差随粗差点数目变化曲线如图 3 所示。由图 3(a) 可以看出, 随着噪点数量的增多, 4 种算法的旋转矩阵误差均有所增大, WAOI 与 WOI 算法精度相近, 旋转矩阵误差约为 OI 和 SRPnP 的一半。由图 3(b) 可以看出, 4 种算法的平移向量误差也会随粗差点数量增多而加大, 解算精度方面, WAOI 优于其余 3 种算法。



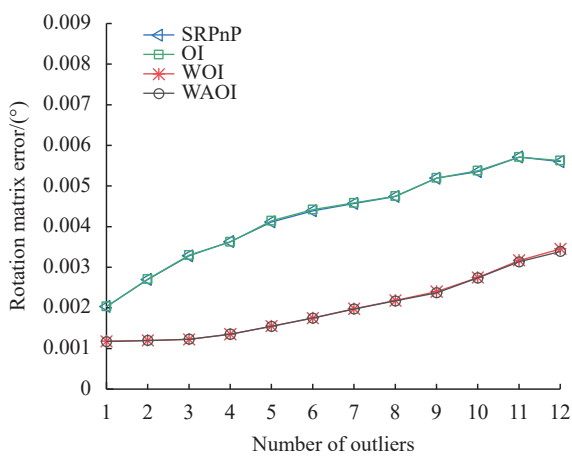
(a) 旋转矩阵误差随图像噪声变化  
(a) Rotation matrix error varies with the noise level



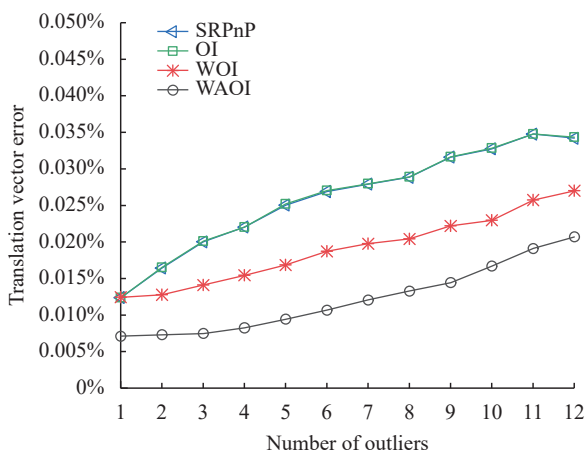
(b) 平移向量误差随图像噪声变化  
(b) Translation vector error varies with the noise level

图 2 姿态解算误差随图像噪声变化图

Fig.2 Attitude calculation error varying with noise level



(a) 旋转矩阵误差随粗差点数目变化  
(a) Rotation matrix error varies with the number of outliers



(b) 平移向量误差随粗差点数目变化  
(b) Translation vector error varies with the number of outliers

图 3 姿态解算误差随粗差点数目变化图

Fig.3 Attitude calculation error changing with the number of outliers

(4) 运行时间

4 种算法的解算时间随参考点数目变化如图 4 所示, 由此可以看出, 非迭代的 SRPnP 的计算时间最少, 且对参考点数目不敏感。迭代算法中 WAOI 耗时最少, 其次是 OI, 耗时最长是 WOI。

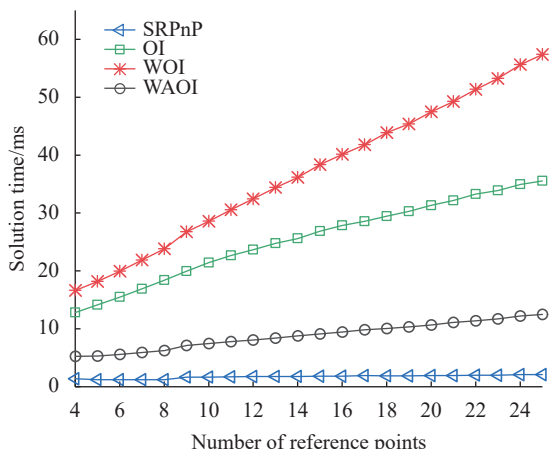


图 4 运行时间随参考点数量变化图

Fig.4 Solution time varies with the number of reference points

文中的 WAOI 优化了计算过程, 通过自适应权重, 整合迭代中的计算过程, 显著降低了计算耗时。在参考点数目为 4 个时, WAOI 相较于 OI 耗时降低了 59.11%, 相较于 WOI 耗时降低了 68.52%; 在参考点数目为 25 个时, WAOI 相较于 OI 耗时降低了 64.83%, 相较于 WOI 耗时降低了 78.21%。随着参考点数目的增多, WAOI 的解算效率的优势越发明显。

仿真实验表明, 当测量数据中存在粗差时, WAOI 与 SRPnP 和 OI 相比, 旋转矩阵以及平移向量均能达到较高的精度; 与 WOI 相比, 旋转矩阵精度相近, 平移向量精度略高, 表明 WAOI 在算法鲁棒性方面具有良好的效果, 在数据存在噪声的情况下依然能获得较高的解算精度。在算法耗时方面, WAOI 相较于 OI 和 WOI 均实现了较大的超越, 速度较快, 实时性较好。

4 实验验证

实验中相机选用德国 Basler ace acA2500-20 gm, 分辨率为 2590 pixel×2 048 pixel, 水平像元尺寸与垂直像元尺寸均为 4.8 μm, 镜头焦距为 12 mm, 相机的内参以及畸变参数由相机标定得到<sup>[21]</sup>。使用 12 个固

定参考点, 通过相机拍摄参考点图像, 实验装置如图 5 所示, 计算机配置与仿真实验相同。

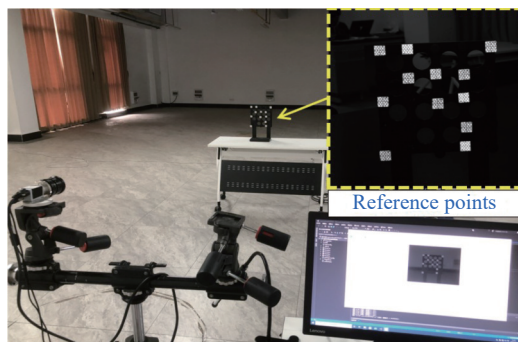


图 5 实验装置

Fig.5 Experiment equipment

随机选取两个参考点添加噪声模拟粗差。通过 4 种算法解算得到位姿参数, 计算参考点的像平面重投影结果, 通过比较重投影结果与实测值来反映位姿解算精度<sup>[22]</sup>。由于 WOI 与 WAOI 精度相近, 重投影图中不易分辨, 因此不包含 WOI。参考点在像平面上的重投影结果如图 6 所示。

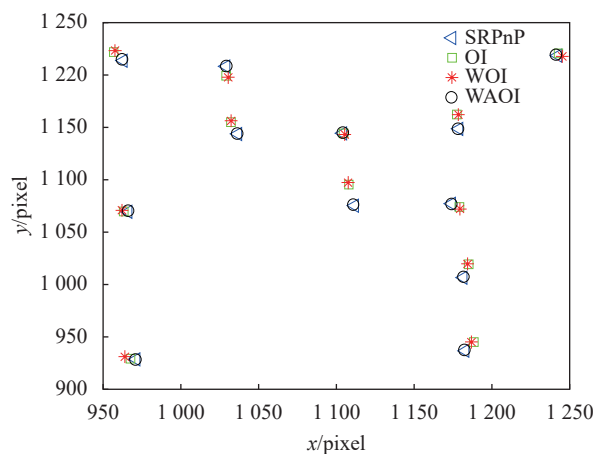


图 6 像平面重投影图

Fig.6 Reprojection results of image plane

从图 6 可以看出, WAOI 对应的解算结果与测量结果非常接近, 而 SRPnP 和 OI 的解算结果与测量结果有较大的偏差。进一步分析各方法测量精度差异, 计算 12 个参考点重投影位置如表 1 所示, 其中  $x$ 、 $y$  分别为参考点在像平面像素坐标,  $r$  为参考点重投影结果与测量结果之间的距离。

从表 1 可以看出, WAOI 解算得到的参考点重投影均方根误差为 0.64 pixel, SRPnP 和 OI 分别为

表 1 像平面投影结果比较 (单位: pixel)

Tab.1 Image plane projection results comparison (Unit: pixel)

Reference point	Measured		SRPnP calculated			OI calculated			WAOI calculated		
	$x$	$y$	$x_s$	$y_s$	$r_s$	$x_o$	$y_o$	$r_o$	$x_w$	$y_w$	$r_w$
1	1112.04	1075.67	1108.25	1095.76	20.44	1107.75	1097.65	22.39	1111.39	1076.11	0.78
2	962.89	1213.99	956.77	1222.14	10.19	957.69	1223.6	10.93	962.52	1214.76	0.86
3	1104.29	1144.72	1104.88	1144.68	0.59	1105.66	1143.59	1.78	1104.67	1144.76	0.39
4	1183.2	937.3	1188.71	945.64	9.99	1187.18	945.64	9.24	1182.99	937.53	0.31
5	1036.74	1144.08	1032.25	1155.25	12.04	1032.4	1156.62	13.27	1036.71	1144.02	0.07
6	1029.16	1208.41	1029.15	1199.62	8.79	1030.57	1198.03	10.48	1029.72	1208.45	0.56
7	1242.78	1218.83	1243.26	1220.98	2.2	1245.74	1217.98	3.07	1241.91	1219.16	0.93
8	1174.45	1077.53	1179.45	1074.51	5.84	1179.88	1072.5	7.4	1174.67	1076.73	0.83
9	1181.95	1007.05	1185.44	1019.91	13.33	1184.75	1020.2	13.45	1182.32	1007.41	0.52
10	966.16	1069.77	963.33	1070.24	2.87	962.12	1071.32	4.33	966.38	1070.43	0.7
11	971.22	929.21	967.57	929.76	3.69	963.96	931.69	7.67	971.2	928.51	0.7
12	1179.21	1148.97	1177.77	1162.65	13.75	1178.8	1162.49	13.53	1178.89	1148.63	0.47

10.29 pixel 和 11.18 pixel。WAOI 与 SRPnP 和 OI 相比,参考点重投影结果与测量结果之间的距离要低一个数量级,表明 WAOI 的位姿解算精度更高。此外解算时间方面,OI 平均耗时为 23.64 ms,WOI 平均耗时为 31.89 ms,WAOI 平均耗时为 8.02 ms,实验表明,WAOI 与 OI 和 WOI 相比,运算效率分别提升了 66.07% 和 74.85%,具有明显优势。

### 5 结论

文中提出了一种用于相对位姿标定的加权加速正交迭代算法,利用加权共线误差作为目标函数,通过物方投影误差更新权值,利用自适应权值和整合迭代过程加快收敛过程,优化位姿估计结果,具有精度高,鲁棒性好,实时性好的优点。仿真分析和实验结果表明,WAOI 的姿态解算精度随噪声影响较小,在强噪声水平下依然可以达到较高的精度,鲁棒性好,相较于 WOI 提升了平移向量解算精度。在 12 个参考点中存在两个粗差点的情况下,WAOI 的参考点重投影精度为 0.64 pixel,与 SRPnP 和 OI 相比高一个量级;运算时间为 8.02 ms,运算效率与 OI 相比提升了 66.07%,与 WOI 相比,提升了 74.85%,实时性好,具有较强的工程实用价值。

### 参考文献:

[1] Wang Ping, Zhou Xuefeng, An Aimin, et al. Robust and linear

solving method for Perspective-n-Point problem [J]. *Chinese Journal of Scientific Instrument*, 2020, 41(9): 271-280. (in Chinese)

[2] Liu Jinbo, Guo Pengyu, Li Xin, et al. Evaluation strategy for camera pose estimation algorithm based on point correspondences [J]. *Acta Optica Sinica*, 2016, 36(5): 0515002. (in Chinese)

[3] Wang Jiabao, Zhang Shirong, Zhou Qingya. Vision based real-time 3D displacement measurement using weighted iterative EPnP algorithm [J]. *Chinese Journal of Scientific Instrument*, 2020, 41(2): 166-175. (in Chinese)

[4] Abdel-aziz Y I, Karara H M, Hauck M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry [J]. *Photogrammetric Engineering & Remote Sensing*, 2015, 81(2): 103-107.

[5] Zhang Huijuan, Xiong Zhi, Lao Dabao, et al. Monocular vision measurement system based on EPnP algorithm [J]. *Infrared and Laser Engineering*, 2019, 48(5): 0517005. (in Chinese)

[6] Li S, Xu C, Xie M. 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.

[7] Zheng Y, Kuang Y, Sugimoto S, et al. Revisiting the PnP problem: A fast, general and optimal solution[C]//IEEE International Conference on Computer Vision, 2013: 2344-2351.

[8] Wang P, Xu G, Cheng Y, et al. A simple, robust and fast method for the perspective-n-point problem [J]. *Pattern Recognition Letters*, 2018, 108(6): 31-37.



- [9] Lao Dabao, Zhang Huijuan, Xiong Zhi, et al. Automatic measurement method of attitude based on monocular vision [J]. *Acta Photoica Sinica*, 2019, 48(3): 0315001. (in Chinese)
- [10] Lu C P, Hager G D, Mjolsness E. Fast and globally convergent pose estimation from video images [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, 22(6): 610-622.
- [11] Zhang X, Wang K, Zhang Z, et al. A new line-based orthogonal iteration pose estimation algorithm[C]//IEEE International Conference on Information Engineering and Computer Science, 2009: 1-4.
- [12] Didier J Y, Ababsa F E, Mallem M. Hybrid camera pose estimation combining square fiducials localization technique and orthogonal iteration algorithm [J]. *International Journal of Image and Graphics*, 2008, 8(1): 169-188.
- [13] Li Xin, Long Gucan, Liu Jinbo, et al. Accelerative orthogonal iteration algorithm for camera pose estimation [J]. *Acta Optica Sinica*, 2015, 35(1): 0115004. (in Chinese)
- [14] Huo J, Zhang G, Cui J, et al. A novel algorithm for pose estimation based on generalized orthogonal iteration with uncertainty-weighted measuring error of feature points [J]. *Journal of Modern Optics*, 2017, 65(3): 331-341.
- [15] Sun C, Dong H, Zhang B, et al. An orthogonal iteration pose estimation algorithm based on an incident ray tracking model [J]. *Measurement Science and Technology*, 2018, 29(9): 095402.
- [16] Zhou Run, Zhang Zhengyu, Huang Xuhui. Weighted orthogonal iteration algorithm for camera pose estimation [J]. *Acta Optica Sinica*, 2018, 38(5): 0515002. (in Chinese)
- [17] Dong H, Sun C, Zhang B, et al. Simultaneous pose and correspondence determination combining softassign and orthogonal iteration [J]. *IEEE Access*, 2019, 7: 137720-137730.
- [18] Zhang Xiongfeng, Liu Haibo, Shang Yang. Robust orthogonal iteration algorithm for single camera pose estimation [J]. *Acta Optica Sinica*, 2019, 39(9): 0915004. (in Chinese)
- [19] Wang Lixing, Cao Fuyuan. HuberLoss Based nonnegative matrix factorization algorithm [J]. *Computer Science*, 2020, 47(11): 80-87. (in Chinese)
- [20] Qi Zhan, Li maojun, Mo Hong, et al. Modified genetic algorithm based on state - space model and its convergence analysis [J]. *Control Theory & Applications*, 2020, 37(10): 2115-2122. (in Chinese)
- [21] Wang Jing, Wei Liang, Xiang Wenhao, et al. High-precision camera calibration method considering projected circular edge blur and eccentricity error [J]. *Infrared and Laser Engineering*, 2021, 50(12): 20210130. (in Chinese)
- [22] Liu Yan, Lei Boping, Fan Bin, et al. Target positioning technology and its structural parameter optimization based on vision measurement [J]. *Infrared and Laser Engineering*, 2020, 49(S2): 20200191. (in Chinese)