

# 一种基于数学形态学与改进的 SUSAN 算子边缘提取快速算法

肖玲玲, 赵秀鸟

(江西理工大学, 江西 赣州 341000)

**摘要:** 提出基于数学形态学和 SUSAN 算子的灰度图像边缘提取方法, 同时对 SUSAN 算子进行了改进。该方法利用数学形态学开运算估计背景, 将原始图像与背景进行几何运算, 在处理后的图像上运用改进的 SUSAN 算子提取边缘, 并进行了仿真实验。实验结果表明, 该方法不仅具有较好的去噪和边缘提取能力, 而且算法简单易于实现, 运算速度快。

**关键词:** 数学形态学; 边缘检测; SUSAN 算子; 图像处理

中图分类号: TP391

文献标识码: B

Draw the fast algorithm on a kind of edge based on mathematics morphology and improved SUSAN operator

XIAO Ling Ling, ZHAO Xiu Niao

(Institute of Technology of Jiangxi, Ganzhou 341000, China)

**Abstract:** Edge-detecting algorithm is based on mathematical morphology and gray SUSAN operator. And SUSAN operator is improved. This method is the use of mathematical morphology open computing the estimated background, the original image and the background of geometric computing in image processing to improve the use of the SUSAN edge extraction operator. And a simulation experiment was carried out. The experimental results show that the method not only has a better ability to de-noising and edge detection, but also algorithm is simple and easy to implement, faster computing.

**Key words:** mathematical morphology; edge detection; SUSAN operator; image processing

在计算机视觉和图像处理系统中, 由于大量信息被包含在图像的边缘中, 因而边缘提取是一种非常重要的图像预处理方法。边缘提取的效果直接决定了后续处理的质量, 边缘提取是图像识别过程中非常重要的环节。数学形态学作为一门新兴的图像处理和图像分析学科, 最初只是分析几何形状和结构的数学方法, 建立在集合代数基础上, 用集合论方法定量描述几何结构的科学<sup>[1]</sup>。数学形态学是一种非线性滤波方法, 可以用来解决抑制噪声、特征提取、边缘检测、图像分割、形状识别、纹理分析、图像恢复与重建、图像压缩等图像处理问题<sup>[2]</sup>。SUSAN (Smallest Univalue Segment Assimilating Nucleus) 算法是一种基于图像灰度相似性比较的边缘检测算法, 不需梯度计算, 具有算法简单、定位准确、抗噪声能力强等特点。本文在介绍基于数学形态和 SUSAN 算法思想的

基础上, 基于两者对图像处理的思想, 提出了一种两者相互结合运用的新算法。实验证明, 该新算法在提高边缘检测效果的基础上, 大大减少了算法的计算量, 提高了运算速度。

## 1 基于数学形态学的图像边缘提取思想

数学形态学滤波是基于信号的几何特征, 利用预先定义的结构元素对图像进行运算, 达到既滤除噪声, 又能较好地保持图像轮廓信息。其基本运算包括: (1) 膨胀: 可以填充图像中的小孔 (相对于结构元素而言的孔洞) 及在图像边缘出现的小的凹陷部分, 对图像外部有滤波的作用; (2) 腐蚀: 可以消除图像中小的成分, 对图像内部有滤波的作用; (3) 开启: 先腐蚀后膨胀的过程, 具有消除细小物体、在纤细处分离物体和平滑较大物体边界的作用; (4) 闭合: 先膨胀后腐蚀的过程, 具有填充物体内部细小

## 技术与方法 Technique and Method

孔洞,连接临近物体和平滑边界的作用。因此,将数学形态学直接用于边缘检测存在结构元素单一的问题,对结构元素同方向的边缘敏感,而与其方向不同的边缘或噪声会被平滑掉<sup>[3-5]</sup>。

### 2 SUSAN 边缘检测算法

#### 2.1 边缘检测原理

如图 1 所示,用 1 个圆形模板在图像上移动,若模板内像素的灰度与模板中心像素(核 Nucleus)灰度的差值小于一定门限值时,则认为该点与核具有相同(或相近)的灰度,由满足这样条件的像素组成的区域称为 USAN (Univalue Segment Assimilating Nucleus)。可以看出,当圆形模板完全处在背景或目标中时,USAN 区域面积最大,如图 1 中的 a 所示;当模板移向目标边缘时,USAN 区域逐渐变小,如图 1 中的 b 所示;当模板中心处于边缘时,USAN 区域很小,如图 1 中的 c 所示;当模板中心处于角点时,USAN 区域最小如图 1 中的 d 所示。因此,可以通过计算每 1 个像素的 USAN 值,并与设定的门限值进行比较,如果该像素的 USAN 值小于门限值,则该点可以认为是 1 个边缘点。这就是 SUSAN 算法思想。

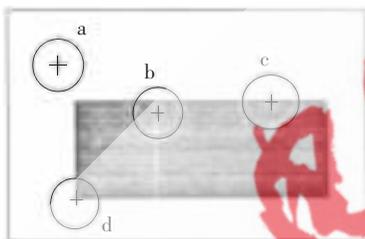


图 1 SUSAN 边缘检测原理

用圆形模板扫描整幅图像,比较模板内每 1 个像素与中心像素的灰度值,通过与给定的门限值  $t$  比较,来判别该像素是否属于 USAN 区域,计算公式是:

$$c(r, r_0) = \begin{cases} 1 & \text{if } |I(r) - I(r_0)| \leq t \\ 0 & \text{if } |I(r) - I(r_0)| > t \end{cases} \quad (1)$$

式中  $c(r, r_0)$  为模板内是否属于 USAN 区域的像素的判别函数,  $I(r_0)$  是模板中心像素(核)的灰度值,  $I(r)$  为模板内其他任意像素的灰度值,  $t$  是灰度差门限。图像中每一点的 USAN 区域大小可用下式计算:

$$n(r_0) = \sum_{r \in D(r_0)} c(r, r_0) \quad (2)$$

$D(r_0)$  为以  $r_0$  为中心的圆形模板区域。

得到每个像素的 USAN 值  $n(r_0)$  之后,再与预先设定的几何门限  $g$  进行比较。当  $n(r_0) < g$  时,所检测到像素位置  $r_0$  可以认为是 1 个边缘点。由 SUSAN 边缘检测原理可知,对于边缘点的 USAN 值应该小于或等于整个圆形模板内所包含像素个数  $n_{\max}$  的一半,因此理论上  $g$  可取  $0.5n_{\max}$ 。

《微型机与应用》2009 年第 23 期

#### 2.1.1 模板的选取

SUSAN 原理采用圆形的模板,目的是使检测达到各向同性。而在实际运用中,由于图像的数字化无法实现真正的圆形模板,因此采用近似圆代替。一般取  $3 \times 3$ ,  $5 \times 5$  或 37 像素的模板,如图 2 所示。

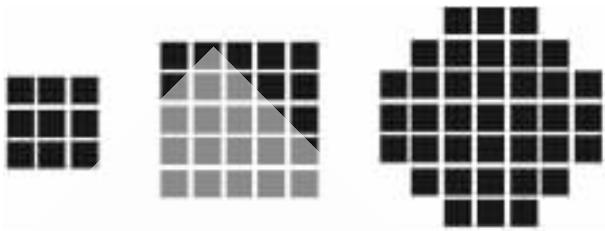


图 2 SUSAN 模板

本文采用 37 像素模板,检测达到各向同性,目的在于弥补数学形态学对结构元素同方向的边缘敏感,而与其方向不同的边缘或噪声将被平滑掉。

#### 2.1.2 门限 $t, g$ 的确定

门限  $t$  表示所能检测边缘点的最小对比度,也是能忽略噪声的最大容限。 $t$  越小,从对比度越低的图像中提取边缘特征越清晰。因此对于不同对比度和噪声情况的图像,应取不同的  $t$  值。门限  $g$  决定了边缘点的 USAN 区域的最大值,即只要图像中的像素 USAN 值小于  $g$ ,该点就被判定为边缘点。当  $g$  过大时,边缘点附近的像素可能会当作边缘被提取出来,过小则会漏检部分边缘点。考虑到模板不是圆形,  $g$  值可取略大于  $0.5n_{\max}$ ,实践证明,  $g$  值取  $0.75n_{\max}$  时,能够较好地检测到边缘点。在 SUSAN 算法中,可以根据不同的情况,即根据图像边缘的形状,以及目标和背景的灰度对比度选择合适的门限  $t, g$ 。经验证,在本文程序设计  $t=10, g=26$  时,就能达到较好效果。

#### 2.2 SUSAN 边缘检测改进算法

本文提出在运用 SUSAN 算法进行边缘检测前,首先以每 1 个像素点为中心,在 2 个垂直的方向计算线段两端像素点的灰度差,并与程序设置的差值门限  $Th$  进行比较,大于  $Th$  的像素点作为候选边缘点,小于  $Th$  的点认为是内部像素点则被剔除,得到的候选边缘点为边缘及其邻域像素,然后只需对候选边缘点利用 SUSAN 边缘检测算法进一步定位边缘。

对于一幅数字图像,一般区域内部像素所占比例比边缘像素大得多,这样就可以在第 1 步剔除掉大部分的内部像素。考虑到边缘模糊的现象,线段选取不宜太短。 $Th$  的选择可以根据具体图像的对比度选取,取值不宜过大,否则会漏检边缘。经验证,在程序设计  $Th=4$  时,能达到较好效果。

如果对图像采用 SUSAN 算法进行边缘检测,以 37 像素的近似圆作为模板,算法加速前,每 1 个像素点首先要与中心像素进行灰度比较,需要做 36 次减法运算。

欢迎网上投稿 www.pcachina.com 85

## 技术与方法 Technique and Method

然后计算USAN值,对区域内部像素也要做36次加法运算,最后与设定的几何门限 $g$ 进行比较再做1次减法运算。共计36次加法运算、37次减法运算。算法加速后,对于区域内部像素,首先计算垂直方向的2个灰度差,需要2次减法运算,然后同 $Th$ 进行比较,再作1次减法运算,即可剔除这一像素,所以只需做3次减法运算。虽然对于候选边缘点,多了3次减法运算,但是由于区域内部像素比例很大,所以总体的计算量大大减少。

### 3 数学形态学与 SUSAN 算子的边缘提取方法

#### (1)背景估计

首先对原始图像采用形态学开运算估计背景,得到背景灰度图像,此时图像中的孤立点或毛刺都被去掉了,达到了噪声滤除的目的。但同时也带来了一定程度上的模糊,所以不能以此图像作为最终的处理图像。

#### (2)灰度变换

在原始图像基础上加上第1步得到的背景灰度图像乘以1个比例系数 $\sigma$ (此系数根据图像背景相对于目标的亮暗程度进行调节),得到一个新的图像。目的是减弱噪声影响的同时又不使目标轮廓模糊,缺点是整个图像变得太亮。

#### (3)生成待检测图像

由于整个图像变亮,使得目标的细微边缘变得不是很清晰,因此在第2步得到的图像基础上减去一个灰度值 $t$ (此灰度值视图像目标的清晰程度进行调节),得到最终待边缘检测的图像。

#### (4)提取边缘

运用SUSAN边缘检测改进算法对边缘进行提取。

### 4 算法验证

从图3(a)中可以看出图像背景含有较多的复杂纹理和噪声,图3(b)为原始图像经过中值滤波后运用SUSAN算法检测得到的边缘图像,目标的轮廓很模糊。其原因是目标与噪声混在一起,图像的背景噪声给检测结果带来了很大的影响。

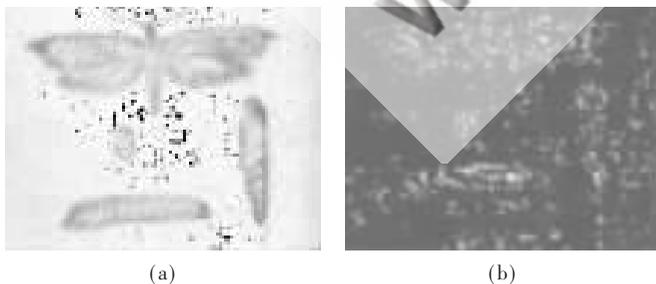


图3 枣豹蠹蛾图像和中值滤波后SUSAN检测效果

从图4(a)中可以看出图像的背景噪声基本上被消

除,虽然带来了图像边缘的轻微模糊,但从提取的效果可以看出目标的主要特征得以保留,且较之图4(b)有了很大改善。

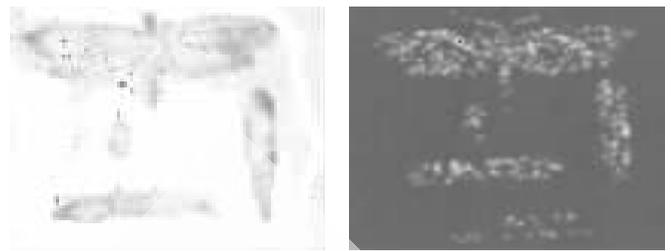


图4 经过处理的枣豹蠹蛾图像和SUSAN检测效果

以上实验证明,采用本文提出的运用数学形态学与SUSAN算子相结合的边缘提取方法,不但对具复杂纹理和噪声的图像有着显著的改善,而且对于光照不均的图像边缘提取效果也较好。

本文在大量实验的基础上,提出了基于数学形态学和改进的SUSAN算子的边缘提取方法。利用形态学开运算估计背景,并与原始图像全局灰度进行几何运算来减弱背景的复杂纹理和噪声对图像的影响,以此来改善目标边缘提取的效果。并结合SUSAN基于图像灰度相似性比较的边缘检测算法,不需梯度的计算,具有算法简单、定位准确、抗噪声能力强等特点,提高了算法效率。实现了两者的优势互补,提高图像边缘检测的速度。通过实验进行验证,其边缘提取效果明显优于利用传统的中值滤波后边缘提取得到的结果。从图像边缘提取的效果来看,图像的检测性能较好,得到的边缘比较清晰,图像的细节表现力和抗噪能力得到加强,而且算法的速度也有了一定的提高。为后续的特征提取、目标识别打下了良好的基础。

#### 参考文献

- [1] 吴丹,刘修国,尚建嘎.数学形态学在图像处理与分析中的应用及展望[J].工程图像学报,2003(2):120-125.
- [2] 戴青云,余英林.数学形态学在图像处理中的应用进展[J].控制理论与应用,2001,18(4):478-482.
- [3] 杨翔宇,孙惠.用数学形态学进行图像边缘检测的新方法[J].开发研究与设计技术,2007(2):501-502.
- [4] ZHAO Xiao Dong. Mathematical morphological binary image real-time parallel processing using optical frequency filtering of complex-valued kernel [J]. Chinese J. Lasers, 1998, A25(11): 1031-1034.
- [5] 朱秀昌,刘峰,胡栋.数字图像处理与图像通信[M].北京:北京邮电大学出版社,2002.

(收稿日期:2009-07-03)