一种基于计算机视觉的飞行器姿态估计算法

赵世峰 张 海 范耀祖

(北京航空航天大学 自动化科学与电气工程学院, 北京 100083)

摘 要: 改进了一种基于图像统计信息的飞行器姿态估计算法. 改进后的算法可以迅速地估计出飞行器的横滚角与俯仰角,减少了运算量,提高了姿态估计的实时性和鲁棒性,精度满足飞行器的姿态控制要求. 将不同的算法进行了仿真比较,提出了一种应用方案,解决了姿态估计过程中横滚角与俯仰角耦合的问题. 改进后的算法适用于装备以视觉系统作为导航系统的飞行器,尤其是只能装备小型视觉系统的微小型飞行器(MAV, Micro Air Vehicles). 将应用该算法的视觉导航系统与微惯性测量单元(MIMU, Miniature Inertial Measurement Unit)组合使用,可以增大飞行器姿态角估计的范围,进一步提高估计精度.

关 键 词: 姿态估计; 图像处理; 计算机视觉; 飞行器

中图分类号: V 249.122^{+}.2

文献标识码: A **文 章 编 号**: 1001-5965(2006)08-0885-04

Attitude estimation method for flight vehicles based on computer vision

Zhao Shifeng Zhang Hai Fan Yaozu

(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: An attitude estimation method for flight vehicles based on statistic of image was improved. The method improved reduces calculation, and improves the real-time performance for attitude estimation. The bank and the pitch angle can be estimated rapidly with a higher robusticity. Its accuracy satisfies the requirement of attitude estimation to control flight vehicles. Several different methods were compared with simulation results, and an application was proposed, which resolved the problem of coupling between the bank and the pitch in the attitude estimation. The method improved is applicable for those which are equipped computer vision systems as navigation systems, especially for MAV(micro air vehicles) equipped only minitype computer vision systems. Range of the attitude angle to be estimated is expanded while computer vision system equipped by the method is adopted with a MIMU(miniature inertial measurement unit) assistant, and the accuracy is higher correspondingly.

Key words: attitude estimation; image processing; computer vision; flight vehicles

视觉系统具有独立性、准确性、可靠性以及信息完整性等优势. 利用视觉系统控制飞行器的姿态是近年发展起来的一种先进的定姿方法,其利用摄像机摄取图像信息,经图像处理提取并识别相关信息,估计出飞行器在环境中的姿态,进而控制飞行器飞行.

飞行器,尤其是无人飞行器,在飞行过程中,需要姿态稳定系统辅助工作. 目前,很多飞行器都装有视觉系统采集图像,用于侦察、勘探. 如果能利用自带的成像系统进行飞行器的飞行姿态估计,势必能提高设备的利用率. 近年来,微小型飞行器(MAV, Micro Air Vehicles)发展迅速. 由于 MAV 的体积与负载能力的限制,无法安装复杂的、体积和重量都相对较大的导航设备,如果在 MAV 上安装小型的视觉系统,利用视觉进行姿态估计,对于提高 MAV 的飞行稳定性、自主飞行能力、战场生存能力有很大帮助.

本文改进了一种图像处理方法——地平线算

收稿日期: 2005-10-18

作者简介: 赵世峰(1979-), 男, 黑龙江牡丹江人, 博士生, zhaoshifeng@asee.buaa.edu.cn.

法,并将其应用于飞行器的姿态估计. 仿真结果表明,这种算法的计算量小,实时性好,精度能满足飞行器姿态控制的需要.

1 地平线算法的提出

建立地面坐标系 $O_d x_d y_d z_d$ 与机体坐标系 $O_t x_t y_t z_t$,如图 1 所示.

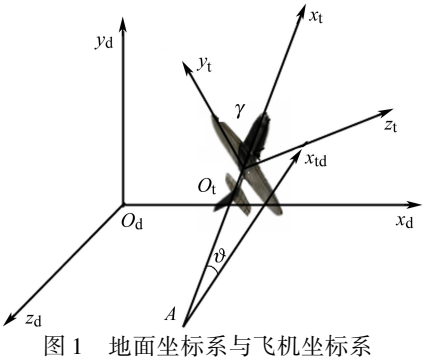


图 1 地面坐标系与飞机坐标系

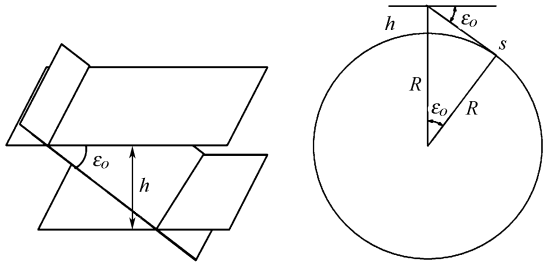
O_d 为地平线任一点;$O_d x_d$ 与地平线重合;$O_d y_d$ 轴垂直地面向上;$O_d z_d$ 轴垂直于 $x_d O_d y_d$ 平面. O_t 取在飞机重心处;$O_t x_t$ 轴处于飞机对称平面内,与机身轴线一致,指向前方;$O_t y_t$ 轴也处于对称平面内垂直于 $O_t x_t$ 轴,指向上方;$O_t z_t$ 轴垂直于 $x_t O_t y_t$ 平面,指向右方. $O_t x_t$ 轴延长线与其在 $x_d O_d z_d$ 的投影 Ax_{td} 交于 A 点.

控制飞行器平稳的飞行,必须知道它的飞行姿态. 俯仰角 ϑ 与横滚角 γ 对于飞行器非常重要. 通常,将 ϑ 定义为 $O_t x_t$ 与地平面的夹角,飞行器抬头时为正;γ 定义为 $O_t y_t$ 与 $O_d y_d$ 的夹角,飞行器右倾时为正[1].

当飞机紧贴地面水平飞行时,地平面可以看作是一个在飞机所在位置与地球相切的平面. 如果沿飞机 $O_t x_t$ 轴向安装一前视成像系统,则像平面垂直于透镜成像轴,同时垂直于地平面. 这时,地平面在像平面中的投影为一条直线,并把像平面分为天、地两部分,且直线在相平面的中间位置. 当飞机仍保持平飞,但并不贴地飞行,而是有一个较小的高度 h 时,地平面与地球的切点不在飞机的位置,而是在 s 处,s 为经过飞机成像系统的地球的切线,见图 2b 所示. 此时地平面已经不再垂直于像平面,而是存在一个夹角 ε_0,如图 2a 所示.

地平面与像平面夹角的大小可以按照下式计算

$$\varepsilon_o = \arccos\left(\frac{R}{R+h}\right) \tag{1}$$



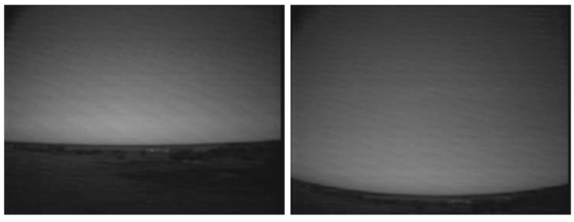
a 地平面不再垂直像平面 b 地平面与飞行高度的关系

图 2 地平面与飞机的位置关系

式中,R 为地球半径;h 为飞机的飞行高度;ε_o 为地平面与像平面之间的夹角.

由式(1)可见,当飞机在中低空水平飞行时,$h \ll R$,ε_o 可忽略不计,仍然可以认为,地平面在飞行器所处的位置与地球相切,并且与像平面垂直. 地平面在像平面上的投影为一条直线,位于像平面的中间位置,把像平面均分为天、地两部分.

视觉系统是一个非线性系统,地平面经透镜成像时,会发生变形,不再是一条直线,如图 3 所示.



a 地平线在图像中心位置 b 地平线在图像边缘位置

图 3 透镜成像与图像变形

可见,景物越偏离透镜主轴,变形越大. 如果透镜视野比较小,或者保证地平线在像平面的中间位置,其变形可以忽略不记.

沿 $O_t x_t$ 安装一前视成像系统,地平线在像平面中的倾角就是飞机的横滚角 γ. 飞机平飞时,地平线在像平面的中间位置. 飞机抬头,地平线的像下移;飞机低头,地平线的像上移. 定义[2]

$$\sigma = \frac{\text{地平线上方图像面积}}{\text{整幅图像面积}} \tag{2}$$

根据 σ 可以判断飞机是否有俯仰角.

2 地平线算法的实现

假设:①图像中的地平线为一条直线;②地平线把图像分为 2 个特性截然不同的区域[2]. 基于这 2 个假设,地平线算法必须解决 2 个问题:①如何在图像中找到一条直线;②找到的直线是不是真的地平线.

基于假设①,只要能在一幅图像的边缘确定 2 个点,就可以在图像中画出一条直线. 假设图像

面积为 256 像素×256 像素,在图像的边缘,每 4 个像素取一个点,在图像四周,共可以取 256 点,能确定不包括图像边缘在内的直线的数目为

$$n = \binom{4 \times V}{2} - 4 \times \binom{V}{2} - 4 \times V \Big|_{V=64} = 24\,320 \tag{3}$$

V 为图像每条边的点数,在 256×256 的图像中,V=64.

这些直线的角度分辨率为

$$\delta = \arctan\left(\frac{4}{256}\right) = 0.895\,2^\circ \tag{4}$$

在找到一条需要的直线之后,在直线附近进一步进行搜索,每个像素作为一个点寻找直线,可将角度分辨率提高到

$$\delta = \arctan\left(\frac{1}{256}\right) = 0.003\,9^\circ \tag{5}$$

在 n 条直线中找到真正的地平线,必须考虑假设②:图像中有天有地,天空区域的像素是相似的,大地区域的像素是相似的,2 个区域之间的像素是不相似的. 绘出图 4a 的像素的 RGB 空间分布,如图 4b 所示.

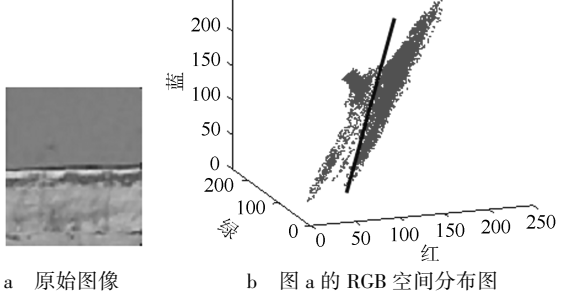
可以看到,图 4a 明显分为天、地 2 个区域,图 4b 中点的分布也分为 2 个区域,已经用一条直线(其实是一个平面)分开. 要实现地平线算法,必须找到一种能将图 4a 中 2 个区域分割开的方法,即在图 4b 中找到一个平面,将像素分为 U,V 两部分.

在飞行器得到的图像中,绘出一条直线,直线任一侧定义为 U,另一侧定义为 V,两侧的像素分别为

$$\left.\begin{aligned} \boldsymbol{x}_i^u &= \begin{bmatrix} r_i^u & g_i^u & b_i^u \end{bmatrix}^{\mathrm{T}} \qquad i \in \{1,2,\cdots,n_u\} \\ \boldsymbol{x}_j^v &= \begin{bmatrix} r_j^v & g_j^v & b_j^v \end{bmatrix}^{\mathrm{T}} \qquad j \in \{1,2,\cdots,m_v\} \end{aligned}\right\} \tag{6}$$

式中,r,g,b 分别为像素的红、绿、蓝通道;$\boldsymbol{x}_i^u$,$\boldsymbol{x}_i^v$ 分别为图像 U,V 中的像素.

定义规则[2]



a　原始图像　　b　图 a 的 RGB 空间分布图

图 4　像素在 RGB 空间中的分布

$$J = (\boldsymbol{\mu}_u - \boldsymbol{\mu}_v)^{\mathrm{T}} \boldsymbol{A}^{-1} (\boldsymbol{\mu}_u - \boldsymbol{\mu}_v) \tag{7}$$

式中

$$\boldsymbol{A} = \boldsymbol{\Sigma}_u + \boldsymbol{\Sigma}_v + \begin{bmatrix} \lambda_1^u & 0 & 0 \\ 0 & \lambda_2^u & 0 \\ 0 & 0 & \lambda_3^u \end{bmatrix}^2 + \begin{bmatrix} \lambda_1^v & 0 & 0 \\ 0 & \lambda_2^v & 0 \\ 0 & 0 & \lambda_3^v \end{bmatrix}^2 \tag{8}$$

$$\boldsymbol{\Sigma}_u = \frac{1}{n_u - 1} \sum_{i=1}^{n_u} \left(\boldsymbol{x}_i^u - \boldsymbol{\mu}_u\right)\left(\boldsymbol{x}_i^u - \boldsymbol{\mu}_u\right)^{\mathrm{T}}$$

$$\boldsymbol{\Sigma}_v = \frac{1}{m_v - 1} \sum_{i=1}^{m_v} \left(\boldsymbol{x}_i^v - \boldsymbol{\mu}_v\right)\left(\boldsymbol{x}_i^v - \boldsymbol{\mu}_v\right)^{\mathrm{T}} \tag{9}$$

$$\left.\begin{aligned} \boldsymbol{\mu}_u &= \frac{1}{n_u} \sum_{i=1}^{n_u} \boldsymbol{x}_i^u \\ \boldsymbol{\mu}_v &= \frac{1}{m_v} \sum_{i=1}^{m_v} \boldsymbol{x}_i^v \end{aligned}\right\} \tag{10}$$

$\boldsymbol{\Sigma}_u$,$\boldsymbol{\Sigma}_v$ 分别为区域 U,V 中像素的方差;λ_i^u,λ_i^v 分别为 $\boldsymbol{\Sigma}_u$,$\boldsymbol{\Sigma}_v$ 的特征值.

$\boldsymbol{\Sigma}_u$,$\boldsymbol{\Sigma}_v$ 值越大,表明相应区域中像素分布偏离均值越远或者偏离均值的点较多. λ_i^u,λ_i^v 表明了 $\boldsymbol{\Sigma}_u$,$\boldsymbol{\Sigma}_v$ 中数据的弥散程度,值越大,表明 $\boldsymbol{\Sigma}_u$,$\boldsymbol{\Sigma}_v$ 中数据分布得越不集中[3]. 规则 J 保证了:当直线两侧像素的均值差别越大、同侧像素差别越小、像素分布越集中,J 值越大. 根据假设①寻找地平线时,满足 $J' = \max J$ 的直线即为真正的地平线.

3　图像的预处理

飞行器的成像系统在获取图像、保存图像、传输图像的过程中有可能引进噪声,对图像造成污染. 对受到噪声污染的退化图像的复原可以采用线性滤波方法来处理. 但是多数线性滤波器具有低通特性,在去除噪声的同时也使图像的边缘变得模糊了[4],这对地平线算法的精度是有影响的. 地平线算法是基于统计的算法,如果能把脉冲性噪声滤掉,又能对图像的边缘有较好的保护,将提高算法的精度.

中值滤波是一种去除噪声的非线性处理方法,对所谓的椒盐噪声(pepper noise)的滤除非常有效[4]. 本文的算法将中值滤波的概念应用到 RGB 图像,设

$$\left\{\boldsymbol{x}_{i,j}[r \quad g \quad b], (i,j) \in \boldsymbol{I}^2\right\} \tag{11}$$

表示数字图像各像素的 RGB 通道值,滤波窗口为 A 的二维中值滤波可定义为

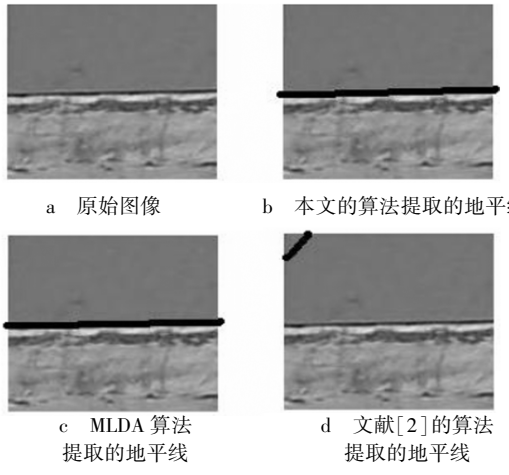
$$\boldsymbol{y}_{i,j} = \underset{A}{\mathrm{Med}}\{\boldsymbol{x}_{i,j}\} =$$

$$\mathrm{Med}\{\boldsymbol{x}_{(i+r),(j+s)}, (r,s) \in \boldsymbol{A}, (i,j) \in \boldsymbol{I}^2\} \tag{12}$$

本文的算法中,滤波窗口选为 $\boldsymbol{A}$=3×3.

4 算法仿真

图 5 为本文提出的地平线算法与参考文献中提到的算法的比较.



a 原始图像 b 本文的算法提取的地平线

c MLDA 算法提取的地平线 d 文献[2]的算法提取的地平线

图 5 地平线算法比较图

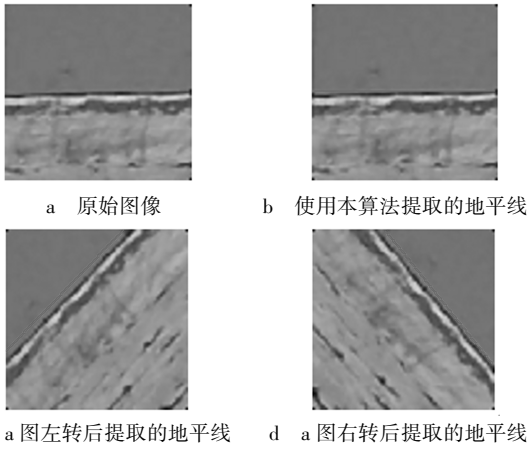
图 5a 为原始图像,在图中能明显看到地平线;图 5b 为使用本文的算法得到的"地平线",可以看出搜索出的"地平线"与真实的地平线很好地重合;图 5c 是 MLDA(Multiscale Linear-Discriminant Analysis)算法[5]得到的"地平线",其得到的直线与真实地平线重合,但 MLDA 算法需要多级别的迭代,运算量远远大于本文的算法;在图 5d 中,参考文献[2]的算法根本无法正确检测到地平线.

图 6 为本文提出的算法的应用仿真,这里只以横滚角为例. 图 a,b 为原始图像,c 将 a 图左转 46°,d 将 a 图右转 51°,分别对 b,c,d 3 图应用本文提出的算法. 在图 6 中可以看到,应用本算法,在 3 幅图像都搜索到了真正的"地平线". 将本算法应用于 b 图得到的"地平线"倾角为 0°,应用于 c 图得到的"地平线"倾角为 46.332 2°, 应用于 d 图得到的"地平线"倾角为-51.044 1°.

仿真结果表明,本文的算法能估计出飞行器的姿态角,精度满足飞行器的姿态稳定要求.

当飞行器俯仰角和横滚角都比较小时,例如,地平线的位置可以用图 7 中的位置 1 表示时,可以使用 σ 表示俯仰角 ϑ,且俯仰角与横滚角无耦合,见图 7.

在地平线的位置由位置 1 运动到位置 2 的过程中,σ 出现了跳变,而且此时俯仰角与横滚角出现耦合. 即,假设飞行器俯仰角不改变,当地平线在图像中的位置 2 附近抖动时,σ 仍会改变. 这种耦合会给姿态估计带来误差. 可以通过算法解耦,



a 原始图像 b 使用本算法提取的地平线

c a 图左转后提取的地平线 d a 图右转后提取的地平线

图 6 地平线算法

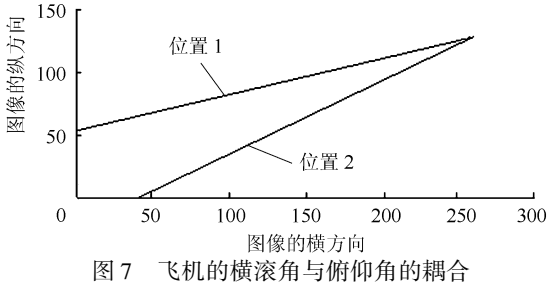


图 7 飞机的横滚角与俯仰角的耦合

但是这会增加算法的复杂度. 为了解决上述耦合问题,可以在飞行器上安装 2 套视觉系统,其中系统 S_1 的光轴与 O_1x_1 重合,系统 S_2 的光轴与 O_1z_1 重合. 对于前视系统 S_1,其图像中的地平线的倾角就是飞行器的横滚角 γ;对于系统 S_2,地平线的倾角是俯仰角 ϑ,S_1 与 S_2 无耦合. 此方案能准确地估计出飞行器的飞行姿态,进而进行姿态控制.

5 结 束 语

本文提出的算法能迅速地估计出飞行器的姿态角,为飞行器的姿态控制提供必需的数据. 本方法只需飞行器安装 2 套成像系统即可工作,无需其他设备,特别适合于只能安装视觉系统的飞行器,尤其是 MAV.

本算法需要对图像进行处理,运算量大,对处理器的运算速度要求高. 为了满足姿态角运算的实时性,在飞行器飞行过程中,只需要在导航系统启动时,进行一次全图像范围的扫描,以寻找地平线. 一旦找到地平线,在以后的运算中,只需要在地平线附近小范围搜索即可,大大减少了运算量,增加了实时性,可以满足飞行器的姿态稳定系统的需要.

(下转第 898 页)