

## 基于遗传算法的摄像机自标定方法

郭秋艳<sup>1)</sup> 刘鹏飞<sup>1)</sup> 安平<sup>1)</sup> 张兆杨<sup>1),2)</sup>

<sup>1)</sup>(上海大学通信与信息工程学院,上海 200072) <sup>2)</sup>(上海大学新型显示技术及应用集成教育部重点实验室,上海 200072)

**摘要** 摄像机标定是计算机视觉领域的关键技术,其中的自标定是只根据图像计算摄像机的内参数,其标定过程简单,适用性强。由于传统的用于摄像机自标定的 Kruppa 方程不仅需要计算基础矩阵,还要计算图像的极点,而图像的极点又不是固定不变的,且会导致计算结果的不稳定,为此,针对传统摄像机自标定方法的上述不足,利用遗传算法完成了 Hartley 新的 Kruppa 方程的摄像机自标定过程,以便将这个过程完全转化为通过代价函数最小化来求得摄像机的内参数,这就排除了极点的不稳定因素。实验结果表明,该方法是简单、有效的,可以作为一种通用的标定工具。

**关键词** 摄像机自标定 Kruppa 方程 基础矩阵 遗传算法

**中图分类号**: TP391.4 **文献标识码**: A **文章编号**: 1006-8961(2006)11-1712-04

### Self-calibration for Cameras Based on Genetic Algorithm

GUO Qiu-yan<sup>1)</sup>, LIU Peng-fei<sup>1)</sup>, AN Ping<sup>1)</sup>, ZHANG Zhao-yang<sup>1),2)</sup>

<sup>1)</sup>(School of Communication and Information Engineering, Shanghai University, Shanghai 200072)

<sup>2)</sup>(Key Laboratory of Advanced Displays and System Application, Ministry of Education, Shanghai 200072)

**Abstract** Camera calibration is a key technology in computer vision, in which camera self-calibration technology is to compute the camera's intrinsic parameters from a series of images. Compared to traditional camera calibration methods, the process of self-calibration is simpler and more convenient for application. The self-calibration technology using Kruppa equation not only requires computing the fundamental matrix, but also computing the epipoles of images which are variable with the different images and will result in unstable computation results. GA algorithm is used to complete the self-calibration processing by estimating the new and simple Kruppa equations defined by Hartley. At last, the self-calibration problem is converted into the minimization of the cost function, so that the epipoles instability is eliminated and the calibration effect is improved. Experimental results show that the proposed method is simple and effective, and can become a versatile tool for camera calibration.

**Keywords** camera self-calibration, Kruppa equations, fundamental matrix, genetic algorithm

## 1 引言

摄像机标定技术是计算机视觉领域里的一项关键技术,它是从 2 维图像提取 3 维空间信息必不可少的步骤,这项技术已被广泛应用于 3 维结构重建、导航、视觉监控等领域。一般来讲,摄像机标定分为:传统的模板标定方法、基于主动视觉的标定方法、自标定方法 3 种。其中传统的标定方法需要使

用标定模板,并通过模板上已知的 3 维坐标点与其图像对应点的对应来得到摄像机的变换矩阵及内外参数,其主要的特点就是精度较高;基于主动视觉的标定方法需要严格控制摄像机的运动,利用这种已知的运动信息来获得摄像机的参数,虽计算简单,但是不能普遍应用,也不适合摄像机的运动未知且无法控制其运动的情况;自标定虽是最适合应用,但也是算法上最难实现,而且获得的精度相对较低的一种标定方法,就其过程而言,其进行的摄像机自标定

**基金项目**:上海市自然科学基金项目(04ZR14056);上海市教委发展基金项目(04AB56);国家自然科学基金项目(60572127)

**收稿日期**:2006-07-03;**改回日期**:2006-08-06

**第一作者简介**:郭秋艳(1982~),女。2005年获河北师范大学学士学位,现为上海大学信号与信息处理专业硕士研究生。研究方向为 3 维视频信号处理。E-mail: gqy820129@163.com

不利用场景和摄像机的运动信息,而只是根据拍摄得到的 2 维图像之间存在的几何关系来获得摄像机的内参数,并操作过程简单。

## 2 自标定算法及 Kruppa 方程

### 2.1 自标定算法

摄像机的自标定就是仅通过图像来计算其内参数。传统的摄像机自标定方法一般是对于单一运动的摄像机,它是摄像机内参数是固定不变且未知,也不知其外部运动信息的条件下,通过求解 Kruppa 方程来求得内参数。近年来又出现了一些新的自标定方法,其中分层重建的摄像机自标定算法是将仿射重构改进到一个欧氏空间,进而求解其内参数。新方法中还包括基于绝对二次曲线和绝对二次曲面的摄像机自标定、Pollefeys 的模约束等等,现在还开始了摄像机的内参数是变化的情况下自标定方法的研究。

### 2.2 Kruppa 方程

表征摄像机与拍摄的图像之间关系的 Kruppa 方程<sup>[1]</sup>如下:

$$(e \times m)^T K K^T (e \times m) = m^T [e]_x K K^T [e]_x m \Leftrightarrow$$

$$(Fm)^T K K^T (Fm) = m^T F^T K K^T Fm$$

$$\forall m, \text{有} [e]_x^T K K^T [e]_x \cong F^T K K^T F \quad (1)$$

其中,  $[ ]_x$  表示矢量的反对称矩阵,  $e$  为图像极点(摄像机两焦点的连线和图像平面的交点),  $m$  表示极线与绝对二次曲线的切点,  $K$  为摄像机内参数矩阵,  $F$  为两幅图像之间的几何关系,称之为基础矩阵,其与场景结构无关。图像的极点是未知的,它是随着图像的变化而变化。Hartley 提出一种不需计算极点求解 Kruppa 方程的新方法<sup>[2,3]</sup>,该方法只根据基础矩阵与摄像机内参数之间的关系求解,其主要的两部分工作就是计算两幅图像之间的几何关系,即基础矩阵  $F$ ,以及与基础矩阵有关的求解内参数的数学算法。

## 3 算法原理

### 3.1 基础矩阵 $F$

基础矩阵是双视点间的几何关系,在双视点结构下,图像点间的对应点遵循众所周知的对极几何关系,称之为极线约束,其可表示为:  $\hat{m}^T Fm = 0$ ,式中,  $F$  是秩为 2 的  $3 \times 3$  基础矩阵(fundamental matrix),  $F$  有两个非零的特征值。由于基础矩阵包含了两幅图像间的所有几何信息,因此精确计算  $F$

对于摄像机标定或者 3 维重建具有重要的意义。

### 3.2 计算基础矩阵 $F$

对于基础矩阵的求解,常用的计算方法是八点算法,其原理如下:

设  $m = [u \ v \ 1]^T$ ,  $\hat{m} = [\hat{u} \ \hat{v} \ 1]^T$ , 并将  $F$  写成由各分量组成的列向量形式,即

$$\hat{F} = [f_{1,1} \ f_{1,2} \ f_{1,3} \ f_{2,1} \ f_{2,2} \ f_{2,3} \ f_{3,1} \ f_{3,2} \ f_{3,3}]^T$$

则对应每一对应点对  $(m, \hat{m})$ , 可写出方程  $[u\hat{u} \ v\hat{v} \ \hat{u}\hat{u} \ \hat{u}\hat{v} \ \hat{v}\hat{v} \ \hat{v}u \ v \ 1]^T \hat{F} = 0$ 。将  $n(n \geq 8)$  个这样的方程叠加起来即得到线性方程组  $A\hat{F} = 0$ , 再利用 SVD(singular value decomposition) 分解就可求出该方程组的最小二乘解。八点算法虽简单易行,但是当图像有噪声时,不太容易得到精确解。

针对八点算法的缺点,本文采用了 RANSAC (RANdom Sample Consensus) 算法<sup>[4]</sup> 求解基础矩阵,从而使得到的值更加精确,计算时间也大为减少。该算法是,首先输入用于标定的左右两幅立体对图像,并在两幅图像中分别进行角点检测;然后进行特征点的寻找和匹配点的确定。虽然其在求解基础矩阵的过程中需要进行迭代,但是耗时并不长。

### 3.3 结合基础矩阵的 Kruppa 方程新形式

#### 3.3.1 预备知识

首先要定义绝对二次曲线。在欧氏空间,绝对二次曲线位于无穷远平面上,有如下的方程:

$$x^2 + y^2 + z^2 = 0 \quad (2)$$

则绝对二次曲线上的点满足  $M^T M = 0$ 。摄像机的投影矩阵为

$$P = K[R | -Rt] \quad (3)$$

其中,  $R, t$  分别为世界坐标系到摄像机坐标系的旋转矩阵和平移向量,被称为摄像机外参数。

由绝对二次曲线上的 3 维点  $x$  投影的 2 维点可以表示为

$$m = P(M) = KRM \quad (4)$$

其中,  $M = R^T K^{-1} m$ 。因为  $R$  为正交矩阵,故由以上的式子可以推导出

$$m^T K^{-1} R R^T K^{-1} m = m^T K^{-1} K^{-1} m = 0 \quad (5)$$

这表明任意 2 维图像点  $m$  均在绝对二次曲线的像上,并且绝对二次曲线的像  $w$  可由矩阵  $K^{-T} K^{-1}$  来表示。对  $w$  取逆可得到  $K K^T$ , 并记为  $C = K K^T$ , 然后通过 Cholesky 分解即可求出摄像机的内参数矩阵  $K$ <sup>[1]</sup>。

#### 3.3.2 基础矩阵与 Kruppa 方程

Kruppa 方程与基础矩阵有着密切的联系,由

Kruppa 方程的最初形式已经看到, Kruppa 方程的计算还需要不稳定的极点。Hartley 提出了一种新的 Kruppa 方程形式<sup>[2,3]</sup>, 其不需要计算图像的极点, 而是采用直接由基础矩阵推导, 得到了 Kruppa 方程的简化形式:

$$\frac{v_2^T C v_2}{r^2 u_1^T C u_1} = \frac{-v_2^T C v_1}{s r u_1^T C u_1} = \frac{v_1^T C v_1}{s^2 u_2^T C u_2} \quad (6)$$

其中,  $F = U D V^T$ , 通过 SVD 分解可得出矩阵  $U, D, V$ ;  $u_1, u_2, u_3$  分别为  $U$  的列向量;  $v_1, v_2, v_3$  分别为  $V$  的列向量;  $r$  和  $s$  为对角矩阵  $D$  的其中两个特征值。只要找到  $C$  (主要是  $K$  的元素) 使得这 3 个式子成立或近似相等, 那么就能完成自标定工作, 记 3 个等式分别标记为

$$f_1 = \frac{v_2^T C v_2}{r^2 u_1^T C u_1}, f_2 = \frac{-v_2^T C v_1}{s r u_1^T C u_1}, f_3 = \frac{v_1^T C v_1}{s^2 u_2^T C u_2}$$

并通过将 Kruppa 方程转化成代价函数来完成自标定工作, 这个代价函数可以用于一对图像, 也可以用于多对图像的标定工作。

对于一对图像的标定工作, 记代价函数  $sum$  为

$$sum = (f_1 - f_2)^2 + (f_1 - f_3)^2 + (f_2 - f_3)^2 \quad (7)$$

并使得这个代价函数值最小, 此时的  $C$  ( $K$  中的元素) 即为所要求的值。在这里记摄像机的内参数矩

$$阵为 K = \begin{bmatrix} k_u & 0 & p_u \\ 0 & k_v & p_v \\ 0 & 0 & 1 \end{bmatrix}, 4 \text{ 个参数均是在像素量级上}$$

定义, 则:

$$C = K K^T = \begin{bmatrix} k_u^2 + p_u^2 & p_u p_v & p_u \\ p_u p_v & k_v^2 + p_v^2 & p_v \\ p_u & p_v & 1 \end{bmatrix} \quad (8)$$

其中,  $k_u, k_v$  指横纵放大倍数,  $p_u, p_v$  指主点坐标。记  $a = k_u^2 + p_u^2, b = p_u p_v, c = p_u, d = k_v^2 + p_v^2, e = p_v,$

$$v_1 = \begin{bmatrix} v_{1,1} \\ v_{1,2} \\ v_{1,3} \end{bmatrix}, v_2 = \begin{bmatrix} v_{2,1} \\ v_{2,2} \\ v_{2,3} \end{bmatrix}, v_3 = \begin{bmatrix} v_{3,1} \\ v_{3,2} \\ v_{3,3} \end{bmatrix}, v_{i,j} \text{ 为上一节计算}$$

的  $F$  的 SVD 分解可得出的具体数值。同理,  $u_i$  也如上述表示, 将以上矩阵、向量中的元素代入  $f_1, f_2, f_3$  即得到如下 3 个具体的数值型式子如式 (9) 所示: 将式 (9) 代入代价函数 (式 (7)) 即可求得  $sum$ :  $sum = (f_1 - f_2)^2 + (f_1 - f_3)^2 + (f_2 - f_3)^2$  就转化为数值型, 这个代价函数中有 4 个参数  $k_u, k_v, p_u, p_v$ , 可通过使代价函数最小 (或接近于 0) 来求得各参数。本文采用遗传算法来完成最小值的求解。

$$f_1 = \frac{a v_{21}^2 + d v_{22}^2 + v_{23}^2 + 2(b v_{21} v_{22} + c v_{21} v_{23} + e v_{21} v_{23})}{r^2 [(a u_{11}^2 + d u_{12}^2 + u_{13}^2) + 2(b u_{11} u_{12} + c u_{11} u_{13} + e u_{12} u_{13})]}$$

$$f_2 = \frac{a v_{21}^2 + d v_{22}^2 + v_{23}^2 + 2(b v_{21} v_{22} + c v_{21} v_{23} + e v_{22} v_{23})}{s r [(a u_{11}^2 + d u_{12}^2 + u_{13}^2) + 2(b u_{11} u_{12} + c u_{11} u_{13} + e u_{12} u_{13})]}$$

$$f_3 = \frac{a v_{11}^2 + d v_{12}^2 + v_{13}^2 + 2(v_{11} v_{12} + c v_{11} v_{13} + e v_{12} v_{13})}{s^2 [(a u_{21}^2 + d u_{22}^2 + u_{23}^2) + 2(b u_{21} u_{22} + c u_{21} u_{23} + e u_{22} u_{23})]} \quad (9)$$

### 4 遗传算法

遗传算法本身就是模拟人类的遗传进化, 定义每个解空间变量为一个染色体, 并定义适当的交叉和变异<sup>[5]</sup>。遗传算法用于不同的问题需要进行具体的调试和量的定义, 本文将遗传算法的思想用于上述的最小值求解。具体的输入参数为:  $bounds$  (边界约束条件, 即 4 个参数的区域范围); 目标函数  $f(x)$ ; 初始种群的数值  $num$ ; 最大迭代次数  $N$ ; 交叉概率  $P_c$ ; 突变概率  $P_m$ 。输出参数为: 目标最优解为  $f^{best}$ ; 最优解向量  $x$  (即所要求解的代价函数最小的 4 个参数的值,  $x = (k_u, k_v, p_u, p_v)$ )。其算法流程如图 1 所示。

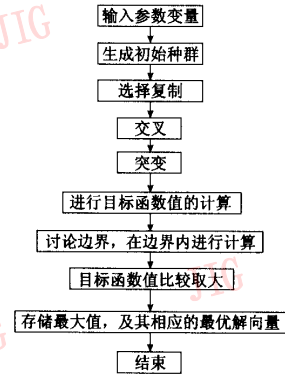


图 1 遗传算法的简单过程

Fig. 1 The process of Genetic Algorithm

在算法中第 1 步将求解代价函数最小的问题转化为求解  $-f(x) = -sum$  最大的问题, 其中输入变量就是上述的 5 个因子; 第 2 步生成初始种群, 并将最后一列作为最大值, 即目标函数的初始值; 第 3 步选择复制, 即模拟人类的繁衍, 复制函数采用小盘轮转法, 并存储概率大的目标函数值; 第 4 步交叉, 即采用点交叉及二进制编码来选优; 第 5 步突变, 对基因突变函数应用二进制编码; 第 6 步将遗传算法衍

生下来的下一代用于目标函数值的计算,如果子代的值没有超过给定的边界条件,则将生成的目标函数值与最初设定的值进行比较,取较大值,最后将所有比较所得到的最大值以及相应的最优解向量存储输出。由此可见,整个过程就是求解得到代价最小的 4 个参数值的优化过程。

## 5 实验结果及分析

对于图 2 所示的两对立体图像对,在 Matlab 7.0.1 平台上用本文算法完成自标定过程。本文算法的边界条件为: $\text{bounds} = [500 \ 1000; 500 \ 1000; -100 \ 400; -100 \ 300]$ ,其余遗传因子分别为以下两种情况:

(1) 交叉概率  $P_c = 0.1$ , 变异概率  $P_m = 0.01$ , 迭代次数  $N = 500$ , 种群个数为 50。

(2) 交叉概率  $P_c = 0.25$ , 变异概率  $P_m = 0.01$ , 迭代次数  $N = 1000$ , 种群个数为 100。

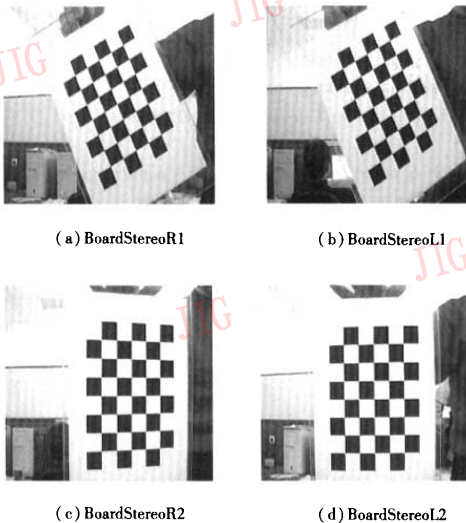


图 2 实验用两对立体图对  
Fig. 2 Two pairs of stereo images

为进行比较,本文还基于 Intel 的 OpenCV 函数库对摄像机进行了标定。对于以上两对立体图对,本文算法标定结果与基于 OpenCV 函数库的传统方法标定得到的摄像机内参数值比较如表 1 所示。由表 1 可见,结果虽有一定的误差,但相对于文献[3]中提到的其他方法的 24.9%、最坏 50% 的误差,本文的结果相对较好。

由于迭代结果有一定的误差,因此需调整各个

表 1 实验结果与传统方法对比

Tab. 1 Test results compared with traditional method

标定法	$k_u$	$k_v$	$p_u$	$p_v$
传统 (OpenCV)	798.2	798.2	349	199
本算法 (针对图 2(a), (b))	829	809	362	214
本算法 (针对图 2(c), (d))	809	822	366	248
误差比较 (1) %	3.63	1.25	3.6	7.54
误差比较 (2) %	1.25	5.51	4.8	24

因子,以得到更好的结果。当所得到的结果偏差较大时,需多运行几次,然后求得到的各参数值的平均值,就可以得到更可靠的结果。另外边界条件对标定的结果也有影响,因此需调整好边界。

## 6 结论

作为简单有效的自标定算法,本文算法不需计算不稳定的图像极点,而只需计算两幅图像之间或者几幅图像之间的几何关系,也就是基础矩阵,进而通过 SVD 分解的向量值,即可将整个过程中完全转化为数学求极值问题,这就简化了自标定的求解过程。由于标定结果受边界条件的影响稍大,因此需谨慎考虑边界区域的划分。由于本文算法具有自适应性,且实现过程简单,因此对于难于完成基于标定物标定的场合有实用价值。笔者今后将在建立多视点实时采集处理系统的过程中应用本算法进行摄像机阵列的校准,并做进一步的优化。

### 参考文献 (References)

- Meng Xiao-qiao, Hu Zhan-yi. Recent progress in camera self-calibration[J]. Acta Automatic Sinica, 2003, 29(1): 110 ~ 123. [孟晓桥, 胡占义. 摄像机自标定方法的研究与进展[J]. 自动化学报, 2003, 29(1): 110 ~ 123.]
- Hartley Richard I. Kruppa's equations derived from the fundamental matrix[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(2): 133 ~ 135.
- Whitehead Anthony, Roth Gerhard. Estimating intrinsic camera parameters from the fundamental matrix using an evolutionary approach[J]. EURASIP Journal on Applied Signal Processing, 2004, 8: 1113 ~ 1124.
- Fusiello Andrea. Appunti di visione computazionale tecniche di basso livello[EB/OL]. <http://www.sci.univr.it/~fusiello>. Maggio, 2001
- Lobo Fernando G, Sastry Kumara, Harik Georges R. Extended compact genetic algorithm in c++: version 1.1[R]. IlliGAL Report 2006012, 2006. Urbana: Illinois, USA Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.