

文章编号 1004-924X(2014)08-2232-10

基于四轴飞行器的单目视觉避障算法

赵 海, 陈星池*, 王家亮, 曾若凡

(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

摘要:针对四轴飞行器平台上单目视觉避障算法实时性不够, 准确率不高的问题, 提出了一种新的基于金字塔 LK(Lucas-Kanade)光流与平移光流融合的单目视觉避障算法并给出了这一方法的数学推导过程。在配有 320 pixel×240 pixel 摄像头的 ArduCopter 四轴飞行器上开发和实现了该算法, 并进行了多次飞行测试验证了算法的可行性与有效性。在飞行器自主飞行过程中, 采用消失点进行导航, 并结合神经网络 PID 对其进行飞行自适应控制。与单目即时定位与地图构建(SLAM)避障算法、Horn-Schunck 光流避障算法以及图像分割避障算法比较了算法的实时性和准确性, 结果显示: 本算法具有实时性好和准确率高的优点, 可以满足实际工程中四轴飞行器的实时避障要求。

关键词:四轴飞行器; 单目视觉; 避障算法; 金字塔 LK 算法; 平移光流; 融合光流

中图分类号:TP242.62; V249 **文献标识码:**A **doi:**10.3788/OPE.20142208.2232

Obstacle avoidance algorithm based on monocular vision for quad-rotor helicopter

ZHAO Hai, CHEN Xing-chi*, WANG Jia-liang, ZENG Ruo-fan

(School of Information Science & Engineering, Northeastern University, Shenyang 110004, China)

* Corresponding author, E-mail: chenxingchi999@126.com

Abstract: For bad real-time ability and poor accuracy rate of the monocular obstacle avoidance algorithm for quad-rotor helicopters in practical engineering applications, a new monocular obstacle avoidance algorithm based on Pyramid Lucas-Kanade optical flow and moving optical flow was proposed and the mathematical derivation process of the algorithm was given. The algorithm was established and implemented on a ArduCopter quad-rotor helicopter equipped with a 320 pixel×240 pixel camera and many experiments was done to verify the feasibility and effectiveness of the algorithm. In the process of vehicle autonomous flight, the vanishing points were used for navigation and a neural net PID was used to control flighting adaptively. As compared with the monocular Synchronous Location Mapping (SLAM) obstacle avoidance algorithm, Horn-Schunck optical flow obstacle avoidance algorithm and the image segmentation obstacle avoidance algorithm, the proposed algorithm shows better real-time ability and accuracy rate, and can meet the requirements of obstacle avoidance in real time for quad-rotor helicopters.

Key words: quad-rotor helicopter; monocular vision; obstacle avoidance algorithm; pyramid LK algorithm; translation optical flow; fusion optical flow

收稿日期: 2014-01-07; 修订日期: 2014-01-30.

基金项目: 国家科技支撑计划资助项目(No. 2012BAH82F04)

1 引言

随着信息技术和电子技术的不断进步,自主导航四轴飞行器得到了迅速发展。其中飞行器自主导航过程中对障碍物的判断和避障是自主飞行中的一个重要问题。目前国内外有许多研究机构在四轴飞行器或者无人机避障领域展开研究。德国 Albert Ludwigs University 的 Slawomir Grzonka 等人使用激光测距仪对障碍物进行判断,但他们只是对走廊环境中能否判断并通过障碍物进行了一组实验测试,并没有对算法的实时性和准确性进行研究^[1]。土耳其 Middle East Technical Universtiy 的 Aydin Eresen 等人提出了一种基于虚拟环境的视觉避障算法,但是仅在 google earth 软件中对纽约街头的背景障碍进行判断与导航,并没有对实际飞行路径中可能出现障碍的情况进行考虑分析^[2]。还有一些学者通过单摄像头采集图像并提取图像中的有关信息,从而进行避障研究。这些研究大部分都基于图像分割,图像深度提取,以及众多的光流算法^[3]。图像分割方法使用颜色标记图像中无障碍物的部分(如空旷的道路或者天空),其优点在于计算效率高,缺点是精度比较低^[4]。基于深度提取的方法使用的更为普遍。由于区域内可见障碍物信息都是可提取的,所以这种方法需要创建相关区域内的 3D 地图^[5-6]。但是这样一来,如果环境信息量大,算法计算量就非常大,实时性就很差,很难满足飞行器平台对实时性的要求,而且,这些方法都需要在均匀照明环境下拍摄出高分辨率的图片,并且需要图片纹理清晰,才能进行有效处理。因此,使用已有算法控制移动机器人或飞行器会产生实时性差和准确率不高等问题。

本文提出了一种新的基于金字塔 LK(Lucas-Kanade)与平移光流融合的单目视觉四轴飞行器避障方法,并给出了其数学推导过程。提出的避障算法在配有 320 pixel×240 pixel 摄像头的 ArduCopter 四轴飞行器上进行开发和实现。四轴飞行器避障过程采用消失点进行导航,采用神经网络比例积分(Proption Integration Differentiation, PID)自适应控制进行飞行控制^[7-8]。本算法的优点是实时性好、准确率高、完全满足实际工程中四轴飞行器的实时避障要求。

2 金字塔 LK 光流算法

光流法是一种能根据相对较小变化产生流动向量的图像运动表达方式。光流拥有丰富的运动信息,表达了一种运动图像的变化,因此可以用来确定目标的运动。LK 算法的假设前提是亮度恒定,时间连续或者运动是“小运动”,空间一致^[9]。

设 $I(x, y, t)$ 是 t 时刻图像上点 (x, y) 的灰度, Δx 和 Δy 分别是 x 和 y 在 Δt 时间内的增量,则有:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \quad (1)$$

根据亮度恒前提假设,则有:

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t) \quad (2)$$

因此,将式(2)代入式(1),得

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0, \quad (3)$$

将公式(3)两边同时除以 Δt ,得到光流约束方程:

$$\frac{\partial I \Delta x}{\partial x \Delta t} + \frac{\partial I \Delta y}{\partial y \Delta t} + \frac{\partial I}{\partial t} = 0, \quad (4)$$

其中: $V_x = \frac{\Delta x}{\Delta t}$, $V_y = \frac{\Delta y}{\Delta t}$ 就是 $I(x, y, t)$ 在 x 和 y 方向的光流。公式(4)就是光流约束方程。

根据 LK 算法的空间一致性,可以将图像分割成若干小窗口,窗口的选择不能太大,也不能太小,否则会产生运动跟踪不一致和孔径问题。

设在窗口 N 中的光流估计误差定义为:

$$E(p, q) = \sum_{(x, y) \in N} W(x, y)^2 \left(\frac{\partial I}{\partial x}p + \frac{\partial I}{\partial y}q + \frac{\partial I}{\partial t} \right)^2, \quad (5)$$

其中: $I(x, y, t)$ 是图像上像素点在 t 时刻的灰度值; $W(x, y)$ 是窗口权重函数; $E(p, q)$ 表示光流误差; $(x, y) \in N$ 表示窗口 N 内任意一点 (x, y) 。

由于光流误差是基于窗口 N 内各点光流向量相同的假设,窗口内随着各点光流与中心点距离的增加,光流偏差也有所增大。所以设从中心向四周递减的 $W(x, y)$ 函数作为权值,以便更精准地控制邻域中心部分的光流偏差。比如当邻域大小为 3×3 时,定义一个窗口函数权值如下:

$$W = \begin{bmatrix} 0.4 & 0.8 & 0.4 \\ 0.8 & 1 & 0.8 \\ 0.4 & 0.8 & 0.4 \end{bmatrix}. \quad (6)$$

对估计误差 $E(p, q)$ 进行极小化,则有:

$$\frac{\partial E}{\partial p} = 2 \sum_{(x,y) \in N} W(x,y)^2 \left[\left(\frac{\partial I}{\partial x} \right)^2 p + \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} q + \frac{\partial I}{\partial x} \frac{\partial I}{\partial t} \right] = 0, \quad (7)$$

$$\frac{\partial E}{\partial q} = 2 \sum_{(x,y) \in N} W(x,y)^2 \left[\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} p + \left(\frac{\partial I}{\partial y} \right)^2 q + \frac{\partial I}{\partial y} \frac{\partial I}{\partial t} \right] = 0, \quad (8)$$

设 $\mathbf{V} = [p, q]^T$, 然后有:

$$\mathbf{G} = \begin{bmatrix} \sum_{(x,y) \in N} W(x,y)^2 \cdot \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{(x,y) \in N} W(x,y)^2 \cdot \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} \\ \sum_{(x,y) \in N} W(x,y)^2 \cdot \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} & \sum_{(x,y) \in N} W(x,y)^2 \cdot \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix}, \quad (9)$$

$$\mathbf{B} = \begin{bmatrix} \sum_{(x,y) \in N} W(x,y)^2 \cdot \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial t} \\ \sum_{(x,y) \in N} W(x,y)^2 \cdot \frac{\partial I}{\partial y} \cdot \frac{\partial I}{\partial t} \end{bmatrix}. \quad (10)$$

则上式可简写为 $\mathbf{G} \cdot \mathbf{V} + \mathbf{B} = 0$, 由此可知光流:

$$\mathbf{V} = -\mathbf{G}^{-1} \cdot \mathbf{B}, \quad (11)$$

当图像帧间的运动超过 1 pixel/frame 时, LK 算法计算误差大, 不适合区域搜索, 所以本文将采用金字塔 LK 算法来减少误差, 以提高光流计算的准确性和鲁棒性。金字塔 LK 算法涉及对不同分辨率图像的检测。从金字塔的最上层开始(像素最少的一层), 逐步向下直到金字塔的像素最多的一层(最高清晰度的图像), 如图 1 所示。

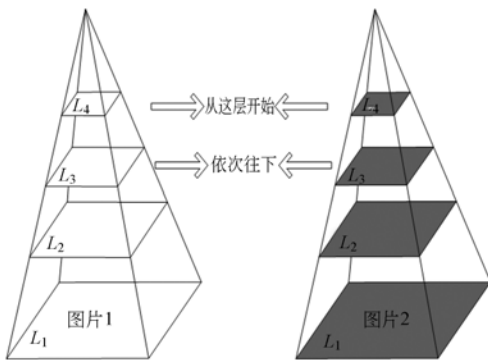


图 1 分层分解图像分析

Fig. 1 Image analysis of hierarchical decomposition

为了兼顾效果和实时性的问题, 采用 $m=4$ 的四层金字塔, $L_1=320 \times 240$, $L_2=160 \times 120$, $L_3=80 \times 60$, $L_4=40 \times 30$ 的像素。光流计算自顶层依次向下进行。在第 k 层内部由初始光流向量

\mathbf{V}_k 开始, 按照 L-K 算法进行搜索, 用计算出的光流残差 E_k 迭代更新光流向量。在第 $k(k=4, 3, 2, 1)$ 层内得到该层精度下的光流向量 \mathbf{V}_k 后, 经尺度变换作为下一层(第 $k-1$ 层)更高精度搜索的初始值, 这一过程不断进行, 直至估计出原始图像的光流。

3 单目 SLAM 避障算法

即时定位与地图构建(Simultaneous Localization and Mapping, SLAM)指机器人在完全未知环境中从未知位置开始运动, 运动过程中依靠自身携带的传感器(包括激光, 超声波和视觉等传感器)感知周围环境。根据控制信息和传感器观测数据同时进行位置估计和地图构建。在定位的基础上建立增量式地图, 从而实现机器人的即时定位以及环境地图的构建^[10]。

卡尔曼滤波器适合解决线性系统问题, 其基本原理是由前一时刻的状态估计值和当前状态的观测值推导出当前状态的估计值。而在 SLAM 过程中, 四轴飞行器的运动和环境特征之间的关系被描述成两个非线性系统, 从而需使用扩展卡尔曼滤波方法使非线性问题转化为线性问题。

对于基于扩展卡尔曼滤波的单目 SLAM 系统来说, 相机状态(四轴飞行器状态)、特征状态和扩展卡尔曼滤波系统总状态公式分别如下:

$$\mathbf{x}_c = (\mathbf{r}^T, \mathbf{q}^T, \mathbf{v}^T, \boldsymbol{\omega}^T)^T, \quad (12)$$

$$\mathbf{f}_i = (x_i, y_i, z_i)^T, \quad (13)$$

$$\mathbf{x} = (\mathbf{x}_c^T, \mathbf{f}_1^T, \mathbf{f}_2^T, \dots, \mathbf{f}_i^T, \dots, \mathbf{f}_n^T)^T, \quad (14)$$

其中: \mathbf{r}^T 为四轴飞行器上摄像头在世界坐标中的三维位置, 即位置状态; \mathbf{q}^T 为摄像头空间角度的四元数表示, 即姿态状态; \mathbf{v}^T 为摄像头线速度, $\boldsymbol{\omega}^T$ 为摄像头角速度; x_i, y_i, z_i 为第 i 个特征的三维空间坐标。

基于扩展卡尔曼滤波的单目 SLAM 避障算法流程如图 2 所示。在图 2 中, 四轴飞行器通过自带的 IMU(Inertial Measurement Unit)为运动模型预测提供线速度和加速度等参数, 再对运动模型预测与摄像头捕获的视频信号进行特征提取和匹配, 从而得出测量模型。然后, 对运动模型和测量模型使用扩展卡尔曼滤波方法进行更新, 得到特征点的坐标, 通过特征点坐标绘制局部地图进行避障。

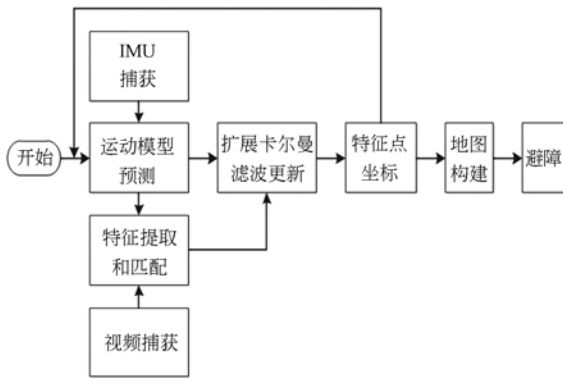


图 2 基于扩展卡尔曼滤波的单目 SLAM 避障算法
Fig. 2 Monocular SLAM obstacle avoidance algorithm based on extended Kalman filter

在结构化室内走廊中,四轴飞行器视野两侧为低纹理的墙壁,而四轴飞行器前方的单目摄像头只能提取走廊的局部信息,所以单目 SLAM 不能通过先建立全局地图信息,再进行避障规划这种方式实现避障。然而,通过设置逆深度方差阈值和摄像头估计距离的阈值,可将四轴飞行器视野中的障碍物从背景中分离出来,从而实现避障功能^[11]。

4 平移光流原理及计算

由于飞行器在室内导航过程中不可避免地会有平移和小角度晃动,这相当于摄像头的平移和晃动动作,这样即使没有障碍物,由于摄像头的平移,相邻两帧图片之间也会产生平移光流。平移光流实际就是第 k 帧图像上像素点在第 $k+1$ 帧图像上的数学投影。

在计算平移光流之前,本文先做一个重要的假设:相机平面与障碍物平面之间没有明显夹角,即认为是相互平行的,误差范围在 10° 以内都不影响对障碍物的检测。鉴于单应性矩阵^[12]携带丰富的运动信息,所以本文使用单应性投影方法计算平移光流,从而求得点在相邻帧图片上的位移,以避免了复杂的迭代过程,同时提高了算法的实时性。

设两帧连续图片 k 和 $k+1$, $p_i(x_k, y_k)$ 是第 k 帧图片上的一个像素点, $p_i(x_{k+1}, y_{k+1})$ 是第一帧图片上的像素点 $p_i(x_k, y_k)$ 经过变换在第二帧图

片上的位置。在单应性投影变换中,像素点都可以用下式表示:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ 1 \end{bmatrix} = H_{k,k+1} * \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix}, \quad (15)$$

其中:

$$H_{k,k+1} = R_{k,k+1} - \frac{T_{k,k+1} \cdot \mathbf{n}^T}{d}, \quad (16)$$

式中: $R_{k,k+1}$ 和 $T_{k,k+1}$ 分别代表第 k 帧和第 $k+1$ 帧图片中该点的旋转分量和平移分量; d 是摄像头和障碍物之间的距离; \mathbf{n}^T 是垂直于摄像头平面的法向量。为了减少计算量,提高计算精度,本文采用 IMU 提供的实时欧拉角信息来计算法向量 \mathbf{n}^T ; $\phi_k, \theta_k, \varphi_k$ 和 $\phi_{k+1}, \theta_{k+1}, \varphi_{k+1}$ 分别是飞行器在第 k 帧图片和第 $k+1$ 帧图片中的姿态角,则根据世界坐标和飞行器坐标的转换关系可得:

$$\mathbf{n}^T = \begin{bmatrix} -\sin \theta_k \\ \sin \phi_k \cdot \cos \theta_k \\ \cos \phi_k \cdot \cos \theta_k \end{bmatrix}, \quad (17)$$

从而得到平移光流向量公式:

$$\mathbf{v}_i = (x_{k+1}, y_{k+1}) - (x_k, y_k). \quad (18)$$

图 3 是计算平移光流的示意图。在图 3 中,假设相机平面始终与物体平面平行,并且只进行平移运动。 $p_i(x_k, y_k)$ 是第 k 帧图上的一个点, $p_i(x_{k+1}, y_{k+1})$ 是使用上面公式在第 $k+1$ 帧图中的点,因此可以得到平移光流向量 \mathbf{v}_i 。

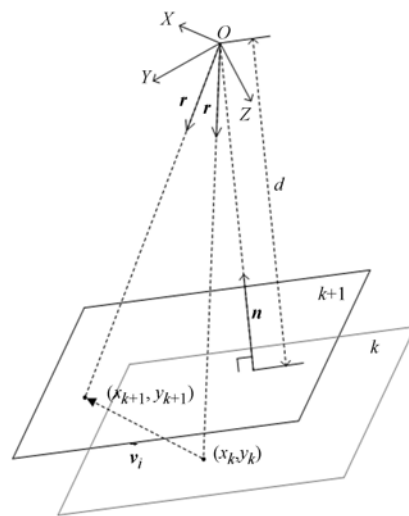


图 3 连续两帧点的平移

Fig. 3 Point translation in two successive frame images

5 融合光流避障算法

融合光流避障算法由障碍物判断和危险区域判断两部分组成。首先,计算检测区域内的融合光流,以判断前方视野内是否有障碍物。然后,根据金字塔 LK 光流长度平均值与前期采样的阈值相比较,来判断飞行器是否进入危险区域。如进入危险区域,则输出一个控制信号,作为 PID 控制输入参数,从而使飞行器做出相应避障动作。

5.1 融合光流向量计算和障碍物判断

为了能从图像特定区域提取出相关融合光流向量,本文使用水平坐标轴和垂直坐标轴对图像进行分区,水平坐标轴和垂直坐标轴的交点就是图像中心,如图 4 所示。由于四轴飞行器对避障算法的实时性要求很高,所以选取以图像中心为中心的大小为 120×120 pixel 的正方形区域为检测区域,这样有效地提高了算法的计算速度。



图 4 检测区域

Fig. 4 Detection region

融合光流的基本原理:

假设场景中存在相对移动的物体,如图 5 所示,图 5 左上部为金字塔 LK 产生的真实光流,其中垂直向上光流为疑似障碍物,左下部为单应性变换得出的平移光流,将二者叠加到同一图像上,就形成融合光流,如图 5 右半部分所示。通过分析这两种光流,可以推导出一种光流的特性。

障碍物上融合光流的夹角(金字塔 LK 光流和平移光流之间的偏移量)明显与其他无障碍部分不同,通过比较偏移量可以推导出一种融合光流的特性。因此,通过建立融合光流和障碍物之间的对应关系,就可以判断障碍。

首先,对单个向量进行滤波,剔除长度过长的

奇异向量;然后,对剩下向量进行 k 均值聚类,剔除各聚类内部奇异点和向量个数过少的聚类。

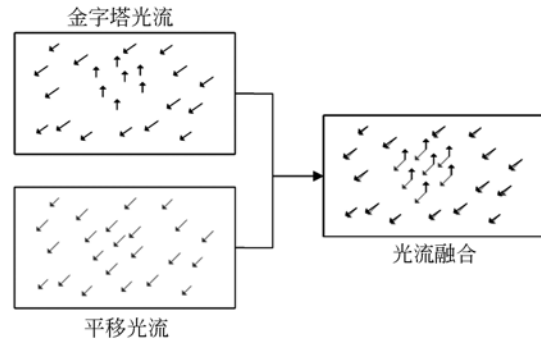


图 5 真实光流和平移光流融合

Fig. 5 Fusion of a real optical flow and a translational optical flow

其次,对融合光流进行滤波,剔除融合光流向量里的奇异向量和夹角过小的向量。最后,对障碍物做出判断。

障碍物判断算法伪代码如下:

```

begin
    变量定义及输入:  $|\theta_r - \theta_l| \Rightarrow \Delta\theta$ ;
                     $30^\circ \Rightarrow \theta$ ;
                     $5 \Rightarrow k$ ; /* 经多次实验聚类个数选择 5 效果最好 */
    if ( $a_i$  and  $\Delta\theta$ ) then
        if ( $\text{Size}(|a_i|) > 0.3 * \text{ImageWidth}$ ) then
            eliminate  $a_i$ ;
            CreateRandomArray( $n, k$ ); /* 随机生成  $k$  个质心向量 */
             $\text{fabs}[a_i(x, y) - j_k(x, y)] \Rightarrow \min$ ;
            /* 计算向量与每个质心向量的距离最小值绝对值 */
            if ( $\text{fabs}[a_i(x, y) - k_i(x, y)] > \min$ ) then
                 $a_i \Rightarrow \text{Cluster}(k)$ ; /* 将  $a_i$  放入距离最短的  $k$  个簇中 */
            end if; /* 剩余向量  $k$  均值聚类结束 */
        end if;
        if ( $|a_i| < |0.3j_k|$ ) then
            eliminate  $a_i$ ; /* 删除偏离质心 30% 的向量 */
        else if ( $n_{k\text{Cluster}} < 5$ ) then
            eliminate Cluster( $k$ );
        end if;
        if ( $\Delta\theta < \theta$  and  $\Delta d_i > d$ ) then
            eliminate  $c_i$ ;
        else if ( $n_c > 8$ ) then
             $1 \Rightarrow \text{Obstacle}$ ; /* 判断为障碍物时, Obstacle
置 1 */
        end if;
    end if;
end;

```

其中: \mathbf{a}_i 为金字塔光流向量,其模为 $|a_i|$; \mathbf{b}_i 为平移光流向量; θ_i 为金字塔光流向量角度; θ_l 为平移光流向量角度; $\Delta\theta$ 为 \mathbf{a}_i 和 \mathbf{b}_i 之间的夹角,阈值为 θ ; \mathbf{c}_i 为融合光流向量; Δd_i 为融合光流向量之间的距离,阈值为 d ; k 为聚类个数; n 为光流总个数; \mathbf{j}_k 为质心向量; \min 为第 i 个向量与每个质心向量距离的最小值; Cluster(k) 为第 k 个簇; $n_{kCluster}$ 为簇内向量个数; n_c 为融合光流向量个数; Obstacle 为障碍物标志位。

5.2 判断飞行器是否在障碍物的危险区域

利用运动视差单目视觉可以判断场景物体与观察者的相对距离^[13-14]。由于观察者观察位置的远近不同,运动物体的视角变化也不同,从而表现为运动速度的差异,具体为:距离观察者远的物体,其视角变化较小,运动速度相对较慢;距离观察者近的物体,其视角变化较大,运动速度相对较快。

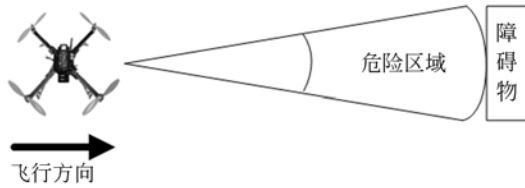


图 6 飞行器与障碍物之间的危险区域

Fig. 6 Dangerous range between quad-rotor helicopter and obstacles

利用近距离物体产生的光流向量相对较大,远距离物体产生的光流向量相对小这一原理,通过计算视野内金字塔 LK 光流向量的平均值来判断飞行器是否进入危险区域。

具体流程如下:

(1) 在飞行器检测区域中,找出当前这帧图像中的强角点集合:

$$C_p = \{C_0, C_1, C_2, \dots, C_n\}, \quad (19)$$

其中: n 为强角点个数, $n \in N$ 。

(2) 通过金字塔 LK 算法求出强角点集合 C_p 的光流向量集合:

$$d_p = \{(p_{s0}, p_{e0}), (p_{s1}, p_{e1}), \dots, (p_{sn}, p_{en})\}, \quad (20)$$

其中: d_p 为视频检测区域的光流向量集合, (p_{si}, p_{ei}) 为每一个光流向量的起始点和结束点;

(3) 计算每个光流向量的大小:

$$m_i = \sqrt{(p_{eix} - p_{six})^2 + (p_{eiy} - p_{siy})^2}, \quad (21)$$

其中: $i \in [0, n]$, (x, y) 是第 i 个强角点的坐标。

(4) 将光流向量大小 $m_i > 0.3 \cdot Image-Width$ 的奇异点从光流向量集合 d_p 中剔除。

(5) 计算检测区域内剩余光流向量模的均值:

$$\bar{m} = \frac{m_0 + m_1 + \dots + m_n}{n}. \quad (22)$$

(6) 判断前方是障碍物时,定时 2 ms,将 2 ms 时刻的检测区内光流向量均值赋给变量 OBS,将 OBS 作为采样阈值。

(7) 设 $T = \bar{m}$, \bar{m} 为检测区域内实时光流向量的均值,当 $T \geq 3OBS$ 时,认定飞行器进入障碍物的危险区域,如果此时不进行飞行姿态调整,飞行器将来不及避障。

6 飞行器避障策略实验及仿真分析

6.1 系统工作流程及硬件平台介绍

本文提出的避障算法是在一台四轴飞行器上实现的,如图 7 所示。该飞行器使用 APM2.5 自动驾驶仪实现自动控制,使用 3DR Radio Telemetry Kit 进行实时飞行数据传输,使用 1.2 GHz Light Weight Video Transmitter Kit 进行视频图像的实时传输。

如图 7 所示,飞行器与地面站(PC)之间可以进行实时的数据和图像传输。由于飞行器硬件资源有限,为了提高飞行器的实时性,图像处理不在飞行器上进行,而是将视频流通过 Video Transmitter Kit 传输给地面站(PC),在地面站(PC)上进行图像处理。然后,根据图像处理结果将 PID

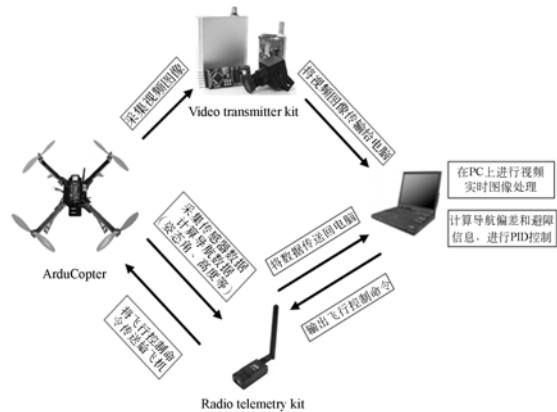


图 7 系统工作流程

Fig. 7 System workflow

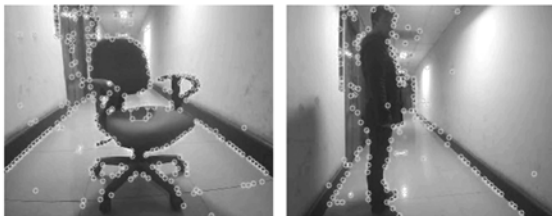
控制输入参数通过 Radio Telemetry Kit 传输给飞行器,实现四轴飞行器的实时避障动作。

飞行器的主要参数如下:最大时速为 14 km/h;最大飞行时间为 20 min;采用 GPS、CCD 摄像头、6 轴陀螺仪,以及超声波传感器。

PC 平台:CPU Intel 酷睿 2 P8600 (2.4G); 2G 内存; Windows XP 操作系统; Arduino 编程环境。

6.2 实验及仿真分析

本章将在实际场景中验证本文所提避障算法的可行性和有效性,并对算法的实时性和准确性进行仿真分析。文中算法使用 Shi-Tomasi^[15]方法进行角点检测,分别对椅子和行人进行检测,检测结果如图 8 所示。



测试 1 测试 2
Test 1 Test 2
图 8 Shi-Tomasi 角点检测结果

Fig. 8 Detection results of Shi-Tomasi corner method

从图 8 测试 1 和测试 2 中可以看出,Shi-Tomasi 角点检测方法可以很好地检测出前方物体的轮廓角点,这将为准确计算检测区域内的光流提供基础。

6.2.1 飞行速度与准确率之间的关系

令飞行器以不同飞行速度对单一障碍物进行判断,共做 15 次相同实验,飞行速度与准确率之间的关系如图 9 所示。其中,横轴为四轴飞行器的飞行速度,飞行器最大飞行速度为 3.8 m/s。纵轴为 15 次实验中成功判断出障碍物的准确率。

本组实验都是在飞行过程中进行的,所以当四轴飞行器没有移动时,成功率为零。飞行速度越慢,准确率越高,反之亦然。为了兼顾室内空间局限性和准确率,本文选取单一障碍物判断准确率为 100%时的飞行速度 0.4 m/s 作为飞行器的速度。

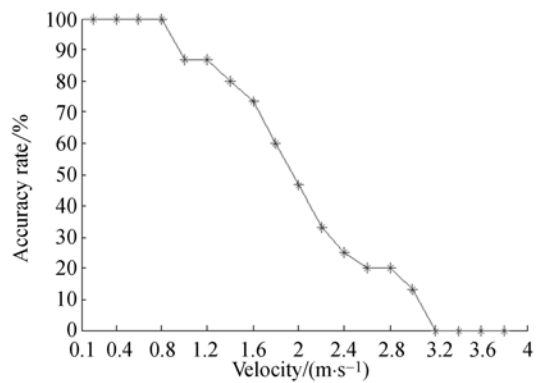


图 9 飞行速度与准确率之间的关系

Fig. 9 Relationship between velocity and accuracy rate

6.2.2 避障算法的可行性和有效性验证

设计实验如下:在一光照充足的走廊中间有一行人充当障碍物,飞行器以 0.4 m/s 的速度由远及近地飞向行人。

如图 10(a)所示,飞行器检测到前方没有障碍物。图 10(a)水平线和垂直线的交点为画面的中心点,中心点外 120 pixel × 120 pixel 的区域为检测区域。箭头表示检测出的金字塔光流向量,方向如图中箭头所示。OBS 为采样阈值,取图 10(a)中光流向量的定时均值作为检测区域内光流向量的实时均值。

图 10(b)(彩图见期刊电子版)表示检测出前方有障碍物,但是还没有进入危险区域,不需要做避障动作。其中,黑色向量表示平移光流,白色向量是需要剔除的奇异向量。检测到障碍物后, OBS 值为 2.01, T 值为 5.43。

图 10(c)中, $T \geq 3OBS$,则可以判断出飞行器已经进入危险区域,屏幕左下方出现 Danger 字样。此时,飞行器做出避障动作。

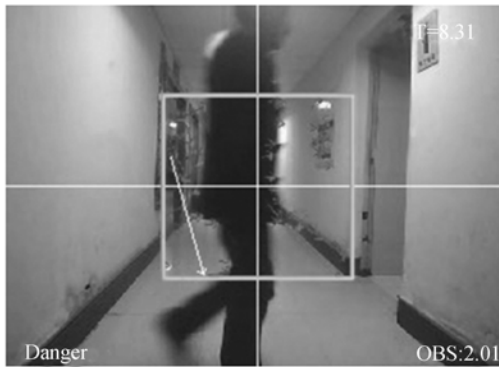


(a) 前方无障碍检测结果

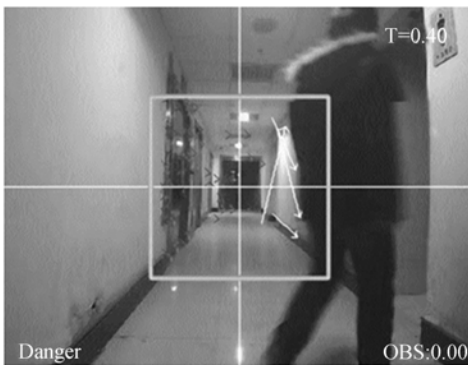
(a) Detection results without obstacles in front



(b) 前方有障碍检测结果
(b) Detection results with obstacles in front



(c) 危险区域检测结果
(c) Detection results of dangerous range



(d) 避开障碍物,视野内均值趋于 0 的结果
(d) Results of avoiding obstacles with mean in view tending to 0

图 10 避障算法验证

Fig. 10 Validation results of obstacle avoidance algorithms

图 10(d)中,飞行器做出避障动作后,障碍物已经不在飞行器检测视野内,Danger 字样消失。此时,OBS 为 0,T 值趋于 0。

6.2.3 对避障算法的实时性进行验证

为了保证实时性,减少金字塔 LK 算法的计算量和亮度变化对光流计算的影响,本算法在进行视频流处理之前,先对图像进行灰度化和高斯滤波去噪等预处理。

采集并记录四轴飞行器飞行过程中连续 200 帧的图片,分别用融合光流避障算法、Horn-Schunck 光流避障算法和单目 SLAM 避障算法对这 200 帧图片进行处理,结果如图 11 所示(彩图见期刊电子版)。图 11 中,横轴为图像帧数,纵轴为每帧图片执行时间。蓝色曲线表示融合光流避障算法的执行时间,红色曲线表示单目 SLAM 避障算法的执行时间,绿色曲线表示为 Horn-Schunck 光流避障算法的执行时间。

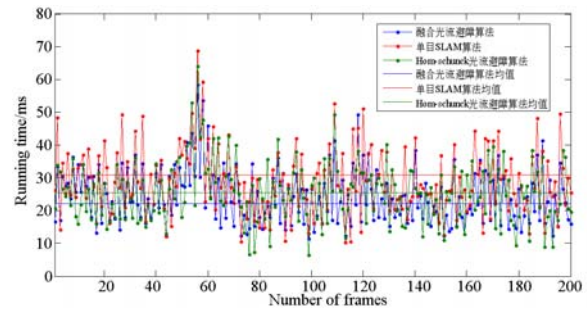


图 11 实时性的比较

Fig. 11 Comparison of real-time ability

避障算法执行时间为:

$$T = T_1 + T_p + T_r, \quad (23)$$

其中: $T_1 = 33.33 \text{ ms}$ 为从飞行器传输 1 帧视频图像到地面站的时间; T_p 为视频在地面站进行处理并给出控制命令时间; $T_r = 10 \text{ ms}$ 为地面站向飞行器传回一个控制命令响应的的时间。

本次实验中,系统每采集并处理 5 帧图片,就可以发送一个控制信号,以便确定飞行器是否需要避障动作。因此,融合光流避障算法、Horn-Schunck 光流避障算法和单目 SLAM 避障算法的 5 帧图片的总执行时间分别为:

$$T_F = T_1 \times 5 + T_p \times 5 + T_r = 287.45 \text{ ms}, \quad (24)$$

$$T_{HS} = T_1 \times 5 + T_p \times 5 + T_r = 303.75 \text{ ms}, \quad (25)$$

$$T_S = T_1 \times 5 + T_p \times 5 + T_r = 331.85 \text{ ms}. \quad (26)$$

融合光流避障算法的平均执行时间比 Horn-Schunck 光流避障算法快 16.3 ms,比单目 SLAM 避障算法快 44.4 ms。显然,融合光流避

障算法的平均执行时间比同类 Horn-Schunck 光流避障算法和单目 SLAM 避障算法的执行时间都短,这说明融合光流避障算法在相似场景下可以更快地输出避障动作控制信号。这对于需要在室内小空间进行实时避障的四轴飞行器来说,实时性就更好。

6.2.4 避障算法的准确性验证

在飞行速度为 0.4 m/s 的情况下,对不同个数的障碍物分别进行 15 次实验,实验结果如图 12 所示(彩图见期刊电子版)。

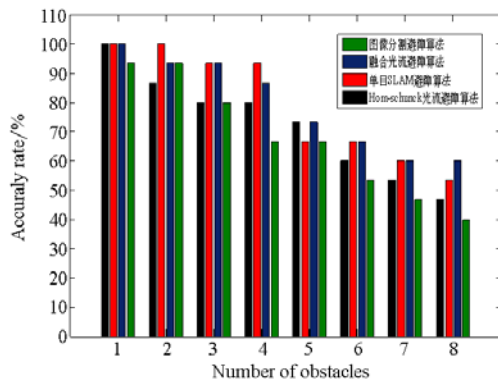


图 12 准确率的比较

Fig. 12 Comparison of accuracy rate

横轴为障碍物个数,纵轴为判断的准确率。该准确率为识别率,即识别出障碍物个数占总障碍物个数的百分比,多次实验取平均值。比如,障碍物个数为 5 时,每次试验判断出障碍物个数占 5 个障碍物个数的百分比,然后求 15 次实验的平均值。在图 12 中,黑色为 Horn-Schunck 光流避

障算法的准确率;红色为单目 SLAM 避障算法的准确率;蓝色为融合光流避障算法的准确率;绿色为图像分割避障算法的准确率。从图中可以看出,在障碍物个数逐渐增加的情况下,融合光流避障算法的准确率明显高于图像分割避障算法和同类型的 Horn-Schunck 光流避障算法;而在障碍物个数小于等于 4 的情况下,融合光流避障算法的准确率与单目 SLAM 避障算法相当;在障碍物个数大于 4 的情况下,融合光流避障算法的准确率要高于单目 SLAM 避障算法。

通过上述实验可知,本文提出的融合光流避障算法在实际工程中可以很好地进行障碍物的判断和避障,从而验证了本算法的可行性。另外,通过对算法执行时间和准确性的测试也验证了算法的较高的实时性和准确性。

7 结 论

本文针对在实际平台 ArduCopter 四轴飞行器上单目视觉避障算法实时性不够,准确率不高的问题,提出并实现了一种融合光流避障算法。从实验结果可以看出,与同类光流避障算法和同类视觉避障算法相比,在使用单一低分辨率摄像头的情况下,四轴飞行器能够快速判断障碍物,并做出实时避障动作,实时性很好,并且在多障碍物的避障过程中,准确率也较高。本文提出的避障算法为未来四轴飞行器在复杂环境中实时避障提出了一种有效的解决方案。

参考文献:

- [1] GRZONKA S, GRISETTI G, BURGARD W. A fully autonomous indoor quad-rotor [J]. *IEEE Transactions On Robotics*, 2012, 28(1): 90-100.
- [2] HERISSE B, HAMEL T, MAHONY R. Russotto F-X. landing a VTOL unmanned aerial vehicle on a moving platform using optical flow[J]. *IEEE Robotics and Automation Society*, 2012, 28(1): 77-89.
- [3] AYDIN E, LU N I, EFE M Ö. Autonomous quad-rotor flight with vision-based obstacle avoidance in virtual environment [J]. *Expert Systems With Applications*, 2012, 39: 894-905.
- [4] 刘正东,高鹏,杨静宇.一种用于道路避障的双目视觉图像分割方法[J]. *计算机应用研究*, 2005, 4: 249-251.
- [5] LIU ZH D, GAO P, YANG J Y. Binocular image segmentation algorithm for obstacles avoidance[J]. *Application Research of Computers*, 2005, 4: 249-251.
- [6] KENDOUL F, FANTONI A, NONAM A. Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles [J]. *Robotics and Autonomous Systems*, 2009, 57(6-7): 591-602.
- [7] GRIGORESCU S M, MACESANU G, COCIAS T T, et al. Robust camera pose and scene structure

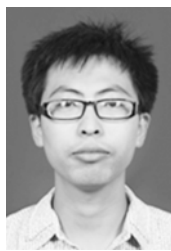
- analysis for service robotics [J]. *Robotics and Autonomous Systems*, 2011, 59(11): 889-909.
- [7] 陈向坚,李迪,续志军,等. 四旋翼微型飞行器的区间二型模糊神经网络自适应控制[J]. *光学精密工程*, 2012, 20(6): 1335-1341.
CHEN X J, LI D, XU ZH J, *et al.*. Adaptive control of Quad-rotor MAV using interval type-II fuzzy neural network [J]. *Opt. Precision Eng.*, 2012, 20(6): 1335-1341. (in Chinese)
- [8] 陈向坚,李迪,白越,等. 模糊神经网络在自适应双轴运动控制系统中的应用[J]. *光学精密工程*, 2011, 19(7): 1643-1650.
CHEN X J, LI D, BAI Y, *et al.*. Application of type-II fuzzy neural network to adaptive double axis motion control system [J]. *Opt. Precision Eng.*, 2011, 19(7): 1643-1650. (in Chinese)
- [9] VIET VO, NGOC LY. An effective approach for human actions recognition based on optical flow and edge features [C]. *ICCAIS2012, Ho Chi Minh City*, 2012: 24-29.
- [10] DAVISON A J, REID L D, MOLTON N D, *et al.*. MonoSLAM: Real-time single camera SLAM [J]. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 2007, 29(6): 1052-1067.
- [11] EINHORN E, SCHRÖTER CH, GROSS H M. Attention-driven monocular scene reconstruction for obstacle detection, robot navigation and map building [J]. *Robotics and Autonomous Systems*, 2011, 59: 296-309.
- [12] HARTLEY R, ZISSERMAN A. *Multiple View Geometry in Computer Vision* [M]. Cambridge: Cambridge University Press, 2004.
- [13] STEFANOSKI N, WANG O, LANG M, *et al.*. Automatic view synthesis by image-domain-warping [J]. *IEEE Signal Processing Society*, 2013, 22(9): 3329-3341.
- [14] 夏军营,徐小泉,熊九龙. 利用平行透视投影模型的位姿迭代估计[J]. *光学精密工程*, 2012, 20(6): 1345-1349.
XIA J Y, XU X Q, XIONG J L. Iterative pose estimation using paraperspective camera model [J]. *Opt. Precision Eng.*, 2012, 20(6): 1345-1349.
- [15] SMITH M J, BOXERBAUM A, PETERSON G L, *et al.*. Electronic image stabilization using optical flow with inertial fusion [C]. *IROS 2010, Taipei*, 2010: 1146-1153.
- [16] MAHONY R, KUMAR V, CORKE P. Multirotor aerial vehicles: modeling, estimation, and control of Quadrotor [J]. *IEEE Robotics & Automation Magazine*, 2012, 19(3): 20-32.

作者简介:



赵海(1959—),男,辽宁沈阳人,博士,教授,博士生导师,主要从事:互联网宏观拓扑特征复杂性、软件网络复杂性、无线传感器网络、信息/数据融合、普适计算、虚拟现实与计算机仿真、图像处理、嵌入式技术和物联网等方面的研究与教学工作。E-mail: zhaoh@mail.neu.edu.cn

通讯作者:



陈星池(1987—),男,辽宁绥中人,博士研究生,2010年于沈阳化工大学获得学士学位,主要从事机器视觉导航和避障方面的研究。E-mail: chenxingchi999@126.com

(版权所有 未经许可 不得转载)