doi: 10.7690/bgzdh.2015.11.003

基于灰度投影的跑道线提取方法

孙煜杰，杨　欢，吴政隆，李　杰
（北京理工大学机电学院，北京 100081）

摘要：针对无人机着陆时采用传统的 Hough 变换或者利用跑道区域先验信息的方法所产生的计算时间开销较大和未知先验信息前提下不适用等问题，提出一种基于灰度投影的跑道线检测算法。通过模板匹配提取跑道区域作为兴趣区(region of interest，ROI)，在 ROI 中进行边缘提取；对于边缘提取后的图像使用灰度投影算法，获得可能的直线在空间内的位置，并使用 K-means 算法对可能的直线进行聚类，从而获得跑道边线的估计位置。仿真结果表明：该算法可以有效提取跑道边线，相比于传统 Hough 变换的直线提取算法，可以减少 50%的时间消耗。

关键词：跑道线提取；灰度投影；线检测；K-means 聚类算法
中图分类号：TJ86　　**文献标志码**：A

Runway Line Extraction Method Based on Grey Projection

Sun Yujie, Yang Huan, Wu Zhenglong, Li Jie
(*School of Mechatronical Engineering*, *Beijing Institute of Technology*, *Beijing* 100081, *China*)

Abstract: Use Hough transform or runway line prior information method will cause large cost of computation time or unsuitable under unknown prior information premise, put forward runway extraction method based on grey projection. Use template matching to extract runway area as region of interest (ROI), and carry out edge exaction in ROI. After edge extraction, use grey projection method to acquire probable line position in space, then use K-means method to carry out clustering for probable line to find out the estimation position of runway boundary. The simulation results shows that the method can effectively extract runway boundary, while it can reduce 50% of computing time compared with Hough transform.

Keywords: runway extraction; rotating projection method; line detection; K-means clustering algorithm

0　引言

基于图像的自动降落技术是微小型无人飞行器(unmanned aerial vehicle，UAV)的关键技术之一，通过应用这项技术，无人机可以实现自主着陆。由于这项技术依赖地面标志物的提取，同时机场跑道线可称得上无人机着陆阶段最明显的标志物，所以对于机场跑道线的提取已经成为无人机着陆阶段控制领域的热门研究方向之一。现有 2 种传统的提取机场跑道线方法：基于多源图像和地理信息(GIS)的方法[1-3]和基于平行关系检测和 Hough 变换的方法[4-6]。但上述算法多数是基于卫星图像和航拍图像而设计，很少考虑到无人机实际的视场。多数情况下，在无人机着陆阶段的视场(field of view，FOV)中，跑道区域通常呈梯形分布，使得其很难满足平行关系约束。文献[7]均采用了 Hough 变换用于提取跑道，Hough 变换的主要问题在于计算时间开销较大，不利于实时计算。文献[8]提出了一种利用了跑道区域的先验信息的算法，但这种算法在未知先验信息的前提下并不适用，这些先验信息包括跑道内部和外部投影的对比信息。文献[9]提出了一种应用平行检测的试探方式的搜索来进行的跑道线提取方法，但这种方法只能应用于俯视的视角，而俯视视角在着陆阶段是不存在的。

为了避免上述问题，笔者提出了一种基于灰度投影的跑道线检测算法，通过模板匹配提取跑道区域作为兴趣区(region of interest，ROI)，之后，计算图像灰度在各个方向上的投影，并使用 K-means 聚类提取出跑道边线所在的各个直线。仿真结果表明：该算法能够有效提取跑道的边缘线，相比基于 Hough 变换的算法，可以减少 50%的时间消耗。

1　跑道兴趣区(ROI)提取与边缘检测

当飞行器进近时，跑道区域可以被看作一个由四条边线组成的四边形，对于跑道区域的提取遵循以下步骤：1) 确定兴趣区；2) 提取边缘；3) 计算在图像坐标系下的直线方程。

兴趣区的提取主要是为了规避跑道提取时复杂的全局计算。笔者使用模板匹配的方法来提取图像中的跑道兴趣区。定义模板图像为 $T(x_t, y_t)$，原始图

收稿日期：2015-07-02；**修回日期**：2015-08-22
基金项目：十二五国防基础科研项目(B222013××××)
作者简介：孙煜杰(1990—)，男，陕西人，在读硕士，从事飞行器导航与控制研究。

像为 $S(x, y)$，令模板图像 T 遍历原始图像 S，并在每一点计算归一化互相关值 (normalized cross correlation，NCC) 作为置信度[10]，如下式

$$R_{\mathrm{NCC}}(x,y)=\frac{\sum_{x_t,y_t}(T(x_t,y_t)\times S(x+x_t,y+y_t))}{\sqrt{\sum_{x_t,y_t}T(x_t,y_t)^2\sum_{x_t,y_t}S(x+x_t,y+y_t)^2}}。\quad (1)$$

其中 R_{NCC} 给出了模板图像和原始图像之间的匹配程度。

对于 ROI 内的图像，使用 Sobel 算子进行边缘提取，如下式：

$$\boldsymbol{G}_x=\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix}\times\boldsymbol{S},\ \boldsymbol{G}_y=\begin{bmatrix}1&2&-1\\0&0&0\\-1&-2&-1\end{bmatrix}\times\boldsymbol{S};\quad (2)$$

$$G=\sqrt{\boldsymbol{G}_x^2+\boldsymbol{G}_y^2},\ \theta=\arctan(\frac{\boldsymbol{G}_x}{\boldsymbol{G}_y})。\quad (3)$$

其中：$\boldsymbol{S}$ 是源图像；$\boldsymbol{G}_x$ 和 $\boldsymbol{G}_y$ 是图像在水平和垂直 2 个方向上的梯度分量；G 是图像的梯度幅值；θ 是梯度方向。设定检测阈值 T，当满足 $G>T$ 时，即可认为是边缘区域。

2 基于灰度投影的直线检测

2.1 灰度投影的定义

灰度投影是一种常用的灰度图像处理方法，通过计算灰度图像轴向和列向的灰度和获取图像灰度分布的信息，从而将一个二维矩阵转化为 2 条特征向量。

对于 $i\times j$ 的源图像

$$\boldsymbol{S}=\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1j}\\a_{21}&&\cdots&a_{2j}\\\vdots&\vdots&&\vdots\\a_{i1}&a_{i2}&\cdots&a_{ij}\end{bmatrix}。\quad (4)$$

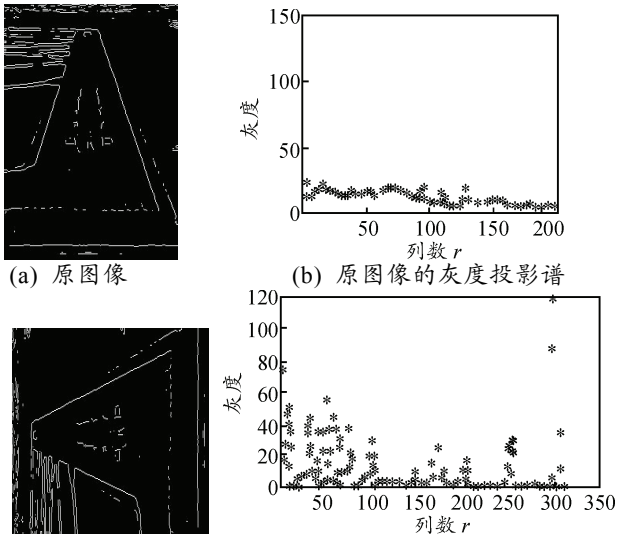
则其行向灰度投影向量为

$$\boldsymbol{GP}_{\mathrm{row}}=[\sum_j a_{1j},\sum_j a_{2j},\cdots,\sum_j a_{ij}]。\quad (5)$$

列向灰度投影向量为

$$\boldsymbol{GP}_{\mathrm{column}}=[\sum_i a_{i1},\sum_i a_{i2},\cdots,\sum_i a_{ij}]。\quad (6)$$

所谓旋转投影是指首先将图像逆时针旋转 θ，之后再计算图像列向灰度投影向量 $\boldsymbol{GP}_{\mathrm{column}}$ 作为图像在该旋转角度下的投影谱。当图像中的直线边缘通过旋转与水平方向垂直时，会在灰度投影谱上形成一个峰值点，如图 1 所示。可以看出：在图 1(d)中存在着不同的峰值点，这些峰值点与旋转图像中的直线存在着对应关系，根据这个对应原理即可完成直线的检测。



(a) 原图像 (b) 原图像的灰度投影谱

(c) 原图像逆时针旋转 90°图像 (d) 旋转后图像灰度投影谱

图 1 旋转图像与其灰度投影谱

定义 $\Re^{\theta}$ 为旋转算子，表征将矩阵逆时针方向旋转 θ，对于形如图 1(b)的图像，其投影灰度谱函数可以计为

$$GP_{\mathrm{c}}(\theta,r)=\sum_i\Re^{\theta}(a_{in})。\quad (7)$$

其中：θ 为图像逆时针旋转角度；$GP_{\mathrm{c}}(\theta,r)$ 为图像第 r 列列向的灰度和。

GP_{c} 相当于权值，GP_{c} 越大，表示图像在逆时针旋转 θ 后，r 处竖直方向上的边缘点越集中，也越可能存在一条直线。将源图像沿逆时针，按照步长 k 进行旋转直到 π，可以获得一系列角度下的灰度投影谱，设直线检测阈值为 T，对于满足

$$GP_{\mathrm{c}}(\theta,r)>T\quad (8)$$

的投影点 (θ,r)，及视其对应的直线为潜在直线。

2.2 改进 K-means 聚类算法

由于图像中直线宽度一般大于一个像素，特别是 T 取值较小，式 (8) 的约束比较宽松的情况下，会出现多个临近峰值对应同一条直线的情况。因此需要对于检测出的潜在直线进一步筛选，笔者采用了一种改进的 K-means 聚类算法来完成。

K-means 算法是一种得到最广泛使用的基于划分的聚类算法，通过把 n 个对象分为 k 个簇，以使簇内具有较高的相似度达到聚类划分。构建相似度函数来评估簇内元素的相似度水平，一般通过簇内

对象的平均值来进行。

传统的 K-means 聚类算法采用对象与聚类中心的距离函数作为各个对象的相似度函数，并通过簇内对象的欧式距离平均值来确定聚类中心。在文中，由于待聚类对象有 3 个参数 $(\theta,r,GP_c(\theta,r))$，但又并不相对独立，因此不适于作为一个传统三维的聚类问题。因此笔者提出了一种带权重的改进 K-means 聚类算法如下：

在 (θ,n) 空间中：定义待聚类对象为 $N_i=(\theta_i,r_i)$，其中 $i=1,2,\cdots,n$；待划分的簇为 S_j，其中 $j=1,2,\cdots,k$；定义各簇的初始聚类中心为 $C_j=(\tilde{\theta}_j,\tilde{r}_j)$，$j=1,2,\cdots,k$；定义对象 N_i 与簇 S_j 的相似度函数

$$V_{ij}=\frac{GP_c(\theta_i,r_i)}{\left\|N_i-C_j\right\|^2}。\tag{9}$$

$\forall i\in[1,n]$，$\exists\mu\in[1,k]$，若满足

$$V_{i\mu}=\min\left\{\frac{GP_c(\theta_i,r_i)}{\left\|N_i-C_j\right\|^2}\middle|j\in[1,k]\right\},\tag{10}$$

则有 $N_i\in S_\mu$，其中 $\mu=1,2,\cdots,k$，即完成初始聚类。

聚类中心的更新算法如下：

定义　$S_\mu=\{\tilde{N}_1,\tilde{N}_2,\cdots,\tilde{N}_m\}$，则聚类中心 $C_\mu=(\tilde{\theta}_\mu,\tilde{r}_\mu)$ 为

$$C_\mu=\frac{\sum_{i=1}^{m}[GP_c(\tilde{N}_i)\times\tilde{N}_i]}{\sum_{i=1}^{m}GP_c(\tilde{N}_i)}=\left(\frac{\sum_{i=1}^{m}[GP_c(\tilde{\theta}_i,\tilde{r}_i)\times\tilde{\theta}_i]}{\sum_{i=1}^{m}GP_c(\tilde{\theta}_i,\tilde{r}_i)},\frac{\sum_{i=1}^{m}[GP_c(\tilde{\theta}_i,\tilde{r}_i)\times\tilde{r}_i]}{\sum_{i=1}^{m}GP_c(\tilde{\theta}_i,\tilde{r}_i)}\right)。\tag{11}$$

获取的聚类中心即可作为直线的位置，其中 θ_μ 是源图像逆时针旋转的角度，n_μ 是直线所在的列。

2.3　直线参数求解

获取聚类中心之后，笔者建立如图 2 所示的坐标系系统，求解图像中的直线参数。

设原图像坐标系为 XOY，旋转后生成的图像坐标系为 $X_WO_WY_W$，图像的旋转中心处建立旋转坐标系 XOY。设原图像上任意一点 p，旋转后位于 p_1。原图像长和宽分别为 W 和 H，旋转后为 W_W 和 H_W。

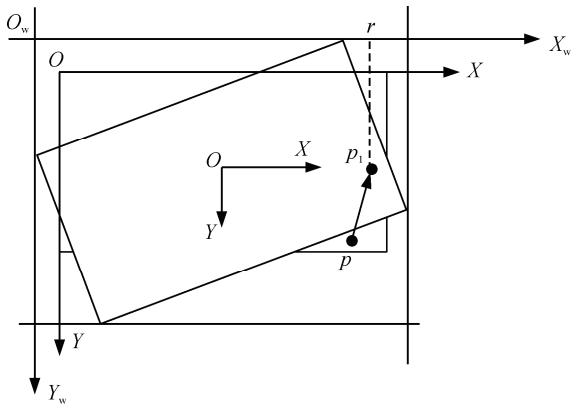


图 2　旋转图像坐标变换

设在 XOY 坐标系下，p 点坐标为 (X_0,Y_0)，在坐标系 xoy 下，P 点坐标为 (x_0,y_0)，根据坐标系的变换关系，存在：

$$\begin{bmatrix}x_0\\y_0\end{bmatrix}=\begin{bmatrix}X_0\\Y_0\end{bmatrix}-\begin{bmatrix}\frac{w}{2}\\\frac{h}{2}\end{bmatrix}。\tag{12}$$

逆时针旋转 θ 后，得到 p_1 在 xoy 坐标系下的坐标设为 (x_1,y_1)。根据旋转矩阵的关系，有

$$\begin{bmatrix}x_1\\y_1\end{bmatrix}=\begin{bmatrix}\cos\theta&-\sin\theta\\\sin\theta&\cos\theta\end{bmatrix}\begin{bmatrix}x_0\\y_0\end{bmatrix}。\tag{13}$$

点 p_1 在 $X_WO_WY_W$ 坐标系下坐标 (x_W,y_W) 为：

$$\begin{bmatrix}x_W\\y_W\end{bmatrix}=\begin{bmatrix}x_1\\y_1\end{bmatrix}+\begin{bmatrix}\frac{W}{2}\\\frac{H}{2}\end{bmatrix}。\tag{14}$$

已知 p_1 在 $X_WO_WY_W$ 坐标系下的横坐标 r，则

$$\begin{bmatrix}x_W-\frac{W}{2}\\y_W-\frac{H}{2}\end{bmatrix}=\begin{bmatrix}\cos\theta&-\sin\theta\\\sin\theta&\cos\theta\end{bmatrix}\begin{bmatrix}X_0-\frac{w}{2}\\Y_0-\frac{h}{2}\end{bmatrix}。\tag{15}$$

$x_W=r$，代入上式即得

$$Y_0=\cot\theta X_0-\frac{w}{2}\cot\theta+\frac{h}{2}-\frac{r-W/2}{\sin\theta}。\tag{16}$$

此即为点 p_1 所在直线方程。

单帧图像的算法整体流程如图 3 所示。

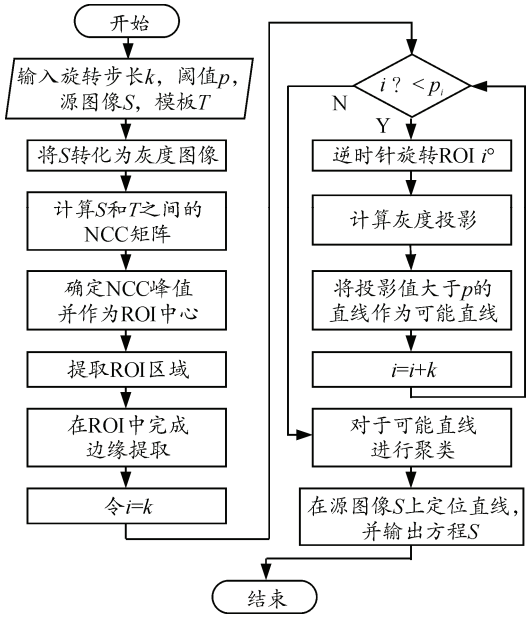


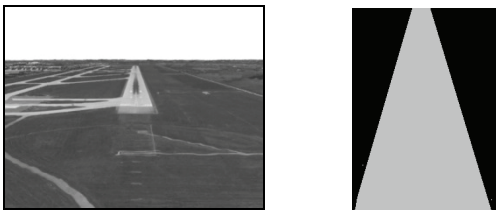
图 3　基于视觉的跑道提取算法架构

3　算法仿真

仿真所使用的图像序列由慕尼黑工业大学开发的 OSGVisual 飞行器可视化工具生成，笔者选取其中部分图像序列说明该算法的工作过程。

3.1　ROI 提取仿真

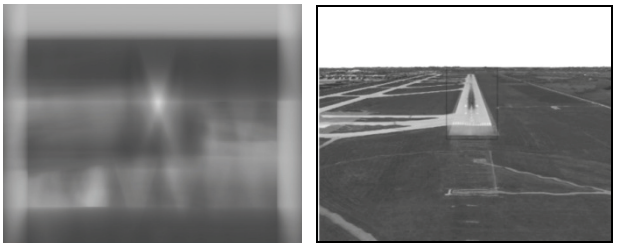
源图像使用图像序列中的第 100 帧图像，如图 4(a)所示。笔者使用图 4(b)作为人工模板，对源图像进行匹配。



(a) 源图像　　(b) 匹配模板

图 4　第 100 帧图像和匹配模板

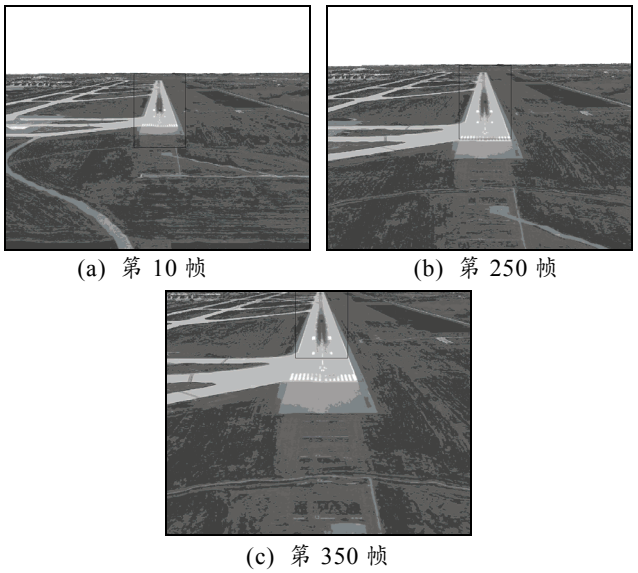
完成对源图像 $\boldsymbol{S}$ 的遍历后，归一化互相关矩阵如图 5(a)所示，以其峰值作为 ROI 中心点，如图 5(b)。



(a) 归一化互相关矩阵图　　(b) 提取出的 ROI

图 5　ROI 提取结果

为了检测算法的在多尺度下变换的性能，笔者分别对于第 10 帧，第 250 帧和第 350 帧图像进行仿真，如图 6 所示。

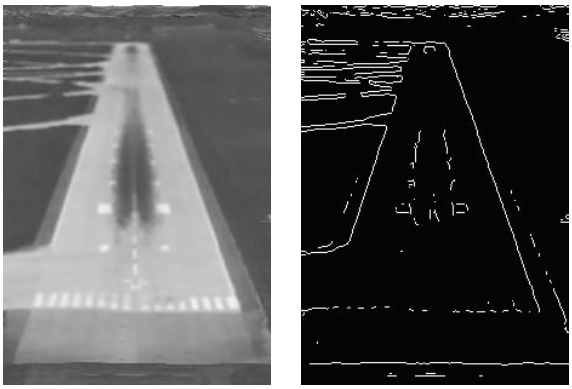


(a) 第 10 帧　　(b) 第 250 帧

(c) 第 350 帧

图 6　不同尺度图像下的 ROI 提取结果

图 6 结果显示，面对跑道区域尺度的变化，本匹配模板具有一定的多尺度匹配能力。跑道兴趣区域(ROI)的边缘提取结果如图 7 所示。



(a) 跑道兴趣区域(ROI)　　(b) ROI 边缘提取结果

图 7　ROI 和其边缘提取结果

3.2　基于灰度投影的线提取仿真

按照图 3 所示的流程图进行仿真，其中相关参数为 T=50，k=2°。运行旋转投影算法后，可能的直线在 (θ,r) 空间的分布如图 8 所示。

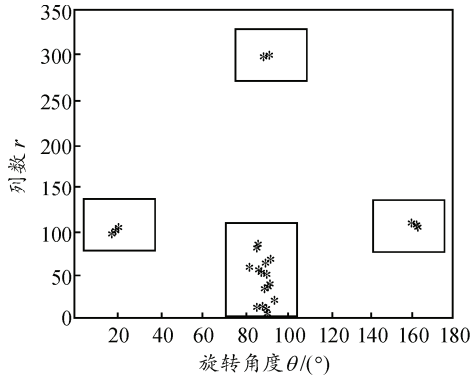


图 8　可能直线在 (θ,r) 空间内的分布

在图 8 中可以看到：ROI 中的 4 条典型直线(起始线，终止线，左边线，右边线)对应在 (θ,r) 空间内，表现为 4 个分布区域，用 4 个矩形框表示。使用 K-means 算法对图 8 进行聚类，计算聚类中心作为直线实际位置，如图 9 所示。

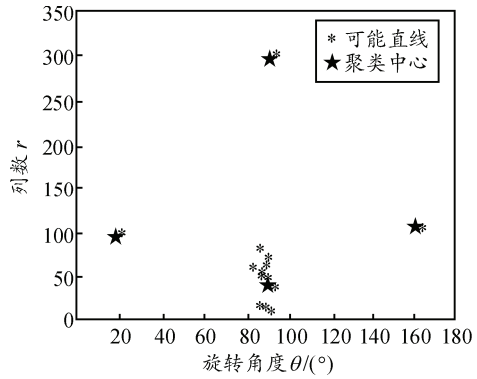


图 9　使用 K-means 算法得到的聚类中心分布

获得的聚类中心矩阵为

$$\text{Cluster\_center}=\begin{bmatrix}89.156\,4 & 39.278\,3\\ 16.500\,0 & 96.000\,0\\ 90.271\,9 & 298.345\,8\\ 160.735\,9 & 107.143\,0\end{bmatrix}。\quad(17)$$

获得的直线方程为

$$\left.\begin{aligned}y&=-0.014\,7x-276.607\,8\\ y&=-3.375\,9x+24.150\,7\\ y&=0.004\,7x-17.678\,9\\ y&=2.861\,3x-622.775\,7\end{aligned}\right\}。\quad(18)$$

跑道线提取结果如图 10 所示。

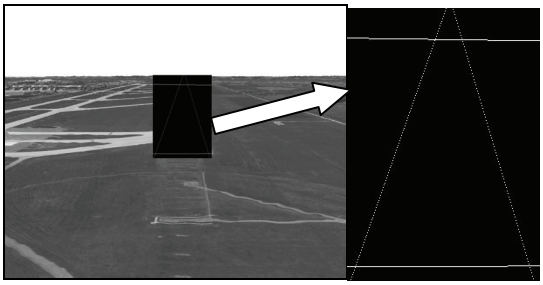


图 10　跑道线提取结果

Matlab 平台下，笔者对大小为 1 024×768 的 300 帧图像进行处理，结果显示：使用本节算法平均每帧计算耗时 0.299 7 s，而传统使用 Hough 变换算法在相同条件下计算耗时则达到 0.634 s，因此本节所述算法比起传统算法在时间上更有优势。

4　结论

笔者给出了一种基于灰度投影的跑道线提取方法。该算法首先使用模板匹配获得跑道区域，作为 ROI，之后在 ROI 中进行边缘提取。对于边缘提取后的图像使用灰度投影算法，获得可能的直线在 (θ,r) 空间内的位置，之后使用 K-means 算法对可能的直线进行聚类，从而获得跑道边线的估计位置。仿真结果表明：该算法可以有效提取跑道边线，同时相比于传统 Hough 变换的直线提取算法，时间开销可降低 50%。

参考文献：

[1] Pascal Dryts, Wim Mees, Dirk Borghys, et al. SAHARA: Semi-automatic help for region analysis[C]. Proceeding of the Joint workshop of ISPRS Working Groups I/1, I/3 and IVA: Sensors and mapping from Space. Germany Hannover, 1997: 267-274.

[2] Michela A. Airport detection using a simple model, multisource images and altimetric information[J]. SPIE, 1998(2315): 604-615.

[3] Huertas A, Cl W, Nevatia R. Detecting runways in complex airport scenes[J]. Computer Vision, Graphics, and Image Processing, 1990, 51(2): 107-145.

[4] 何勇, 徐新, 孙洪, 等. 机载 SAR 图像中机场跑道的检测[J]. 武汉大学学报(理学版), 2004, 50(3): 393-396.

[5] 鲍复民, 李爱国, 覃征. 合成孔径雷达图像中机场跑道的自动识别[J]. 西安交通大学学报, 2005, 38(12): 1243-1246.

[6] 贾承丽, 周晓光, 计科峰, 等. 复杂 SAR 场景中机场跑道的提取[J]. 信号处理, 2007, 23(3): 374-378.

[7] 柴洪林, 李红, 彭嘉雄. 基于视觉的夜间无人机导航特征的提取[J]. 计算机工程, 2007, 33(22): 217-219.

[8] 朱宪伟, 李由, 于起峰. 机载视觉自主着陆过程中的跑道提取方法[J]. 国防科技大学学报, 2009, 31(2): 20.

[9] Dong Yinwen, Yuan Bingchen, Wang Hangyu, et al. A Runway Recognition Algorithm Based on Heuristic Line Extraction[C]. 2011 International Conference on Image Analysis and Signal Processing (IASP), 2011: 292-296.

[10] Adrian Kaehler, Dr. Gary Rost Bradski. Learning OpenCV: Computer Vision with the OpenCV Library[R]. Sebastopol, CA: O'Reilly Media, 2008.