

地面目标特征识别与无人飞行器位姿估计*

张 梁¹, 徐锦法¹, 夏青元², 于永军¹

(1. 南京航空航天大学 直升机旋翼动力学国家级重点实验室, 江苏 南京 210016;
2. 南京理工大学 高维信息智能感知与系统教育部重点实验室, 江苏 南京 210094)

摘要:针对小型无人飞行器位置姿态估计问题,提出了一种基于视觉图像目标特征的相对位姿估计算法。应用 Camshift 算法获取目标初始位置,利用非线性尺度空间下的 KAZE 特征进行跟踪区域特征点提取,与源目标特征点进行匹配,得到精确的目标位置信息,实现了在图像平面内的目标快速跟踪,并得到机体轴系下无人飞行器与目标间相对位置和姿态角的估计值。对算法进行了实验验证,具有优良的跟踪性和实时性。

关键词:无人飞行器;目标识别;位姿估计;KAZE 特征;Camshift 算法

中图分类号:V249.31 文献标志码:A 文章编号:1001-2486(2015)01-159-06

Feature recognition of ground target and position and attitude estimation for unmanned aerial vehicle

ZHANG Liang¹, XU Jinfa¹, XIA Qingyuan², YU Yongjun¹

(1. National key Laboratory of Rotorcraft Aeromechanics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China;
2. Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: Aimed at the problem of position and attitude estimation for UAV, a relative position and attitude estimation algorithm based on target features in image was proposed. The initial location of the target was obtained with Camshift algorithm. The feature points in tracking area based on the nonlinear scale space were picked up with KAZE features, which were used to match with the feature points of the source target. The exact location of the target could be obtained and the target could be tracked quickly in the picture plane. The estimation of relative position and attitude between the unmanned aerial vehicle and target was conducted in the body frame of axes. Some experiments were fulfilled for the verification of the algorithm. Results show that the proposed algorithm has strong tracking and real-time performance.

Key words: unmanned aerial vehicle; target recognition; position/attitude estimation; KAZE features; camshift algorithm

目前利用视觉图像对无人飞行器(Unmanned Aerial Vehicle, UAV)目标进行识别与跟踪算法的研究集中在两大方向,即基于色彩的目标识别与跟踪^[1-2]和基于特征点的目标识别与跟踪^[3-5]。前者大多采用 Camshift 跟踪算法^[6],通过对特定颜色目标的色彩提取,实现快速跟踪以及目标定位。后者通常采用 SIFT(Scale Invariant Feature Transform)特征^[7]或者 SURF(Speeded Up Robust Features)特征^[8]检测算法,通过特征点匹配,计算目标位置与姿态。两大研究方法各有所长,亦各有缺点。基于颜色的 Camshift 算法运算相对简单,能满足实时性要求,但是易受环境干扰,且受

限于算法本身,无法计算完整的三轴姿态变化。基于特征点的目标识别算法可以较准确地计算得到目标与相机的相对位置与姿态,但是特征点检测算法耗时相对较长, SURF 算法相比 SIFT 算法在加快速度的同时也牺牲了精度。飞行器进行目标识别时因场景复杂,且当飞行器大机动飞行时容易丢失图像细节,对大尺度下图像特征点检测提出了较高要求,要确保特征点检测的准确性。

SIFT, SURF 都是对图像细节在基于高斯核的线性尺度空间大尺度下进行滤波处理,使得图像中的边缘特征变得模糊,降低了大尺度下特征点检测的准确度。KAZE 特征是对像素点梯度信

* 收稿日期:2014-05-28

基金项目:国家部委预研基金资助项目(9140C400504130C40003);教育部重点实验室基金资助项目(30920140122006);中国博士后科学基金资助项目(2013M541668)

作者简介:张梁(1984—),男,浙江湖州人,博士研究生, E-mail: zerolion@nuaa.edu.cn;
徐锦法(通信作者),男,教授,博士,博士生导师, E-mail: xjfae@nuaa.edu.cn

息进行基于非线性尺度空间下的非线性滤波,突出了图像的边缘细节,提高了特征点检测准确性,但全局图像 KAZE 特征点检测耗时太长。为解决上述问题,本文提出了基于 KAZE 特征^[9]和 Camshift 算法的 UAV 目标识别与跟踪方法,运用 Camshift 算法进行初始目标识别与跟踪,在跟踪框内利用 KAZE 特征进行目标精确识别和飞行器相对位置与姿态计算。

1 特征点检测与描述

对目标进行特征点检测与描述是进行特征点匹配以及目标定位的前提,目前通常采用基于高斯线性尺度空间的 SIFT、SURF 特征,包括线性多尺度空间构建、特征点检测、特征点描述等步骤。高斯线性尺度空间可在大尺度下对噪点进行平滑并能突出图像显著特征,但在所有尺度上对图像细节和噪声进行了相同的平滑,尺度越大模糊程度越大、细节信息丢失越严重、特征点定位误差越大。由此提出了用非线性尺度空间来改进高斯尺度空间的方法,可使图像在大尺度下保持边缘特征不变,KAZE 特征正是基于非线性尺度空间的特征点检测与描述算法,其表现证明可以提高非线性尺度空间中不同尺度下图像特征检测和描述的可重复性和准确性^[9]。

1.1 非线性扩散滤波

非线性扩散方法是构建非线性尺度空间的基础,它描述了图像亮度随尺度增加而发生的变化过程,此过程与热运动扩散方程相似。经典的非线性扩散公式如式(1)所示。

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, t) \cdot \nabla L) \quad (1)$$

式(1)中 div 和 ∇ 分别表示散度和梯度算子。 c 为传导系数,依赖于图像的局部微分形式,可以是标量或者张量。时间 t 是尺度参数,取值越大滤波效果越明显、图像越简单。

由式(1)可知通过改变传导系数 c 可实现图像非线性滤波,达到大尺度下保留图像边缘特征细节的目的。Perona 与 Malik^[10]通过使传导系数 c 仅依赖于梯度幅值的方法来减少特征边缘处的扩散,从而使图像在局部区域内实现平滑,突出了边缘特征。传导系数形式如式(2)所示。

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|) \quad (2)$$

其中, ∇L_σ 是原始图像 L 经过高斯平滑后 L_σ 的梯度。Perona 与 Malik^[10]和 Weickert^[11]分别给出了关于传导系数 g 的计算公式。

1.2 KAZE 特征

KAZE 特征采取与 SIFT 相似的方法,通过递增的对数序列构建尺度空间,但在构建时保持图像分辨率不变。

$$\sigma_n(o, s) = \sigma_0 \times 2^{o+s/S} \quad (3)$$

$$o \in [0 \cdots O-1], s \in [0 \cdots S-1], n \in [0 \cdots N]$$

其中, o 表示组, s 表示层, σ_0 是初始尺度值,一般取 1.6, N 是尺度空间中图像的总数。由于非线性扩散滤波模型以时间为单位,需要把尺度 σ 转换为时间 t , KAZE 特征采用如下映射关系由尺度参数转换为时间参数 $\sigma_n \rightarrow t_n$, 并得到一组进化时间。

$$t_n = \sigma_n^2/2, n \in [0 \cdots N] \quad (4)$$

KAZE 特征首先对图像进行标准偏差为 σ_0 的高斯核卷积运算,减少图像噪声,之后计算得到传导系数 g , 运用组与层的关系得到 σ_n 递推公式以及时间参数 t_n , 运用加性算子分裂 (Additive Operator Splitting, AOS) 算法^[11]对式(1)进行半隐式差分,得到非线性尺度空间下各层迭代表达式。

$$L^{n+1} = \frac{1}{2} \sum_{l=1}^2 [I - 2(t_{n+1} - t_n)A_l(L^n)]^{-1} L^n \quad (5)$$

KAZE 特征采用类似 SURF 的方法,通过计算多尺度下的尺度规范 Hessian 矩阵行列式响应值来检测特征点, KAZE 特征对每一点都通过其所在的当前层、上一层以及下一层滤波图像上的 $\sigma_n \times \sigma_n$ 矩形窗口进行比较,从尺度空间与图像空间两方面寻找局部极大值。最后 KAZE 特征采用与 SIFT 类似的方法,通过子像素插值法^[7]将特征点位置精度提高到亚像素级别。

KAZE 特征在非线性尺度空间下使用 M-SURF 描述符^[12]对特征点进行描述。为使特征点描述符具有旋转不变性, KAZE 特征在尺度参数为 σ_n 的图像上以特征点为中心,使用搜索半径为 $6\sigma_n$ 的圆域寻找主方向。在特征点主方向运用 M-SURF 描述符得到 64 维描述符向量。最后将长度为 64 的描述符向量规范化为单位向量,用以去除光照影响,保证不同对比度下描述符的不变性。

2 特征点匹配

图像特征点匹配是通过两幅图像中的特征点描述符向量比较来得到匹配特征点集的过程,如穷尽搜索、基于 Kd 树的 Knn、BBF 算法等。穷尽搜索即将所有待匹配特征点与源特征点进行比较,具有最短欧氏距离的特征点作为匹配特征点,

算法较简单但耗时较长。Knn 算法基于 Kd 树,将特征点集用二叉树的方式建立索引,在此二叉树中寻找最近邻特征点。目标源图像事先给定,Kd 树可以提前建立而无需循环计算,可以节省匹配时间,提高效率。

尽管如此,用 Knn 算法搜索得到的匹配特征点集必定还存在一定数量的误匹配,需要进行提纯操作。用随机抽样一致性算法 (RANDOM SAmple Consensus, RANSAC)^[13] 能剔除误配点,完成匹配特征点集的提纯操作,算法每次随机提取 4 组匹配特征点,通过最小二乘法求取单应性矩阵 H ,计算其余匹配特征点集满足此单应性矩阵的数量,重复多次运算得到具有最大匹配数量的内点集作为提纯后的匹配特征点集。特征点匹配效果如图 1 所示,目标源图像与场景图像目标具有 25 对匹配特征点。

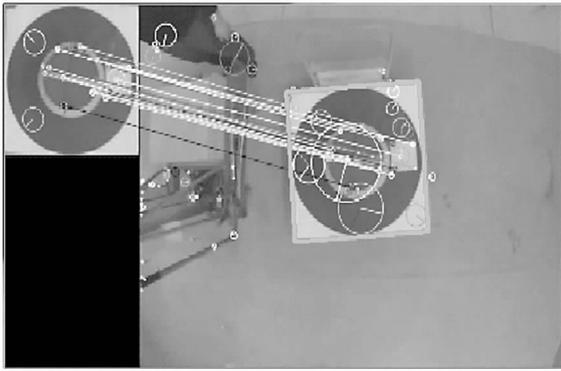


图 1 KAZE 特征匹配图

Fig. 1 Matches of KAZE features

3 目标识别算法优化

KAZE 特征比 SIFT、SURF 具有更好的准确性与可重复性,但是由于采用了非线性尺度空间,增加了计算复杂度,使得计算时间有所增长。为了满足实时性要求,引入 Camshift 算法对目标识别与跟踪算法进行优化。被跟踪目标通常在图像中尺寸较小,用 Camshift 算法将特征点检测范围缩小至目标附近,提高运算速度和准确度。

Camshift (Continuously Adaptive Mean-Shift) 算法是指连续自适应 Meanshift 算法,Meanshift 算法为均值偏移算法,经过不断改进,已在图像跟踪应用领域中演变为在数据密度分布中寻找局部极值的迭代方法。Camshift 算法以 Meanshift 算法为核心,每一时刻均对图像进行 Meanshift 迭代计算,并以此刻得到目标窗口,作为下一时刻 Meanshift 算法搜索窗口初始值,实现自适应目标跟踪。

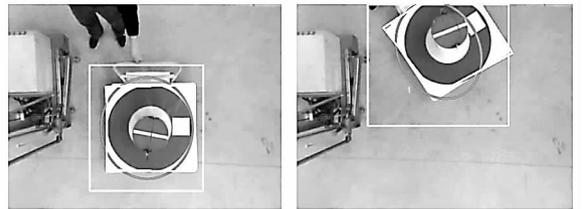
Camshift 算法得到目标搜索窗,作为初始目标识别与跟踪窗口,在窗口内进行 KAZE 特征点匹配,能减少特征点检测与匹配时间,且 Camshift 算法基于迭代运算,适合于计算机数值计算,运算时间相对较短。但是 Camshift 算法严重依赖目标颜色,且易受光照影响,识别与定位精度较低,因此通过如下方法减小 Camshift 跟踪误差影响,提高目标特征点匹配成功率:

1) 场景图像经过 Camshift 检测后得到初始跟踪窗口,以此窗口中心为中心,纵、横向分别扩大至原窗口 1.2 倍,并以此新窗口作为 KAZE 特征搜索窗口,可以减少 Camshift 跟踪窗口偏离目标所引起的匹配误差。若 Camshift 检测成功则执行步骤 3,否则执行步骤 2。

2) 当本次 Camshift 检测失败,则选用最近一次目标特征点匹配成功时的搜索窗口作为本次搜索窗口并执行步骤 3。

3) 在搜索窗口内进行 KAZE 特征点检测与匹配,若匹配成功(匹配特征点大于 12)则返回结果并将匹配窗口作为下一帧 Camshift 跟踪起始窗口,否则执行步骤 4。

4) 将整幅图像作为本次搜索窗口并进行 KAZE 特征匹配,若匹配成功则返回结果并将匹配窗口作为下一帧 Camshift 跟踪起始窗口,否则认为匹配失败。



(a)

(b)

图 2 结合 Camshift 与 KAZE 的目标识别

Fig. 2 Target recognition of KAZE and Camshift

如图 2 所示为基于 Camshift 与 KAZE 特征的目标识别结果,图 2(a)中椭圆区域为 Camshift 跟踪结果,可见正常跟踪情况下目标原点明显偏移,而 KAZE 特征点匹配(方形区域)则精确显示目标方位。由于目标图像颜色易于识别,因此 Camshift 跟踪失败情况极少发生,而如图 2(b)所示由于目标偏出画面等原因导致 Camshift 跟踪窗口明显偏移情况时有发生,但经过放大搜索窗口(方形区域)处理后,目标仍然可以实现匹配与精确定位。

4 无人飞行器相对位姿估计

利用 KAZE 特征点匹配算法得到与目标源图像特征点相匹配的场景图像目标特征点,能计算得到飞行器与目标的相对位置和姿态,实现目标识别与跟踪。为便于理解相对位置与姿态,参考坐标系选用如图 3 所示,相机与机身固连,镜头向下,忽略相机安装,认为相机坐标系与机体坐标系二原点重合。

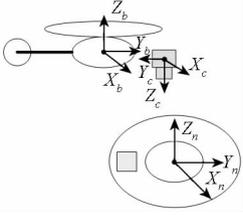


图 3 坐标系对应关系

Fig. 3 Relationship of coordinates

特征点匹配提纯操作时已得到目标源图像与场景目标图像间的单应性矩阵 \mathbf{H} , 目标相对姿态估计值从矩阵 \mathbf{H} 计算中得到。图像间像素点单应性投影关系如式(6)所示。

$$p' = \mathbf{H}p \quad (6)$$

式(6)中, p 为源图像中像素点, p' 为场景图像中像素点。单应性矩阵 $\mathbf{H}[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$ 中包含有旋转矩阵 $\mathbf{R}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3]$, 对应相机坐标系下目标源图像与场景图像中目标的旋转关系, 旋转矩阵求解如式

$$\mathbf{C}_n^b = \begin{bmatrix} \cos\varphi\cos\psi + \sin\varphi\sin\theta\sin\psi & -\cos\varphi\sin\psi + \sin\varphi\sin\theta\cos\psi & -\sin\varphi\cos\theta \\ \sin\theta\sin\psi & \cos\theta\cos\psi & \sin\theta \\ \sin\varphi\cos\psi - \cos\varphi\sin\theta\sin\psi & -\sin\varphi\cos\psi - \cos\varphi\sin\theta\cos\psi & \cos\varphi\cos\theta \end{bmatrix} \quad (9)$$

式(9)中, φ 表示相对滚转角, θ 表示相对俯仰角, ψ 表示相对航向角。

$$\begin{cases} \theta = \arcsin(c_{23}) \\ \varphi = \arctan(-c_{13}/c_{33}) \\ \psi = \arctan(c_{21}/c_{22}) \end{cases} \quad (10)$$

三轴相对姿态角由式(10)得到, 再通过象限判断进行确认。目标相对位置通过相机原点在场景图像中的坐标得到, 取目标图像中心作为目标原点, 目标源图像四个顶点坐标为 $p_i(u_i, v_i)$, $i \in [1, \dots, 4]$, 则经过矩阵 \mathbf{H} 变换后场景图像中目标图像顶点坐标 p'_i 为:

$$p'_i = \mathbf{H}p_i, i \in [1, \dots, 4] \quad (11)$$

场景图像中目标原点 $p'_i(u'_i, v'_i)$ 通过计算 4 个顶点型心得到:

$$\begin{cases} u'_i = 1/4 \sum u'_i \\ v'_i = 1/4 \sum v'_i \end{cases} \quad (12)$$

(7)所示^[14]。

$$\begin{cases} \mathbf{r}_1 = \lambda \mathbf{M}^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 = \lambda \mathbf{M}^{-1} \mathbf{h}_2 \\ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \end{cases} \quad (7)$$

式(7)中, \mathbf{M} 为相机内参数矩阵, 可通过标定得到, $\lambda = 1 / \|\mathbf{M}^{-1} \mathbf{h}_1\|$ 为比例因子。

使用实验数据解算旋转矩阵 \mathbf{R} 时, 由于数据存在误差, 无法得到一个精确的旋转矩阵 \mathbf{R} 满足 $\mathbf{R}^T \mathbf{R} = \mathbf{R} \mathbf{R}^T = \mathbf{I}$ 。将旋转矩阵 \mathbf{R} 进行 SVD 分解^[14], 令 $\mathbf{R} = \mathbf{U} \mathbf{D} \mathbf{V}^T$, $\mathbf{D} = \mathbf{I}$, 使旋转矩阵 \mathbf{R} 强制转化为满足条件的旋转矩阵 $\mathbf{R}_1 = \mathbf{U} \mathbf{I} \mathbf{V}^T$ 。

选取世界坐标系为目标原点, 方向为东、北、天, 机体坐标系下特征点与世界坐标系下特征点对应关系如式(8)所示。

$$\begin{cases} P'_b = C_n^b(P_n + t_n) \\ C_n^b = C_c^b R_1 C_{n0}^c \end{cases} \quad (8)$$

式(8)中, P'_b 为特征点在机体坐标系下的三维坐标, P_n 和 t_n 为特征点在世界坐标系下的三维坐标与平移量, C_n^b 是世界坐标系到体轴系的转换矩阵, C_c^b 是相机坐标系到体轴系的转换矩阵, C_{n0}^c 是源图像中相机坐标系到体轴系的转换矩阵, 是一个常值矩阵。

拍摄目标源图像时, 图像平面与目标平面平行, 通过分解 C_n^b 得到三轴相对姿态角如式(9)所示。

相机经标定后焦距为 f , 光心坐标为 (u_0, v_0) , 相机坐标系取 X 轴向右, Y 轴向后, Z 轴向下, 相机固定安装于机身下方, 镜头垂直向下。相机相对目标距离 z 从激光高度传感器得到, 目标相对相机坐标系位置估计如式(13)所示。

$$\begin{cases} x'_{i,c} = (u'_i - u_0)f/z'_{i,c} \\ y'_{i,c} = (v'_i - v_0)f/z'_{i,c} \end{cases} \quad (13)$$

式(13)中, $P'_{i,c}(x'_{i,c}, y'_{i,c}, z'_{i,c})$ 为相机坐标系下目标原点的坐标, 下标 t 表示目标, 下标 c 表示相机坐标系。根据式(8)可知:

$$P'_{i,c} = C_c^c C_n^b(P_{i,n} + t_{i,n}) \quad (14)$$

$$t_{i,n} = C_b^n C_c^c P'_{i,c} \quad (15)$$

式(14)中, $P_{i,n}$ 为目标原点在世界坐标系的坐标, 由于目标原点与世界坐标系原点重合, $P_{i,n} = 0$ 。 $t_{i,n}$ 是相机坐标系原点(机体坐标系原点)在世界坐标系的位移。

至此,飞行器与目标的相对姿态与位置根据式(10)与式(15)分别求得。计算流程如图4所示。

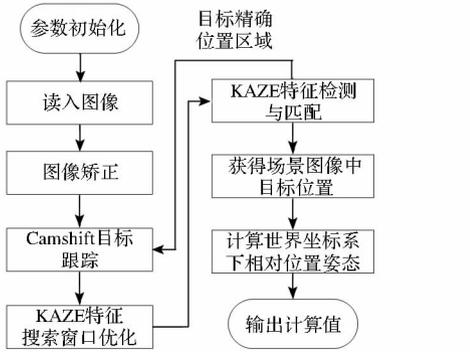


图4 目标识别与位姿估计流程图

Fig.4 Flow chart of target recognition and position and attitude estimation

5 实验及结果

为验证所提出的算法,用高性能 Gopro Hero3 运动相机(图像分辨率 640×480 像素)作为图像采集设备,用笔记本电脑(CPU 为 Intel I7 - 3520M,主频达到 2.9GHz,内存为 4Gb)作为图像处理计算机,构建了一套实验验证系统并进行了实验验证。应用程序用 OpenCV_2.4.7 和 VS2010 C++ 设计开发完成。

表1所示为利用所构建实验验证系统得到的目标图像分别利用 KAZE 特征、Camshift 算法和本文提出的算法完成的目标识别估计所需要的计算时间对比。本文算法中的非线性尺度空间选取 3 组,每组 4 层,AOS 运算使用 Boost 多线程技术,提高了运算速度。KAZE 特征尽管具有良好的尺度与旋转不变性、稳定性与可重复性^[9],但运算时间偏长、难以满足实时运算要求,Camshift 算法运算耗时较短,与 KAZE 特征结合,达到了优化目的。表1中4项测试场景图像是不相同的,所需运算时间也不相同,KAZE 特征运算时间均已超过 450ms,利用 Camshift 算法进行优化,使目标识

表1 运算时间比较

Tab.1 Comparison of operation time

实验样本	KAZE(ms)	Camshift(ms)	本文算法(ms)
测试1	477.4	6.5	178.4
测试2	450.6	6.2	114.5
测试3	514.3	6.7	221.8
测试4	480.6	6.5	180.7

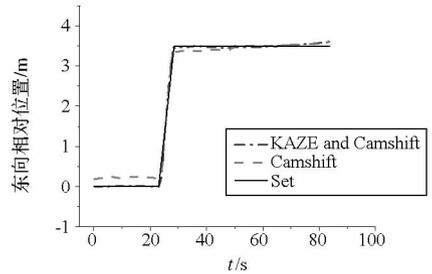
别估计时间提高了 2 倍以上,更新频率可升至 5Hz 左右,满足实时要求。

表2 SURF,SIFT 与本文算法比较

Tab.2 Comparison of SURF, SIFT, KAZE and Camshift

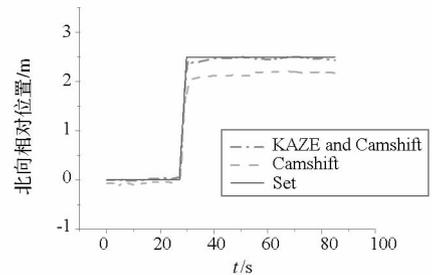
	SURF	SIFT	本文算法
匹配特征点数	17.5	21.3	25.6
内点	87.6%	85.7%	94.2%
检测与匹配时间(ms)	50.6	179.5	181.4

表2所示为 SURF、SIFT 与本文算法特征匹配以及运算时间比较情况,测试场景如图1所示。针对相同场景与相同目标检测 5 次后计算平均结果,SURF、SIFT 特征匹配采用整帧场景图像进行匹配运算。可见本文算法拥有最多匹配特征点数以及内点数,计算时间与 SIFT 相当。



(a) 东向相对位置

(a) Relative position of east



(b) 北向相对位置

(b) Relative position of north

图5 相对位置曲线

Fig.5 Curves of relative position

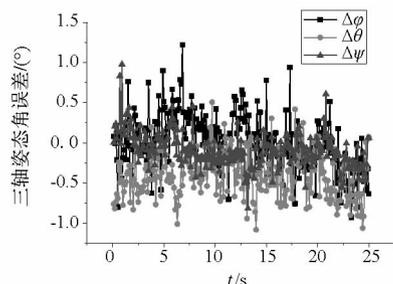
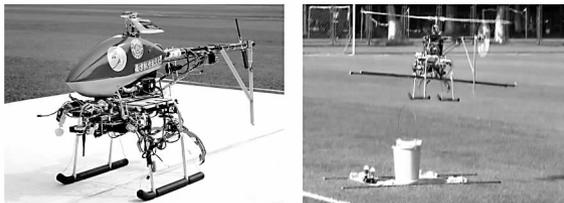


图6 三轴姿态角误差曲线

Fig.6 Curves of 3-axis attitude error

如图 5 所示为东、北向相对位置曲线,测试中在向东位移起始位置与北向位移终点均设置了干扰色,图中单纯用 Camshift 算法得到的相对位置估计值明显有偏移,误差近 20%,用本文所提算法得到的位置估计值,相对位置误差小于 4%,不但位置估计值误差小,抗干扰能力也得到增强,且跟踪迅速,满足快速响应要求。图 6 所示为地面测试三轴姿态角误差,测试过程中保持相机水平运动且无人姿态角改变。由图 6 可见三轴姿态角偏差较小,姿态角误差精度为 $\pm 1^\circ$,满足飞行器目标识别跟踪以及位姿估计要求。如图 7 所示为真实飞行环境下相机、图像处理计算机与“灵鸢”号无人旋翼飞行器,实验验证平台自主研发而成,利用该实验平台获取跟踪目标图像,用于验证本文所提出的飞行器目标识别与跟踪算法。图 8 所示为算法所得到的飞行器在目标上空自主悬停时的轨迹估计值。轨迹估计参考值选用差分 GPS 获得的估计值,因机身振动会给机载 GPS 天线带来振动干扰影响,造成差分 GPS 精度有所下降。由图 8 可见,“灵鸢”飞行器在自主飞行时定位精度约为 $\pm 1\text{m}$,飞行器定高为 2.5m(由激光高度仪测得高度进行定位),要求定位在目标正上方,用 GPS 定位估计值明显存有较大偏移,用图像特征进行位置估计,得到的估计值位于目标上方,范围圆域(1m 圆内)明显优于 GPS 定位估计,符合飞行器定位精度范围要求,估计精度高于差分 GPS。



(a) 飞行平台 (b) 飞行实验测试
(a) Flying platform (b) Test of flying experiment

图 7 无人旋翼飞行器算法验证平台

Fig. 7 UAV Platform for algorithm verification

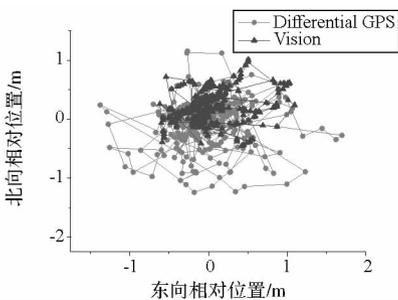


图 8 目标与飞行器相对位置曲线

Fig. 8 Curves of relative position between target and UAV

6 结论

通过引入基于非线性尺度空间的 KAZE 特征实现无人飞行器目标识别与位姿估计算法,提高了目标识别与位姿估计算法的稳定性,满足了无人飞行器飞行环境复杂多变、自身运动灵活等应用需求。

应用 Camshift 算法得到的跟踪区域作为目标识别与跟踪的初始值,有效缩小特征点检测区域,缩短了 KAZE 特征检测区域的搜寻过程,解决了 KAZE 特征检测与描述时间过长的问題,提高了在场景图像中目标区域的检测运算速度。

利用地面实验和无人旋翼飞行器飞行实验验证了本文所提出的目标识别与跟踪定位算法,结果表明无人飞行器目标识别与位姿估计算法是稳定可靠的,估计精度高于差分 GPS 估计精度,用于无人飞行器的目标识别与跟踪定位。

参考文献 (References)

- [1] Teuliere C, Eck L, Marchand E. Chasing a moving target from a flying UAV[C]. IEEE, 2011: 4929 - 4934.
- [2] Olivares Méndez M, Mondragon Bernal I F, Campoy Cervera P, et al. Aerial object following using visual fuzzy servoing[J]. 2011: 61 - 70.
- [3] Artieda J, Sebastian J, Campoy P, et al. Visual 3 - D SLAM from UAVs[J]. Journal of Intelligent & Robotic Systems. 2009, 55(4): 299 - 321.
- [4] Mondragon I F, Campoy P, Correa J F, et al. Visual model feature tracking for UAV control[C]. Alcalá de Henares: 2007: 1 - 6.
- [5] Jay H C, Won - Suk L, Hyochoong B. Helicopter guidance for vision-based tracking and landing on a moving ground target[C]. Gyeonggi - do: 2011: 867 - 872.
- [6] Bradski G R. Computer vision face tracking for use in a perceptual user interface[J]. 1998.
- [7] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision. 2004, 60(2): 91 - 110.
- [8] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF) [J]. Computer vision and image understanding. 2008, 110(3): 346 - 359.
- [9] Alcantarilla P F A N, Bartoli A, Davison A J. KAZE features[Z]. Springer, 2012214 - 227.
- [10] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion [J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1990, 12(7): 629 - 639.
- [11] Weickert J, Romeny B T H, Viergever M A. Efficient and reliable schemes for nonlinear diffusion filtering [J]. Image Processing, IEEE Transactions on. 1998, 7(3): 398 - 410.
- [12] Agrawal M, Konolige K, Blas M R. Censure: Center surround extremas for realtime feature detection and matching[Z]. Springer, 2008102 - 115.
- [13] Derpanis K G. Overview of the RANSAC Algorithm[Z]. Technical report, Computer Science, York University, May 2010. http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf, 2010.
- [14] Bradski G, Kaehler A. Learning OpenCV: Computer vision with the OpenCV library [M]. O' Reilly Media, Incorporated, 2008.