光电技术及应用

DOI：10.16818/j.issn1001－5868.2021092802

基于测量误差不确定性加权的快速相机位姿估计算法

徐　浩[1,2]，张秀山[1*]，黄　健[1]，朱　磊[2]

（1. 海军工程大学 电子工程学院，武汉 430000；2. 32382部队，武汉 430000）

摘　要： 在相机位姿估计任务中，参考点位置的测量容易受测量仪器或者所处环境影响，出现一定程度的测量误差，其中误差较大的点对最终求解精度的影响很大，被称为野值点。文章针对现有算法抗野值点能力较弱的问题，提出一种基于测量误差不确定性加权的快速相机位姿估计算法。该算法以经典的正交迭代算法为基础，引入 M 估计方法，对目标函数根据参考点的测量误差赋予相应大小的权值，最大程度地将野值点的影响排除，并引入 Kronecker 积对计算过程进行优化，减少了迭代过程中的冗余计算，提高运算速度。仿真实验结果表明，改进后的算法提升了原算法的抗野值点能力，且计算速度更快。

关键词： 相机位姿估计；正交迭代算法；M 估计方法

中图分类号： TP391　**文章编号：** 1001－5868(2022)01－0169－07

Fast Camera Pose Estimation Algorithm Based on Measurement Error Uncertainty Weighting

XU Hao[1,2], ZHANG Xiushan[1], HUANG Jian[1], ZHU Lei[2]

(1. College of Electronic Engineering, Naval University of Engineering, Wuhan 430000, CHN;
2. PLA Unit 32382, Wuhan 430000, CHN)

Abstract: In the task of camera pose estimation, the measurement of the position of the reference point is easily affected by the measuring instrument or the environment, leading a certain degree of measurement error. And the larger error point has a great influence on the final solution accuracy, which is called the outliers. Aiming at the problem that the existing algorithms have weak ability to resist outliers, a fast camera pose estimation algorithm based on measurement error uncertainty weighting is proposed. Based on the classic orthogonal iterative algorithm, the algorithm introduces M estimation method, assigns the corresponding weight to the objective function according to the measurement error of the reference point, eliminates the influence of the outlier point to the greatest extent, and introduces the Kronecker product to optimize the calculation process. Thus the redundant calculations in the iterative process are reduced and the calculation speed is increased. The simulation results show that the improved algorithm not only improves the anti-outliers ability of the original algorithm, but also realizes faster calculation speed.

Key words: camera pose estimation; orthogonal iterative algorithm; M estimation method

0　引言

相机位姿估计广泛用于机器人定位[1]、目标识别与跟踪[2]、增强现实[3]以及相机标定[4-5]等方面。

在已知相机内参的基础上，通过世界坐标系下的特征点坐标和成像平面像素点坐标之间的对应关系，求解包括旋转矩阵和平移向量的相机位姿的过程叫做 PnP 问题。该问题最早由 Fischler 等[6]提出，并指出要获得其封闭解，至少需要 3 对匹配点对。PnP 问题的解决思路主要有线性求解和非线性

收稿日期：2021－09－28.

*通信作者：张秀山　　E-mail：xiu3zh@139.com

迭代求解两种，其中线性求解方法的求解效率高、计算量小，但估算位姿精度不高。经典的如直接线性变换(DLT)算法[7]，利用世界坐标系下的特征点和其在图像坐标系下的投影点之间的线性关系构建方程组，再利用最小二乘法求解，1 对匹配点可以构建两个方程，因此至少需要 6 对匹配点才能求出稳定解，无法对 4 或 5 个参考点的情况进行求解。大多数非迭代 PnP 算法会首先求解特征点的深度，以获得特征点在相机坐标系中的 3D 坐标，而高效 PnP(EPnP)算法[8]则是将世界坐标系中的 3D 坐标表示为 4 个非共面控制点的加权和，从而将求解 2D-3D 的 PnP 问题转换为求解经典 3D-3D 的刚体变换问题，计算结果可靠，但 EPnP 在求解过程中采用了线性化策略，在点数较少的情况下求解精度较低。鲁棒 PnP(RPnP)算法[9]将参考点每三个点组成一个点集，然后利用 P3P 方法构建一系列的四阶多项式，并计算求出最优解。其他如 P3P[10]、直接最小二乘法(DLS)[11]等算法都是利用建立线性方程组进行求解的。

非线性迭代求解方法，通常利用物方残差或者像方残差作为目标函数，使用高斯-牛顿迭代(GN)算法和列文伯格-马夸尔特(LM)算法等常见的迭代算法，迭代最小化目标函数，求解得出相机位姿，非线性迭代算法求解相机位姿精度高，但是计算量大，实时性较差。

目前，正交迭代算法(LHM)[12]是性能最优的非线性迭代求解相机位姿的算法之一，相比于其他算法，其迭代次数少、求解精度高，但该算法没有考虑到参考点的测量误差对求解精度的影响，因此求解精度还有提高的空间，同时迭代计算过程中存在的冗余计算也浪费了计算资源，影响了算法运行效率。鉴于此，本文针对传统正交迭代算法因未将测量误差考虑进去而导致估计误差的问题，提出了基于测量误差不确定性加权的快速相机位姿估计方法。其核心思想是对迭代目标函数根据参考点的测量误差分别赋予相应大小的权值，平衡参考点测量误差对整体求解精度的贡献度，从而减少测量误差对精度的影响，同时对迭代过程进行加速操作，对重复的计算常量进行优化，以提高算法的运算速度。

1 基于测量误差不确定性加权的快速相机位姿估计算法

在使用迭代法求解相机位姿的各种算法中，主要使用两种误差作为迭代的目标函数，图 1 中给出了物方残差和像方残差的物理意义。如图所示，O_c-$X_cY_cZ_c$ 表示相机坐标系，$\boldsymbol{X}_i$ 为参考点在世界坐标系下的坐标观测值，$\boldsymbol{x}'_i$ 为参考点 $\boldsymbol{X}_i$ 在图像平面上的投影点估计值；$\boldsymbol{x}_i$ 为投影点在图像上的实际坐标值，$\boldsymbol{X}'_i$ 为 $\boldsymbol{x}_i$ 在测量视线下参考点的估计位置坐标，E_A 表示物方残差又叫代数误差，E_G 表示像方残差又叫几何误差[13]。

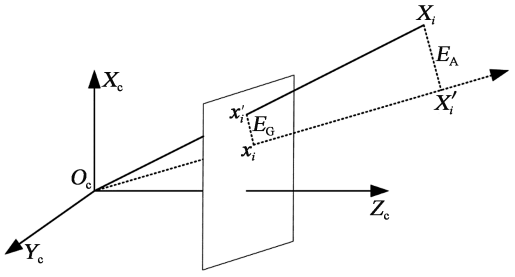

图 1 像方残差和物方残差示意图

1.1 正交迭代算法

正交迭代算法是以物方残差为目标函数的迭代算法，物方残差是图中 $\boldsymbol{X}_i$ 和 $\boldsymbol{X}'_i$ 之间的距离，其中 $\boldsymbol{X}_i$ 为空间参考点的观测值，$\boldsymbol{X}'_i$ 是图像平面上的投影点在视线方向按照透视投影模型的空间位置得到的投影点估计值，可以表示为

$$E(\boldsymbol{R},\boldsymbol{t})=\min_{\boldsymbol{R},\boldsymbol{t}}\sum_{i=1}^{n}\left\|(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)(\boldsymbol{R}\boldsymbol{P}_i+\boldsymbol{t})\right\|^2 \quad (1)$$

其中，$\boldsymbol{I}$ 为单位矩阵，$\hat{\boldsymbol{V}}_i=\hat{\boldsymbol{v}}_i\hat{\boldsymbol{v}}_i^{\mathrm{T}}/(\hat{\boldsymbol{v}}_i^{\mathrm{T}}\hat{\boldsymbol{v}}_i)$ 为视线 O_cx_i 确定的投影矩阵，$\hat{\boldsymbol{v}}_i=(u_i,v_i,1)$ 为参考点在成像平面上的像点坐标，$\boldsymbol{P}_i$ 是参考点在世界坐标系下的坐标，$\boldsymbol{R}$，$\boldsymbol{t}$ 分别为旋转矩阵和平移向量。

上述目标函数在已知第 k 次旋转矩阵 $\boldsymbol{R}^{(k)}$ 的情况下，平移向量 $\boldsymbol{t}$ 可以通过对式(1)关于 $\boldsymbol{t}$ 的偏导求得：

$$\boldsymbol{t}^{(k)}=\frac{1}{n}\left(\boldsymbol{I}-\frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{V}}_i\right)^{-1}\sum_{i=1}^{n}(\hat{\boldsymbol{V}}_i-\boldsymbol{I})\boldsymbol{R}^{(k)}\boldsymbol{P}_i \quad (2)$$

然后可以通过求解式(3)的绝对定向问题得到 $\boldsymbol{R}^{(k+1)}$：

$$\boldsymbol{R}^{(k+1)}=\arg\min_{\boldsymbol{R}}\sum_{i=1}^{n}\left\|\boldsymbol{R}\boldsymbol{P}_i+\boldsymbol{t}-\boldsymbol{q}_i^{(k)}\right\|^2 \quad (3)$$

其中，$\boldsymbol{q}_i^{(k)}=\hat{\boldsymbol{V}}_i(\boldsymbol{R}^{(k)}\boldsymbol{P}_i+\boldsymbol{t}^{(k)})$。

然后不断迭代求解，直到算法收敛或者物方残差小于设置的阈值，即可求得相机位姿。

1.2 基于测量误差不确定性加权的快速正交迭代算法

经典的正交迭代算法是建立在参考点和投影点

的测量“绝对精确”的基础上，但是在实际情况下，位置的测量容易受测量仪器或者参考点所处环境的影响，如受参考点所处空间限制，无法使用高精度的测量仪器测量参考点的位置，只能通过间接方法计算得到，此时，参考点位置往往存在一定程度上的测量误差。经典的正交迭代算法没有把这些测量误差考虑进去，给所有的参考点都赋予一样的权重，即认为所有参考点对最终位姿结果影响一样，这显然会造成误差。因此为了确保从包含不同程度测量误差的投影点坐标数据中获得准确的相机位姿，在经典正交迭代算法的基础上，采用归一化加权的方法来描述测量误差对结果的影响程度，包含测量误差越小的点，赋予的权重越大；包含测量误差越大的点，赋予的权重越小。通过调节权重的大小，可以将测量误差大的异常数据对整体结果的影响进行抑制甚至排除，使得最终的位姿结果更准确、鲁棒性更高。

1.2.1　测量误差不确定性加权的正交迭代算法

包含权重的目标函数可以表示为

$$E(\boldsymbol{R},\boldsymbol{t})=\min_{\boldsymbol{R},\boldsymbol{t}}\sum_{i=1}^{n}\omega_i\left\|(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)(\boldsymbol{R}\boldsymbol{P}_i+\boldsymbol{t})\right\|^2 \tag{4}$$

其中，$\omega_i>0$，表示第 i 个测量数据的权重系数。

加权之后的目标函数求解过程与正交迭代算法的求解过程基本一致。相应的已知第 k 次的旋转矩阵 $\boldsymbol{R}^{(k)}$ 之后，平移向量 $\boldsymbol{t}^{(k)}$ 可以表示为

$$\boldsymbol{t}^{(k)}=\left[\sum_{i=1}^{n}\omega_i(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)\right]^{-1}\sum_{i=1}^{n}\omega_i(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)\boldsymbol{R}^{(k)}\boldsymbol{P}_i \tag{5}$$

第 $k+1$ 次的旋转矩阵 $\boldsymbol{R}^{(k+1)}$ 可以表示为

$$\boldsymbol{R}^{(k+1)}=\underset{\boldsymbol{R}^{(k)}}{\operatorname{argmin}}\sum\omega_i\left\|\boldsymbol{R}\boldsymbol{P}_i+\boldsymbol{t}-\boldsymbol{q}_i^{(k)}\right\|^2 \tag{6}$$

其中，$\boldsymbol{q}_i^{(k)}=\hat{\boldsymbol{V}}_i(\boldsymbol{R}^{(k)}\boldsymbol{P}_i+\boldsymbol{t}^{(k)})$。

求解式(5)与前面的求解过程类似，也是求解绝对定向问题，但是加入权值之后，坐标系原点发生了变化，因此求解过程也需要做一些改变，首先定义标记点坐标的加权平均值 $\bar{P}=\sum_{i=1}^{n}\omega_i\boldsymbol{P}_i$，$\boldsymbol{q}^{(k)}=\sum_{i=1}^{n}\omega_i\boldsymbol{q}_i^{(k)}$，定义参考点坐标与加权平均值之差，$\boldsymbol{P}'_i=\boldsymbol{P}_i-\bar{\boldsymbol{P}}$，$\boldsymbol{q}'^{(k)}_i=\boldsymbol{q}_i^{(k)}-\boldsymbol{q}^{(k)}$，定义矩阵 $\boldsymbol{M}=\sum_{i=1}^{n}\omega_i\boldsymbol{q}'^{(k)}_i\boldsymbol{P}'^{\mathrm{T}}_i$，对 $\boldsymbol{M}$ 矩阵进行 SVD 分解，得到 $\boldsymbol{M}=\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\mathrm{T}}$，其中 $\boldsymbol{U}$，$\boldsymbol{V}$ 为正交矩阵，$\boldsymbol{D}$ 为对角矩阵。由文献[12]可知 $\boldsymbol{R}^{(k+1)}=\boldsymbol{V}^{\mathrm{T}}\boldsymbol{U}$。

1.2.2　正交迭代算法加速

正交迭代算法作为非线性迭代算法，计算时间相对于线性算法长很多，但是仍有优化的空间。文献[14]提出引入 Kronecker 积，对矩阵进行拉直操作，从而对迭代过程中的重复性计算进行简化。

首先引入 Kronecker 积和拉直运算的关系式：

$$\operatorname{vec}(\boldsymbol{A}\boldsymbol{X}\boldsymbol{B})=(\boldsymbol{B}^{T}\otimes\boldsymbol{A})\operatorname{vec}(\boldsymbol{X}) \tag{7}$$

式中，vec()表示将矩阵按列堆栈成一个列向量，也就是拉直操作，表示的是 Kronecker 积。

首先对加权之后的 $\boldsymbol{t}^{(k)}$ 进行拉直操作，得到：

$$\operatorname{vec}(\boldsymbol{t}^{(k)})=\left[\sum_{i=1}^{n}\omega_i(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)\right]^{-1}\sum_{i=1}^{n}\omega_i\left[\boldsymbol{P}'^{\mathrm{T}}_i\otimes(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)\right]\operatorname{vec}(\boldsymbol{R}^{(k)}) \tag{8}$$

令 $\left[\sum_{i=1}^{n}\omega_i(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)\right]^{-1}\sum_{i=1}^{n}\omega_i\left[\boldsymbol{P}'^{\mathrm{T}}_i\otimes(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)\right]=\boldsymbol{G}_{3\times9}$，$\operatorname{vec}(\boldsymbol{R}^{(k)})=\boldsymbol{r}$，则上式又可以化简为

$$\operatorname{vec}(\boldsymbol{t}^{(k)})=\boldsymbol{G}_{3\times9}\boldsymbol{r} \tag{9}$$

对投影点坐标 $\boldsymbol{q}_i^{(k)}$ 拉直，得到：

$$\operatorname{vec}(\boldsymbol{q}_i^{(k)})=\operatorname{vec}(\boldsymbol{R}^{(k)}\boldsymbol{P}_i+\boldsymbol{t}^{(k)})=(\boldsymbol{P}_i^{\mathrm{T}}\otimes\hat{\boldsymbol{V}}_i+\hat{\boldsymbol{V}}_i\boldsymbol{G})\boldsymbol{r} \tag{10}$$

$$\operatorname{vec}\left(\sum_{i=1}^{n}\omega_i\boldsymbol{q}_i\right)=\sum_{i=1}^{n}\omega_i\boldsymbol{q}_i=\sum_{i=1}^{n}\omega_i(\boldsymbol{P}_i^{\mathrm{T}}\otimes\hat{\boldsymbol{V}}_i+\hat{\boldsymbol{V}}_i\boldsymbol{G})\boldsymbol{r}=\boldsymbol{A}\boldsymbol{r} \tag{11}$$

其中，$\boldsymbol{A}=\sum_{i=1}^{n}\omega_i(\boldsymbol{P}_i^{\mathrm{T}}\otimes\hat{\boldsymbol{V}}_i+\hat{\boldsymbol{V}}_i\boldsymbol{G})$。

然后对 $\boldsymbol{M}$ 矩阵进行操作：

$$\boldsymbol{M}=\sum_{i=1}^{n}\omega_i\boldsymbol{q}'^{(k)}_i\boldsymbol{P}'^{\mathrm{T}}_i=\left(\sum_{i=1}^{n}\omega_i\boldsymbol{q}_i\boldsymbol{P}_i^{\mathrm{T}}\right)-\frac{1}{\sum_{i=1}^{n}\omega_i}\left(\sum_{i=1}^{n}\omega_i\boldsymbol{q}_i\right)\left(\sum_{i=1}^{n}\omega_i\boldsymbol{P}_i\right)^{\mathrm{T}} \tag{12}$$

$$\operatorname{vec}(\boldsymbol{M})=\left(\sum_{i=1}^{n}\omega_i\boldsymbol{P}_i\otimes(\boldsymbol{P}_i^{\mathrm{T}}\otimes\hat{\boldsymbol{V}}_i+\hat{\boldsymbol{V}}_i\boldsymbol{G})\boldsymbol{r}\right)-\frac{1}{\sum_{i=1}^{n}\omega_i}\left(\sum_{i=1}^{n}\omega_i\boldsymbol{P}_i\right)\otimes\boldsymbol{A}\boldsymbol{r}=\boldsymbol{B}_{9\times9}\boldsymbol{r} \tag{13}$$

其中：

$$\boldsymbol{B}_{9\times9}=\sum_{i=1}^{n}\omega_i\boldsymbol{P}_i\otimes(\boldsymbol{P}_i^{\mathrm{T}}\otimes\hat{\boldsymbol{V}}_i+\hat{\boldsymbol{V}}_i\boldsymbol{G})-\frac{1}{\sum_{i=1}^{n}\omega_i}\left(\sum_{i=1}^{n}\omega_i\boldsymbol{P}_i\right)\otimes\boldsymbol{A} \tag{14}$$

得到 $\boldsymbol{M}$ 之后就可以通过 SVD 得到 $\boldsymbol{R}^{(k+1)}$。迭代结束后，输出 $\boldsymbol{R}_{\text{out}}$。然后再利用式(15)计算 $\boldsymbol{t}_{\text{out}}$，但是因为一开始对参考点进行了均值化操作，所以最后还需要对 $\boldsymbol{t}$ 平移一下，最终输出 $\boldsymbol{t}$，如式(15)所示：

$$\boldsymbol{t}_{\text{out}}=\boldsymbol{t}_{\text{out}}-\boldsymbol{R}_{\text{out}}\overline{\boldsymbol{P}} \tag{15}$$

对加权目标函数进行拉直得到：

$$\text{vec}(E(\boldsymbol{R},\boldsymbol{t}))=\text{vec}\Big(\sum_{i=1}^{n}\omega_i\|(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)(\boldsymbol{R}\boldsymbol{P}_i+\boldsymbol{t})\|^2\Big)=$$

$$\sum_{i=1}^{n}\omega_i\|(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)(\boldsymbol{P}_i^{\mathrm{T}}\otimes\boldsymbol{I}+\boldsymbol{G})\boldsymbol{r}\|^2=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{C}_{9\times9} \tag{16}$$

式中，$\boldsymbol{C}_{9\times9}=\sum_{i=1}^{n}\omega_i^2(\boldsymbol{P}_i\otimes\boldsymbol{I}+\boldsymbol{G}^{\mathrm{T}})(\boldsymbol{I}-\hat{\boldsymbol{V}}_i)(\boldsymbol{P}_i^{\mathrm{T}}\otimes\boldsymbol{I}+\boldsymbol{G})$。

在刚开始迭代时，就先计算好 $\boldsymbol{G}$ 矩阵，然后计算 $\boldsymbol{B}$ 矩阵和 $\boldsymbol{C}$ 矩阵。然后利用 $\boldsymbol{B}$ 和 $\boldsymbol{C}$，通过式(13)迭代求解 $\boldsymbol{R}$，再利用式(16)计算目标函数，每次迭代的计算复杂度为 $\boldsymbol{O}(1)$，从而可以大幅度减少迭代过程中的计算量。迭代结束后通过(15)式求出 $\boldsymbol{t}$ 即可。而且不难发现，计算矩阵 $\boldsymbol{G}$，$\boldsymbol{B}$ 和 $\boldsymbol{C}$ 的过程，计算复杂度均为 $\boldsymbol{O}(n)$。

1.2.3 收敛性证明

为验证加权的正交迭代算法的收敛性，设第 $k+1$ 次的误差为 $E(\boldsymbol{R}^{(k+1)})$，则 $E(\boldsymbol{R}^{(k+1)})$需满足：

$$E(\boldsymbol{R}^{(k+1)})=\sum_{i=1}^{n}\omega_i\|\boldsymbol{x}_i^{(k+1)}-\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k+1)}\|^2=\sum_{i=1}^{n}\omega_i\cdot$$

$$\|\boldsymbol{x}_i^{(k+1)}-\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k)}\|^2-\sum_{i=1}^{n}\omega_i\|\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k+1)}-\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k)}\|^2 \tag{17}$$

$$\sum_{i=1}^{n}\omega_i\|\boldsymbol{x}_i^{(k+1)}-\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k)}\|^2\leqslant\sum_{i=1}^{n}\omega_i\cdot$$

$$\|\boldsymbol{x}_i^{(k)}-\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k)}\|^2=E(\boldsymbol{R}^{(k)}) \tag{18}$$

$$E(\boldsymbol{R}^{(k+1)})\leqslant E(\boldsymbol{R}^{(k)})-\sum_{i=1}^{n}\omega_i\|\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k+1)}-\hat{\boldsymbol{V}}_i\boldsymbol{x}_i^{(k)}\|^2 \tag{19}$$

式(17)中，$\boldsymbol{x}_i$ 是 $\boldsymbol{X}_i$ 的视线投影向量，由式(19)可知，等式右侧减数一项肯定不为零，说明在迭代停止前，$E(\boldsymbol{R}^{(k+1)})\leqslant E(\boldsymbol{R}^{(k)})$ 恒成立，满足全局收敛，表明改进的加权方法有效。

1.2.4 权值的确定

观察式(4)，加权正交迭代算法的求解过程就是一个如式(20)所示的加权优化问题：

$$\sum_{i=1}^{n}\omega_i(\varepsilon_i)^2 \tag{20}$$

而对于如式(20)的加权优化问题的权重估计经常会用到 M 估计方法[15]，它是指在粗差不可避免的情况下，通过给粗差分配不同的权重，最大程度地减少粗差对结果的影响。前面分析了正交迭代算法存在的误差问题，这些未知误差的存在就是相机位姿估计中的粗差，可以使用 M 估计方法进行稳健估计，以减轻测量误差带来的影响。

M 估计是基于最小二乘估计发展出来的稳健性估计方法，最小二乘估计是对所有样本点赋予一样的权值，这会导致混入正常样本中的异常样本值对估计结果造成影响。为了解决这个问题引入了 M 估计方法。

由文献[15]我们可以得到 M 估计的权值为

$$\omega_i=\begin{cases}\left[1-\left(\dfrac{u_i}{4.685}\right)^2\right]^2, & |u_i|\leqslant4.685\\ 0, & |u_i|>4.685\end{cases} \tag{21}$$

式中，u_i 由各参考点的像方残差进行归一化得到，定义 $\sigma=\dfrac{MAD}{0.6745}=\dfrac{\text{median}\,|e_i-\text{median}(e_i)|}{0.6745}$，其中 e_i 为第 i 个参考点的像方残差，则 $u_i=\dfrac{e_i}{\sigma}$，阈值 4.685 是文献[15]根据 Tukey 的双权数函数给定的。

图 2 给出了权重 ω_i 随 u_i 的变化曲线，从图中可以看出，归一化残差绝对值越大，表示越有可能是异常值，赋予的权值越小。当归一化残差绝对值大于 4.685 的时候，就认为该次的测量值为野值，赋予权值为 0，表示该测量值被排除，进而排除该参考点对位置参数估计的影响。

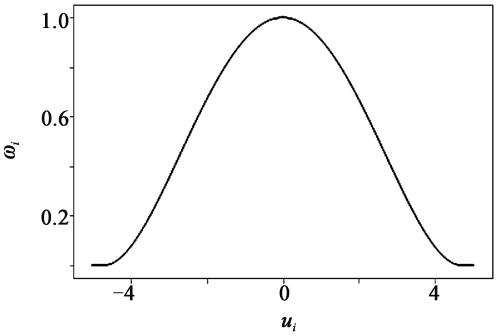

图 2 权重变化曲线

2 仿真实验

为了验证提出的加权正交迭代算法的有效性和优越性，在计算精度上，将其与原正交迭代算法及其他的一些相机位姿估计算法进行比较。

仿真实验环境为 Matlab R2018b，使用分辨率为 640 pixel×480 pixel 的虚拟相机，该相机焦距为 $f=800$，相机坐标系下的参考点均匀分布在[−2，2] m×[−2，2] m×[4，8] m 的长方体内。随机生成旋转矩阵 $\boldsymbol{R}_{\text{true}}$ 和平移向量 $\boldsymbol{t}_{\text{true}}$，利用该矩阵和向量将 $\boldsymbol{C}_i$ 逆刚体变换到世界坐标系下生成 $\boldsymbol{P}_i$，不考虑虚拟相机的畸变，通过给定内参将 $\boldsymbol{C}_i$ 投射到图像坐标系下生成 $\boldsymbol{p}_i$，为了测定算法的鲁棒性，根据实验条件随机旋转部分图像点加入不同等级的高斯白噪声。通过算法计算得到的相机位姿分别记为旋转矩阵 $\boldsymbol{R}$ 和平移向量 $\boldsymbol{t}$，定义旋转误差 e_{rot} 和平移误差 e_{trans}，如式(22)所示：

$$e_{\text{rot}}=\max_{i=1}^{3}\cos^{-1}[\text{dot}(\boldsymbol{r}_{\text{true}}^{k},\boldsymbol{r}^{k})]\times\frac{180}{\pi}$$

$$e_{\text{trans}}=\frac{\|\boldsymbol{t}_{\text{true}}-\boldsymbol{t}\|}{\|\boldsymbol{t}_{\text{true}}\|}\times 100 \tag{22}$$

式中，$\boldsymbol{r}_{\text{true}}^{k}$ 和 $\boldsymbol{r}^{k}$ 分别表示 $\boldsymbol{R}_{\text{true}}$ 和 $\boldsymbol{R}$ 的第 k 列。

参与比较的其他相机位姿估计算法包括：LHM 算法、EPnP 算法、DLT 算法、RPnP 算法，这其中既包括原始的 LHM 算法，与之比较以证实效果是否得以改进，还包括线性求解方法的代表 DLT 算法、EPnP 算法以及 RPnP 算法。

2.1　参考点数目变化对算法的影响

改变输入参考点的数目为 4～15，来验证各算法的求解精度，仿真实验结果如图 3～6 所示，通过分析可以得出以下结论：

(1)DLT 和 EPnP 算法作为线性求解方法，整体误差都大于其他三种非线性迭代算法；

(2)传统 LHM 作为非线性迭代算法，在点数较少的时候容易陷入局部最优，因此误差较大，而在参考点较多的时候，能利用的信息增加了，所以也能够得到全局最优解；

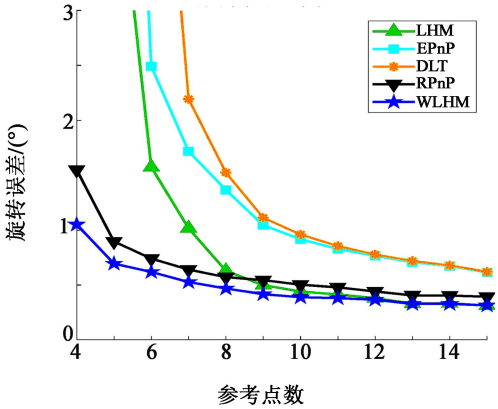

图 3　参考点数变化时的旋转平均误差

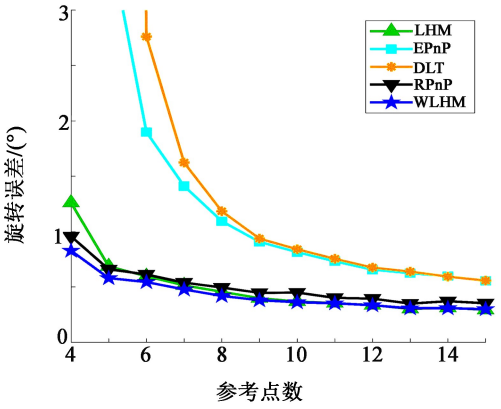

图 4　参考点数变化时的旋转中值误差

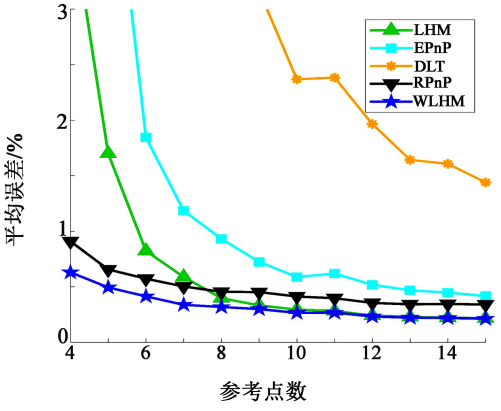

图 5　参考点数变化时的平移平均误差

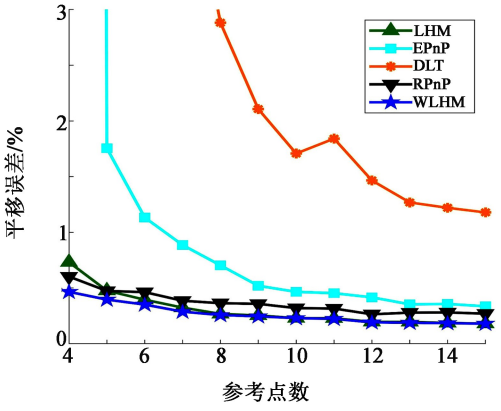

图 6　参考点数变化时的平移中值误差

(3)RPnP 算法与改进后的 LHM 算法效果持平，无论点多点少，求解精度都很高且很稳定。

2.2　参考点噪声对算法的影响

固定参考点数目为 10，分别给各参考点添加方差 0～5 pixels，步长为 0.5 pixels 的白噪声来验证参考点噪声对算法精度的影响，实验结果如图 7～10 所示。分析图 7～10 可得出以下结论：

1)总地来看，各算法的精度都随噪声的增加而降低，而且两者之间近似线性相关；

2)DLT 和 EPnP 比其他三种非线性迭代算法的

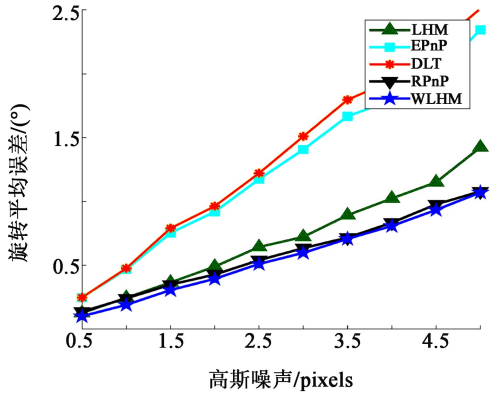

图 7 不同噪声水平下的旋转平均误差

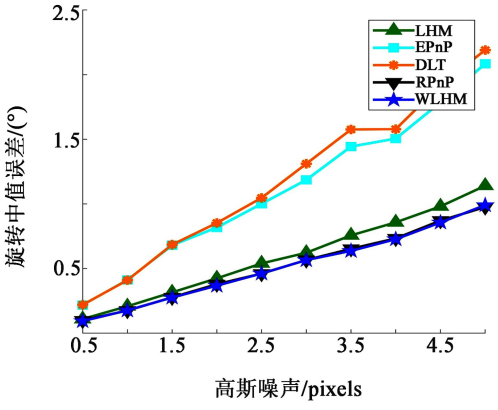

图 8 不同噪声水平下的旋转中值误差

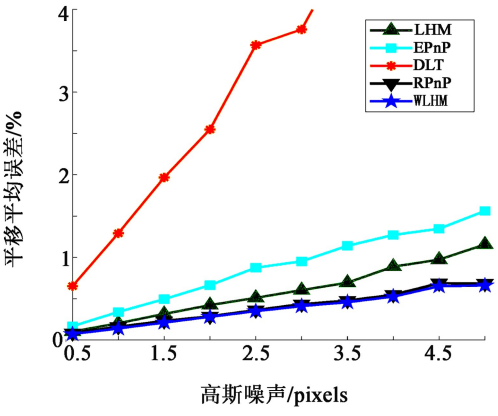

图 9 不同噪声水平下的平移平均误差

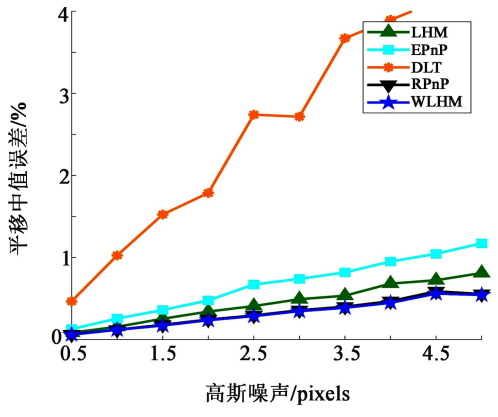

图 10 不同噪声水平下的平移中值误差

误差大，而且误差随噪声的变化率也比其他算法大；

3）RPnP 算法和改进后的 LHM 算法受误差的影响差不多，求解精度相似；

4）通过四个图可以发现改进后的 LHM 算法相对于原始的 LHM 算法有所改善，说明通过对参考点进行加权，针对不同误差大小的参考点赋予不同的权值，可以提高求解精度，对噪声的鲁棒性也优于原算法。

2.3 算法运行效率对比

在 4～100 范围内改变参考点数目，在不同参考点数目下重复实验 500 次，然后统计各算法的平均运行时间，结果如图 11 所示。

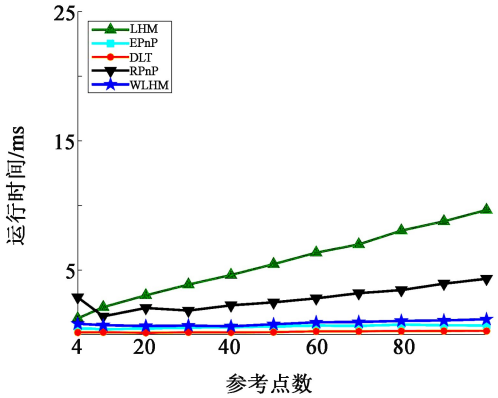

图 11 所有算法的平均运行时间

从图 11 不难看出：DLT 和 EPnP 算法作为线性算法的运行时间明显短于其他算法，LHM 算法作为迭代算法，效率最低，RPnP 算法效率次之，而改进后的 LHM 算法，运行时间明显比原始的 LHM 算法短了很多，与线性求解的 DLT 和 EPnP 接近，表明对算法的加速操作产生了效果。

3 结语

针对现有相机位姿估计算法抗野值点能力较差的问题，提出了基于测量误差不确定性加权的快速相机位姿估计方法。该方法在经典正交迭代算法的基础上根据参考点测量误差的大小分配给目标函数相应大小的权重，权重由 M 估计方法给出，能最大程度地将测量误差较大的野值点排除，提高算法的求解精度，同时引入 Kronecker 积对算法进行加速，将迭代计算过程中重复计算的常量提前计算好，减少了其中的冗余计算从而提高运算速度。设计仿真对比实验与多种算法进行对比，实验结果表明改进算法提高了正交迭代算法的抗野值点能力，且算法

速度更快。

参考文献：

[1] Tahri O, Chaumette F. Complex objects pose estimation based on image moment invariants[C]//IEEE Proc. of the 2005 IEEE Inter. Conf. on Robotics and Automation, 2005: 436-441.

[2] Caron G, Dame A, Marchand E. Direct model based visual tracking and pose estimation using mutual information[J]. Image and Vision Computing, 2014, 32(1): 54-63.

[3] Azuma R, Baillot Y, Behringer R, et al. Recent advances in augmented reality [J]. IEEE Computer Graphics and Applications, 2001, 21(6): 34-47.

[4] Ke T, Roumeliotis S I. An efficient algebraic solution to the perspective-three-point problem[C]//Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 2017: 7225-7233.

[5] 吴凡路，刘建军，任　鑫，等. 基于圆形标志点的深空探测全景相机标定方法[J]. 光学学报，2013，33(11)：139-145.
Wu F L, Liu J J, Ren X, et al. Deep space exploration panoramic camera calibration technique based on circular marker[J]. Acta Optica Sinica, 2013, 33(11): 139-145.

[6] Fischler M A, Bolles R C. 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.

[7] 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.

[8] Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate $O(n)$ solution to the PnP problem[J]. Inter. J. of Computer Vision, 2009, 81(2): 155.

[9] Li S, Xu C, Xie M. A robust $O(n)$ solution to the perspective-n-point problem[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1444-1450.

[10] Gao X S, Hou X R, Tang J, et al. Complete solution classification for the perspective-three-point problem [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2003, 25(8): 930-943.

[11] Hesch J A, Roumeliotis S I. A direct least-squares (DLS) method for PnP[C]// IEEE Inter. Conf. on Computer Vision, 2011: 383-390.

[12] Lu C P, Hager G D, Mjolsness E. Fast and globally convergent pose estimation from video images[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000, 22(6): 610-622.

[13] 杨　森，吴福朝. 摄像机位姿的加权线性算法[J]. 软件学报，2011，22(10)：2476-2487.
Yang S, Wu F C. Weighted linear methods for the camera pose estimation[J]. J. of Software, 2011, 22(10): 2476-2487.

[14] 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)：258-265.
Li Xin, Long G C, Liu J B, et al. Accelerative orthogonal iteration algorithm for camera pose estimation[J]. Acta Optica Sinica, 2015, 35(1): 258-265.

[15] Susanti Y, Pratiwi H. M estimation, S estimation, and MM estimation in robust regression[J]. Inter. J. of Pure and Appl. Mathematics, 2014, 91(3): 349-360.

作者简介：

徐　浩(1994—)，男，湖北省黄冈市人，硕士，主要研究方向为增强现实、虚拟现实与军用仿真；

张秀山(1968—)，男，湖北省洪湖市人，副教授，硕士生导师，主要研究方向为虚拟现实与军用仿真。