

结合像素局部对比度的直线特征检测算法*

孙俊锋^{1,2}, 丁少闻¹, 张小虎¹, 张跃强¹

(1. 国防科技大学, 空天科学学院, 湖南长沙 410073; 2. 中国载人航天工程办公室, 北京 100000)

摘要:鲁棒高效的直线提取算法在空间目标位姿估计、跟踪、骨架重建等领域有广泛的应用。提出基于像素局部对比度和全局虚警抑制的直线段检测算法。算法分为四步:第一步,计算像素点的梯度以及局部对比度;第二步,将方向大致相同的像素点分成直线支撑区域(像素方向为像素梯度方向的顺时针垂直方向),并对支撑区域进行矩形近似;第三步,在直线支撑区域内拟合直线段,计算直线参数;第四步,根据直线支撑区域信息验证直线段,抑制虚警。实验测试结果表明:该算法在复杂条件下,特别是光照影响条件下,具有更好的鲁棒性和效率。

关键词:直线检测;局部对比度;虚警抑制

中图分类号:TP391.4 **文献标志码:**A **文章编号:**1001-2486(2017)06-031-08

Linear feature detection algorithm combined with pixel local contrast

SUN Junfeng^{1,2}, DING Shaowen¹, ZHANG Xiaohu¹, ZHANG Yueqiang¹

(1. College of Aeronautics and Astronautics, National University of Defense Technology, Changsha 410073, China;

2. China Manned Space Engineering Office, Beijing 100000, China)

Abstract: Robust and efficient line extraction algorithm has been widely used in space photogrammetry field such as target pose detection, tracking and skeleton reconstruction. A line segment detector based on local contrast of pixels and global false detection control was proposed. The method consists of four steps. Firstly, the gradient and local contrast of each pixel was calculated. Secondly, the pixels with similar orientations were clustered into the line support region which was approximated by a rectangle. After that the line support region was fitted into a line segment with the above one pixel width and the line parameters were estimated. Finally, the line segment candidates were validated and the false detection was controlled. The experimental results show that the proposed method is much more robust and efficient than the traditional methods in hash conditions, especially in complex illumination.

Key words: line detect; local contrast; false detection control

由于空间人造目标多为由多面体组成的结构性目标,且目标与背景具有不同的反射特性,因而目标上包含了大量的能够很好地反映目标结构信息的直线段特征。利用直线段特征对目标进行表达能够降低空间三维目标跟踪、识别与姿态估计等算法的复杂度,同时算法的鲁棒性也将大大提高。

直线检测方法可以分为两大类:基于边缘图的直线检测和基于相位编组的直线检测。基于边缘图的直线检测主要为经典的 Hough 变换算法^[1]以及改进算法。这类算法首先利用经典边缘检测算子,如 Canny 算子^[2],检测图像边缘,然后将图像空间映射到参数空间,并在参数空间进行投票,最后选取参数空间的极值作为图像直线

输出。传统的 Hough 变换算法需要占用较多系统资源,实时性较差。另外 Hough 变换定位误差同参数空间的离散率密切相关,同时也无法准确定位直线端点。针对传统 Hough 变换存在的问题,一些有价值的改进算法相继被提出,如概率 Hough 变换算法^[3]、随机 Hough 变换算法^[4-5]、模糊 Hough 变换算法^[6]和分层迭代 Hough 变换算法^[7]等,但是由于上述方法只用到了图像的全局统计特性来检测直线,未用到图像点的局部信息,因而容易受噪声和纹理的影响,检测出虚假直线。另外一种比较经典的基于边缘图的直线检测算法为基于迭代拟合的直线检测算法^[8]。这类方法通过迭代拟合直线,将边缘图分割为直线段,但是这类方法效率较低,且容易检测到大量虚假直线。

* 收稿日期:2017-01-24

基金项目:国家自然科学基金资助项目(11272347)

作者简介:孙俊锋(1980—),男,山西朔州人,博士研究生,E-mail:66708558@qq.com;

张小虎(通信作者),男,研究员,博士,博士生导师,E-mail:zhangxiaohu@vip.163.com

基于相位编组的直线检测算法源于 Burns 的工作^[9],将图像中方向相近的像素连接为直线。由于像素的方向易受噪声干扰,因而该方法容易产生大量的虚假直线。针对 Burns 方法存在的问题,目前学者提出了大量的改进方法,且主要集中在抑制虚假直线上。Desolneux 等首次提出了一种无参数直线检测算法^[10-11]。该方法利用 Helmholtz 准则^[10]定义感知上有意义的结构为 ε 的有意义直线,并利用该准则对检测的直线进行验证,剔除虚假直线。但是该方法在抑制虚假直线的同时容易将图像中正确的直线剔除,且算法的效率很低。van Goi 等对 Burns 的工作进行扩展,将直线视为方向一致的区域,并利用 Desolneux 设计的准则,提出了近年公认的综合性能最好的直线检测算法,直线分割检测子 (Line Segment Detector, LSD)^[12],但是该方法对白噪声较敏感,且实时性有待提高。Akinlar 等^[13-14]首先使用边缘检测 (Edge Detection, ED) 算法^[15-16]得到连续的边缘,并对边缘直线进行拟合,最后同样基于 Desolneux 设计的准则剔除虚假直线。由于该方法将 LSD 中的初始直线检测替换为了更为高效的 ED 算法,因此算法的时间效率较高,但是 ED 算法得到的为近似的边缘直线,因而其对直线的定位精度不如 LSD^[17]。

传统的直线检测算法^[4,8]是在 Canny^[2]等边缘检测基础上进行的。但是由于空间环境光照变化剧烈,目标受光线照射的一面图像亮度较大,甚至出现饱和,而未受光线照射的一面图像很暗,因此很难找到统一的参数阈值进行 Canny 边缘检测,以保证目标上关键区域的边缘能够被检测到。当场景光照条件发生变化时,需要重新设置 Canny 参数以保证目标边缘被检测到;当图像尺度发生变化时,为了保证目标边缘段不被滤除,需要对控制最小边缘的参数进行重新设置;当图像受噪声影响较大时,也会检测到大量的错误直线。为了提高直线检测算法的自适应性,同时提高算法在复杂空间光照、运动模糊、尺度变化以及噪声等条件下的鲁棒性,本文提出了基于像素局部对比度和全局虚警抑制的直线段检测算法。

1 算法原理

1.1 直线支撑区域的构建

给定图像,首先计算图像像素梯度场,像素 (x, y) 的梯度根据式(1)进行计算。

$$\begin{cases} g_x(x, y) = \frac{I(x+1, y) + I(x+1, y+1) - I(x, y) - I(x, y+1)}{2} \\ g_y(x, y) = \frac{I(x, y+1) + I(x+1, y+1) - I(x, y) - I(x+1, y)}{2} \end{cases} \quad (1)$$

像素 (x, y) 的方向和梯度模值为:

$$\begin{cases} \theta(x, y) = \arctan \left[\frac{g_x(x, y)}{-g_y(x, y)} \right] \\ g(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \end{cases} \quad (2)$$

使用式(2)计算的实际上为像素 $(x + 0.5, y + 0.5)$ 的梯度,在后续算法步骤中将对该偏移进行处理。

由于强阳光会引起场景图像的高对比度和高动态范围,受太阳照射的一面,图像亮度很大,甚至过饱和,而未受太阳照射的一面,光线很暗,而造成目标体上的弱边缘由于绝对梯度模值太小而无法被检测。然而,尽管目标体上弱边缘的绝对梯度模值很小,但是其在局部的对比度仍然较大,因此定义像素点局部对比度为:

$$s(x, y) = [\max(r_+, r_-)]^{\lambda_1} |g - \bar{g}|^{\lambda_2} \quad (3)$$

其中: r_+, r_- 分别为沿当前像素点法线正向和反向搜索发现的首个比当前像素点梯度模值大的像素点的距离; \bar{g} 为以当前像素为中心半径 $\min(r_+, r_-)$ 的线性区域内的像素梯度模值均值; λ_1, λ_2 分别为同距离和灰度相关的对比度的权重,实验中分别取 0.3 和 0.1。

式(3)实际上衡量的是像素在局部区域的梯度显著性。对于受环境光照影响而梯度模值较小的边缘点,如果其相对于邻近像素的对比度较大(在较大范围内其梯度模值最大),则其对应的局部对比度值也较大。

由于实际八位图像的灰度量范围在 0 ~ 255,且均为正整数,因此 q 取为 2 像素。实验中 τ 取为 22.5° ,可以计算出像素梯度模值阈值 $\rho_g = 5.2$ 。如果像素点梯度模值大于或等于 ρ_g ,则将其标记为候选边缘点;如果像素点梯度模值小于 ρ_g ,但是其局部对比度值大于 ρ_s ,则也将其标记为候选边缘点;如果像素点梯度模值小于 ρ_g ,同时其局部对比度值也小于 ρ_s ,则将其剔除。

初始条件下,支撑区域为单一像素点,将直线支撑区域方向设置为该像素点方向,并对该像素点进行标示;对像素的八连通域进行测试,将同支撑区域方向夹角小于 τ 的像素加入直线支撑区域内,并对该像素进行标示;重新计算直线支撑区域方向,即支撑区域内方向响应的合成方向。迭代进行以上步骤,直到没有新的像素点加入直线支

撑区域为止。最终方向相近的像素点被分在相同的直线支撑区域内。

由于局部对比度值大的像素点更有可能为边缘点,因此起始像素点根据像素点局部对比度值大小进行选取,局部对比度值大的被优先选取。为了避免像素点的重复遍历,对被选取直线支撑区域的边缘点设置标志。对于图像中较显著的边缘直线,起始像素点的选择不会影响直线支撑区域的生成结果;相反,对于图像中的非直线段,直线支撑区域将依赖于起始像素点的选择。

1.2 直线支撑区域的矩形近似

得到直线支撑区域后,对该支撑区域进行矩形近似(包含直线支撑区域的最小矩形),并利用该矩形区域对直线进行表示(直线表示为有宽度的区域),直线的中点为矩形区域的中心,其计算方法为:

$$\begin{cases} c_x = \frac{\sum_{p_i \in \text{LSR}} |\nabla p_i| x_i}{\sum_{p_i \in \text{LSR}} |\nabla p_i|} \\ c_y = \frac{\sum_{p_i \in \text{LSR}} |\nabla p_i| y_i}{\sum_{p_i \in \text{LSR}} |\nabla p_i|} \end{cases} \quad (4)$$

其中, $|\nabla p_i|$ 为像素点 $p_i = (x_i, y_i)$ 的梯度模值, p_i 为直线支撑区域(Line Support Region, LSR)内的边缘像素点。

直线支撑区域可以表示为矩形区域,直线的方向取为矩形区域的主轴方向,其为式(5)中矩阵最小特征值对应的特征向量方向。

$$\mathbf{M} = \begin{pmatrix} m^{xx} & m^{xy} \\ m^{xy} & m^{yy} \end{pmatrix} \quad (5)$$

其中,

$$\begin{cases} m^{xx} = \frac{\sum_{p_i \in \text{LSR}} |\nabla p_i| (x_i - c_x)^2}{\sum_{p_i \in \text{LSR}} |\nabla p_i|} \\ m^{xy} = \frac{\sum_{p_i \in \text{LSR}} |\nabla p_i| (x_i - c_x)(y_i - c_y)}{\sum_{p_i \in \text{LSR}} |\nabla p_i|} \\ m^{yy} = \frac{\sum_{p_i \in \text{LSR}} |\nabla p_i| (y_i - c_y)^2}{\sum_{p_i \in \text{LSR}} |\nabla p_i|} \end{cases} \quad (6)$$

由于梯度大的像素点更有可能对应图像边缘像素点,因此在计算直线中点和方向时赋予其更大的权重。直线的长度和宽度对应于矩形区域的长和宽。赋予直线宽度在一定程度上可以提高直线检测算法对于图像运动模糊以及离焦模糊的鲁

棒性。

1.3 直线段验证

1.3.1 Helmholtz 原则

传统的直线检测算法是基于 Gestalt 理论的。当图像上出现大量噪声时,传统的方法也将检测到大量的错误直线;当图像受运动模糊或者离焦模糊影响时,传统的算法也将检测到大量的虚假平行直线。

直线段验证阶段的主要任务是对初始直线段进行检验,筛查错误直线段,保证每次检测的错误直线段的数量最少。根据 Helmholtz 原则^[10],对于噪声图像,将不进行感知或者检测,保证错误感知最少。另外 Helmholtz 原则认为,如果图像中的构型(共线的边缘点)出现的次数的期望很小,那么它就对应有意义的构型,即要感知或者检测的特征。

1.3.2 ε 有意义直线

假设图像模型为一随机场,图像上每一个像素点 m 的梯度方向为服从 $[0, 2\pi]$ 上均匀分布的随机变量,且随机变量相互独立。在直线支撑区域构建步骤,将同直线支撑区域方向夹角小于 τ 的像素点认为是同支撑区域一致的像素点。由于像素点同直线的夹角范围为 $[0, \pi]$,因此像素点位于直线构型内的概率 $p = \tau/\pi$ 。

定义 ε 有意义直线支撑区域为至少包含 $k(r)$ 个一致像素点的直线支撑区域。

$$k(r) = \min \left\{ k \in \mathbb{N}, P[S_r \geq k] \leq \frac{\varepsilon}{(MN)^{5/2}} \right\} \quad (7)$$

其中, S_r 对应区域 r 内同直线支撑区域方向一致的像素点个数(方向夹角小于 p)。

定义错误警报数(Number of False Alarm, NFA),即直线支撑区域出现次数的期望为:

$$\begin{aligned} NFA(r, k) &= (MN)^{5/2} P[S_r \geq k] \\ &= (MN)^{5/2} \sum_{n=k}^l \binom{l}{n} p^n (1-p)^{l-n} \end{aligned} \quad (8)$$

其中, l 为直线支撑区域内像素点的总个数, n 为区域内每个非一致像素点。

如果直线支撑区域 r 对应的 NFA 小于 ε ,则认为 r 是 ε 有意义的直线支撑区域。

1.3.3 最小直线区域大小

根据式(8),可以求得 ε 有意义直线支撑区域的最小尺寸为:

$$\begin{cases} NFA(r, l) = (MN)^{5/2} p^l \leq \varepsilon \\ l \geq \frac{\ln(\varepsilon) - \frac{5}{2} \ln(MN)}{\ln(p)} \end{cases} \quad (9)$$

如果取 $\varepsilon = 1$, 则式(9)可以简化为:

$$l_{\min} = \frac{-2.5 \ln(MN)}{\ln(p)} \quad (10)$$

式(10)定义了 ε 有意义的最小直线支撑区域的尺寸, 如果取图像尺寸为 $M = N = 640$, 像素点为八连通, 即 $p = 0.125$, 则 $l_{\min} = 16$ 。在进行直线方向和中心拟合前, 可以根据式(10)将尺寸小于 l_{\min} 的直线支撑区域剔除。

1.3.4 联合验证

式(8)定义的 NFA 实际上是衡量一条直线段是否有意义。这种做法容易忽略图像中较短的直线段, 同时也忽略了直线段同周围邻近直线段的关系, 即它们可能处于同一目标边缘上。为此, 对于未能通过单一直线验证的直线段, 可以联合其邻近共边缘的直线段进行直线验证。

$$\begin{aligned} NFA(E, k_E) &= N_E P[S_E \geq k_E] \\ &= N_E \sum_{n=k_E}^{l_E} \binom{l_E}{n} p^n (1-p)^{l_E-n} \end{aligned} \quad (11)$$

其中, E 为包含当前直线段的共边缘直线段的集合, k_E 为 E 内一致性边缘点的个数, l_E 为边缘点的总个数, N_E 为测试的总个数。

$$N_E = (MN)^{(L+1)+0.5} \quad (12)$$

式中, L 为 E 中直线段集合包含的直线段个数。

为了保证图像中的虚假检测直线段总数目小于 ε , ε_E 取为 ε/L 。如果候选直线段通过联合验证, 则将其作为最终检测直线段输出。实验中只需联合候选直线段周围最多两条直线段进行验证。

2 算法流程

Step 1: 算法初始化, 包括图像 I , 参数 $\rho_g, \rho_s, \tau, \varepsilon$ 和 ρ_d ;

Step 2: 计算每个像素点的梯度模值、方向以及显著性值;

Step 3: 按照显著度值对像素点进行排序, 组成序列, 并设置所有像素点的标志为 Not Used, 将模值小于 ρ_g 且显著度值小于 ρ_s 的像素设置为 Used;

Step 4: 从序列中选取一个标志为 Not Used 且显著性值最大的像素点 p , 进行如下步骤:

1) 以 p 为起始点, 构建直线支撑区域, 将同直线支撑区域间方向夹角小于 τ , 标志为 Not Used 的像素点加入支撑区域内, 并将其标志设置为 Used, 更新支撑区域方向, 不断进行该步骤, 直到没有新的像素点加入为止;

2) 计算支撑区域的近似矩形, 并计算支撑区域内一致性像素点比率;

① 如果矩形内一致像素点比率小于等于 ρ_d , 且矩形尺寸小于 l_{\min} , 则跳回步骤 4 中的 1);

② 如果矩形内一致像素点比率大于 ρ_d , 则转入步骤 4 中的 3);

③ 如果矩形内一致像素点比率小于 ρ_d , 且矩形尺寸大于 l_{\min} , 修剪矩形区域, 若修剪结果满足要求则转入步骤 4 中的 3), 否则跳回步骤 4 中的 1);

3) 计算直线参数;

4) 计算矩形区域的 $NFA(r)$, 如果 $NFA(r) \leq \varepsilon$, 将其作为结果输出;

Step 5: 对于直线序列中未通过单一直线验证的候选直线, 结合其同邻域内直线段(已验证通过)进行联合验证, 如果验证通过, 则将其加入检测直线序列;

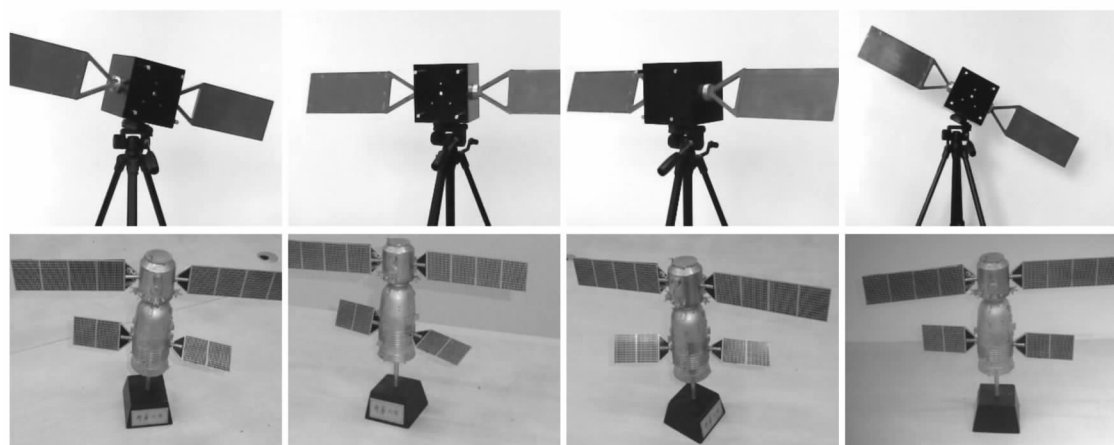
Step 6: 算法输出直线序列。

3 实验测试

为了测试算法的性能, 设计了两组实验。第一组: 利用各种类型的图像, 包括受各种光照、噪声、模糊(包括高斯模糊、运动模糊等)等影响的图像, 对基于直线支撑区域的算法进行参数自适应性测试, 测试过程中不进行参数重新设置; 第二组: 算法性能对比试验, 为了进一步反映算法性能, 利用测试图像集, 将本节直线检测算法同基于 Canny 边缘检测^[2]的方法(在边缘检测的基础上进行直线段拟合的方法, 后文称基于 Canny 方法)进行对比, 同样为了对比的公平性, 测试过程中不改变算法参数设置, 对比指标包括直线总条数、检测时间等。实验计算机配置为: CPU i3 (3.19 G), 内存 4 G。

3.1 算法性能测试实验

为了测试算法的鲁棒性和参数的自适应性, 对图 1 中第一组图像进行光照变换, 如图 2 所示; 对图 1 中第二组图像加零均值高斯噪声, 协方差分别为 10, 20, 30, 如图 3 所示; 对图 1 中第三组图像进行高斯模糊, 核大小分别为 2, 4, 6, 如图 4 所示; 对图 1 中第四组图像进行运动模糊处理, 旋转角度为 5° , 像素平移分别为 10, 20, 30, 如图 5 所示。利用本文提出的算法, 分别对以上变换后的测试图像集进行直线检测。从检测结果来看, 基于直线支撑区域的算法对光照、噪声、模糊等影响具有较好的适应性, 均能实现图像中目标边缘直线的检测。



(a) 第一组
(a) The first group

(b) 第二组
(b) The second groups

(c) 第三组
(c) The third groups

(d) 第四组
(d) The fourth groups

图1 测试图像集
Fig.1 Test image set

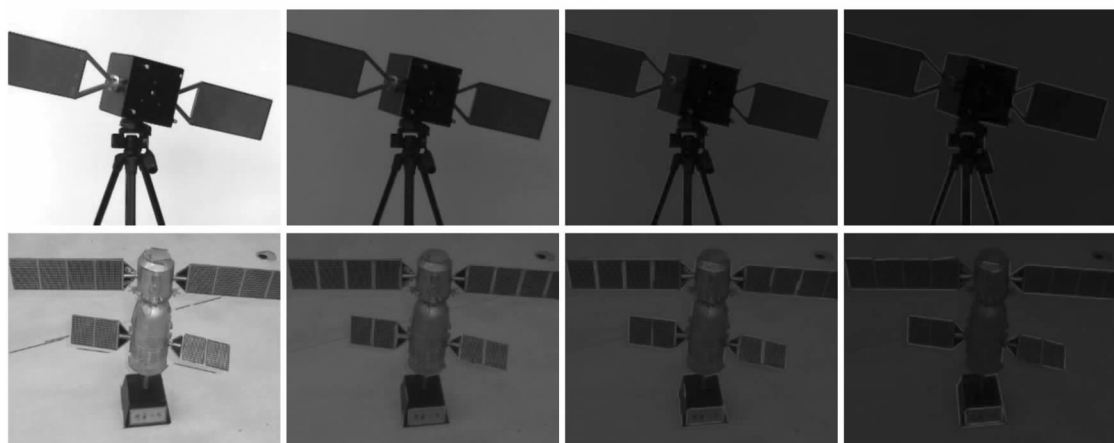
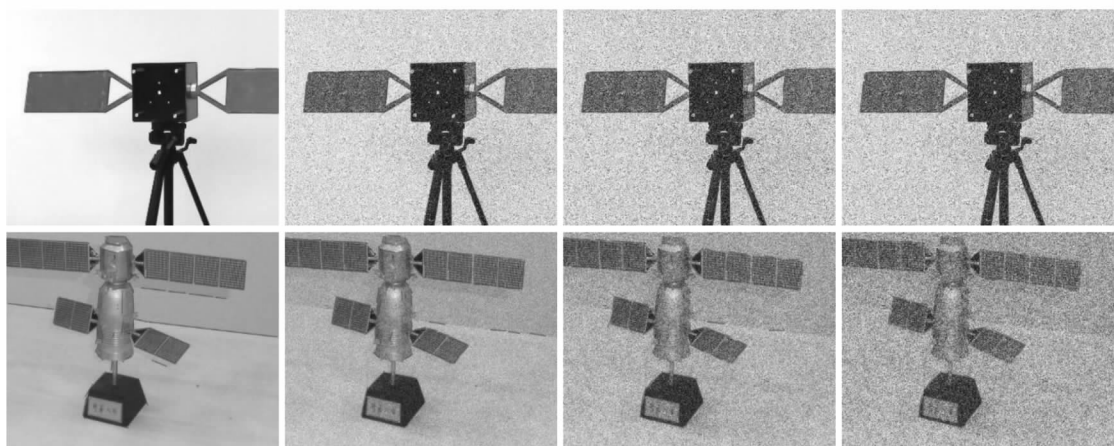


图2 光照变化图直线检测结果
Fig.2 Line detection results of illumination variation figure



(a) 原图
(a) Original

(b) 噪声协方差 10
(b) Noise covariance 10

(c) 噪声协方差 20
(c) Noise covariance 20

(d) 噪声协方差 30
(d) Noise covariance 30

图3 噪声图直线检测结果
Fig.3 Line detection results of noise figure

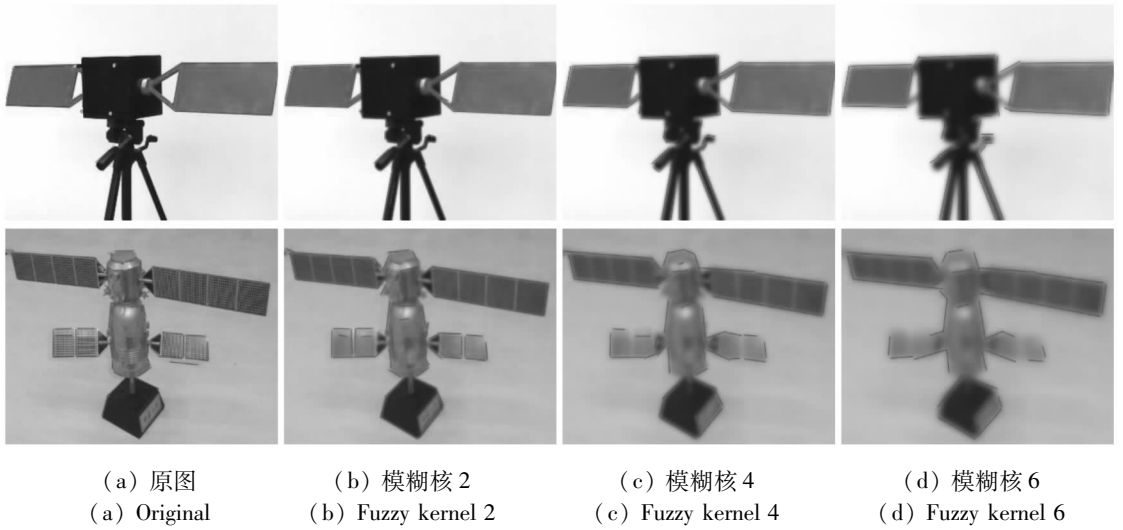


图 4 高斯模糊图直线检测结果
Fig. 4 Line detection results of Gauss fuzzy figure

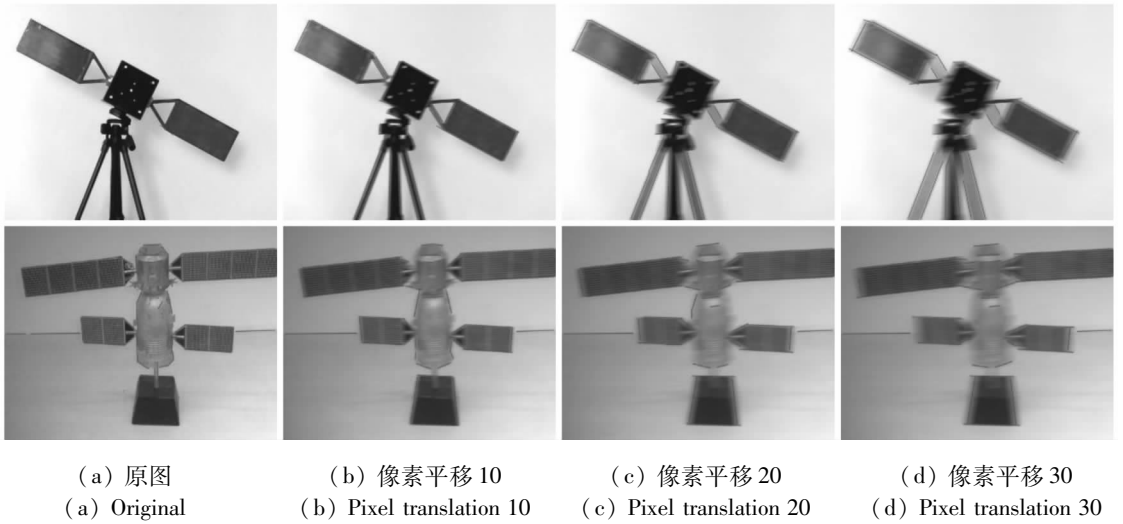


图 5 运动模糊图直线检测结果
Fig. 5 Line detection results of motion blurred figure

3.2 算法性能比较实验

利用 3.1 节中受光照、噪声、模糊等影响的测试图像集,将本节直线检测算法同基于 Canny 边缘检测的算法进行性能比较。比较过程中,两种算法参数设置都保持不变,同时记录下两种算法检测的直线总条数以及算法总时间,以对算法效果和效率进行对比。基于 Canny 的直线检测算法,Canny 边缘检测参数设置如下:高阈值为 100,低阈值为 60,算法运行过程中不进行参数调整。两种算法检测直线总数和检测时间见表 1。

在实验中发现,当图像光照条件发生变化时,基于 Canny 的直线检测算法受影响较大,特别当图像光照条件变化较大时,Canny 算法几乎检

测不到直线,而本文方法所受的影响相对较小;当图像受噪声影响时,基于 Canny 的直线检测算法检测到大量的错误直线,而本文方法所受的影响相对较小,尽管检测的直线总数目有所减少,但是几乎不会检测到错误直线;对图像进行高斯模糊处理时,两种算法检测的直线数目都有所减少,基于 Canny 的直线检测算法受影响更大;当图像发生运动模糊时,基于 Canny 的直线检测算法容易检测到大量虚假直线(平行线段),同时直线的连续性以及定位精度也受到很大影响,而本文方法受影响相对较小。由表 1 可以看出,本文方法直线检测效率在总体上优于基于 Canny 的直线检测算法,同时本文算法的检测时间跟图像大小呈线性关系;而基于 Canny 的直线检测算法跟图

像纹理以及噪声有关系,当图像受噪声影响较大时,算法的检测时间显著增加。

表1 两种算法性能比较

Tab. 1 Performance comparison of two algorithms

测试图	本文方法直线检测结果(直线数目/时间(s))				Canny方法直线检测结果(直线数目/时间(s))			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
1	55/0.063	41/0.063	39/0.047	36/0.047	47/0.078	42/0.078	41/0.078	4/0.093
	105/0.047	75/0.031	58/0.031	38/0.047	86/0.108	61/0.063	35/0.047	10/0.032
2	63/0.062	91/0.094	91/0.109	81/0.109	44/0.125	51/0.718	96/2.109	126/2.859
	85/0.046	78/0.047	53/0.063	28/0.062	78/0.109	60/0.187	64/1.172	87/2.015
3	58/0.062	52/0.063	41/0.078	37/0.094	52/0.094	48/0.078	63/0.172	74/0.188
	90/0.063	77/0.046	44/0.047	36/0.031	80/0.141	46/0.063	14/0.047	0/0.062
4	48/0.078	38/0.063	54/0.078	49/0.094	48/0.125	86/0.187	108/0.218	66/0.187
	82/0.031	51/0.032	47/0.031	50/0.047	71/0.109	37/0.078	30/0.062	29/0.047

由于 Canny 算子需要人工设置高低阈值,实际使用中如果环境光照条件发生变化,目标边缘梯度值也将发生变化,从而造成边缘检测的不完整,进而造成直线检测的不完整。而本文方法未直接进行提取阈值设置,同时结合像素局部对比度值对其进行判别,因而对光照变化拥有更好的鲁棒性。同时本文方法在直线检测的基础上引入了虚假直线抑制,因而算法对噪声拥有更好的鲁棒性。另外本文方法将直线看作矩形区域,并在矩形区域内拟合直线参数,因而算法对高斯模糊、运动模糊等的鲁棒性更好。

LSD 直线提取方法是近些年广泛使用的一种直线检测方法。相对于 LSD 算法,本文方法引入了局部对比度,因而算法受局部光照变化的影响更小;同时本文方法引入了联合验证,结合直线段的邻域信息(组成共线构型)对直线进行验证,因而算法检测的直线段连续性更好。为了更好反映本文方法的性能,将本文算法同 LSD 算法进行对比,结果如图 6 所示。

图 6 给出了 3 幅图像的处理结果,可以更为直观地看到本文方法对直线检测算法性能的改变。



(a) 原始图像
(a) Original image



(b) LSD 直线检测结果
(b) LSD line detection results



(c) 本文方法直线检测结果
(c) Line detection result of the paper's method

图 6 本算法对 LSD 直线检测算法性能对比示例
Fig. 6 Performance comparison between the proposed algorithm and the LSD linear detection algorithm

善,当光照环境变暗时,LSD 算法已不能很好地对图像中的直线进行检测。而相比 LSD 算法,本文方法能检测到更多的边缘直线,即更好地消除光照带来的不利影响。

4 结论

本文提出了基于直线局部支撑区域和全局虚警抑制的直线检测算法。算法首先构建像素支撑

区域并计算其局部显著度;其次将方向大致相同的像素点分成直线支撑区域,并将其修剪拟合为矩形区域;然后对矩形区域内的像素点进行直线拟合得到初始直线;最后根据赫尔姆霍茨准则,剔除由于噪声,光照等的影响产生的虚假直线。实验结果表明,本文算法对复杂光照、噪声、模糊等条件具有更强的鲁棒性。

参考文献 (References)

- [1] Ballard D H. Generalizing the Hough transform to detect arbitrary shapes [J]. *Pattern Recognition*, 1981, 13(2): 111–122.
- [2] Canny J. A computational approach to edge detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, 8(6): 679–698.
- [3] Bergen J R, Shvaytser H. A probabilistic algorithm for computing Hough transforms [J]. *Journal of Algorithms*, 1991, 12(4): 639–656.
- [4] Xu L, Oja E. Randomized Hough Transform (RHT): basic mechanisms, algorithms, and computational complexities [J]. *CVGIP: Image Understanding*, 1993, 57(2): 131–154.
- [5] Vassilios C, Ioannis P. Fuzzy cell Hough transform for curve detection [J]. *Pattern Recognition*, 1997, 30(12): 2031–2042.
- [6] Han J H, Kóczy L, Poston T. Fuzzy Hough transform [J]. *Pattern Recognition Letters*, 1994, 15(7): 649–658.
- [7] Tuytela T, Proesmans M, van Gool L. The Cascaded Hough transform [C]//*Proceedings of IEEE International Conference on Image Processing (ICIP)*, 1997: 736–739.
- [8] Lowe D G. Three-dimensional object recognition from single two-dimensional images [J]. *Artificial Intelligence*, 1987, 31(3): 355–395.
- [9] Burns J B, Hanson A R, Riseman E M. Extracting straight lines [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, 8(4): 425–455.
- [10] Desolneux A, Moisan L, Morel J M. From gestalt theory to image analysis: a probabilistic approach [M]. USA: Springer-Verlag New York, 2008.
- [11] Desolneux A, Moisan L, Morel J M. Meaningful alignments [J]. *International Journal of Computer Vision*, 2000, 40(1): 7–23.
- [12] von Gioi R G, Jakubowicz J, Morel J M, et al. LSD: a fast line segment detector with a false detection control [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, 32(4): 722–732.
- [13] Akinlar C, Topal C. Edlines: real-time line segment detection by Edge Drawing (ED) [C]//*Proceedings of IEEE International Conference on Image Processing*, 2011: 2837–2840.
- [14] Akinlar C, Topal C. EDLines: a real-time line segment detector with a false detection control [J]. *Pattern Recognition Letters*, 2011, 32(13): 1633–1642.
- [15] Topal C, Ozsen O, Akinlar C. Real-time edge segment detection with Edge Drawing algorithm [C]//*Proceedings of IEEE International Symposium on Image and Signal Processing and Analysis*, 2011: 313–318.
- [16] Ye J, Fu G, Poudel U P. High accuracy edge detection with blurred edge model [J]. *Image & Vision Computing*, 2005, 3(5): 453–467.
- [17] 傅叶涛, 杜昌平, 赵耀. 基于主方向的直线检测算法 [J]. *计算机应用*, 2015(S2): 220–222, 249.
FU Yetao, DU Changping, ZHAO Yao. Line detection based on principal direction [J]. *Journal of Computer Applications*, 2015(S2): 220–222, 249. (in Chinese)