

用带形状校正的腐蚀膨胀实现 Euclidean 距离变换

陆宗骥 朱煜

(华东理工大学信息学院, 上海 200237)

摘要 提出一种用带形状校正的腐蚀膨胀实现的 Euclidean 距离变换新算法。该方法的特点是采用新的数据结构——线段表来表示区域与边界。对于用线段表表示的区域作腐蚀膨胀比用卷积型腐蚀膨胀算法效率提高数十倍。通过总结腐蚀膨胀造成失真的规律, 设计出形状校正的方法来消除所造成的误差。与传统基于局部距离累加的 Chanfer 算法相比较, 该方法在保真度与处理效率两方面都有提高。新的距离变换算法也可用于数字图像的合成, 优点是生成羽化蒙板时形状保真度高并且运行速度快。特别适用于任意形状区域可选宽度边界条带上的羽化处理。

关键词 Euclidean 距离变换 腐蚀 膨胀 线段表 线处理 图像合成

中国法分类号: TP391.41 文献标志码: A 文章编号: 1006-8961(2010)02-0294-07

Implementation of Euclidean Distance Transform Using Erosion and Dilation with Form Correction

LU Zong-qij ZHU Yu

(The College of Information, East China University of Science and Technology, Shanghai 200237)

Abstract A novel method of implementation for Euclidean distance transform using erosion and dilation with form correction is proposed in this paper. The characteristic of the method is a new data structure, which uses line segment table to represent region and edge. Compared with traditional convolution mask, the efficiency of erosion and dilation operation for region expressed by line segment table is improved more than 10 times. Through analyzing the rules of distortions caused by erosion and dilation, we propose a form correction method to decrease the errors. Compared with the Chanfer algorithm, our new approach is better for fidelity and efficiency. The new distance transform can also be used for image blend thanks to the advantage of fidelity and high speed when creating feather mask. Furthermore, it is especially propitious to feather operation of selected boundary in arbitrarily shape regions. The principle and implementation are discussed in detail. And the efficiencies of different processing methods are compared.

Keywords euclidean distance transform (EDT), erosion, dilation, line segment table (LST), line processing (LP), image blend

0 引言

在二值图像的处理中, 距离变换是一种极为重要的处理方法, 精确计算距离的距离变换称为 Eu-

clidean 距离变换 (EDT)。距离变换可以用于多尺度腐蚀、膨胀以及测量距离, 例如区域的平滑、骨架化、颗粒尺寸分布、形状描述、对象的确定与识别、图像分割、路径搜索^[1], 以及图像合成中羽化蒙板的生成^[2]等。

收稿日期: 2008-08-01; 改回日期: 2008-10-27

第一作者简介: 陆宗骥 (1945—), 男, 教授。主要研究领域为图像处理与模式识别, 研究兴趣为图像分割技术。编著《C/C++ 图像处理编程》和合作编著《Visual C++ .NET 图像处理编程》。E-mail: zongqili@163.com

通常将距离变换算法分为 4 种类型^[1]:

- 1) 基于局部距离累加的 Chamfer 法^[1,3-4]。
- 2) 采用复杂数据结构进行操作的方法。
- 3) 卷积法,即采用圆形结构元素的数学形态学腐蚀、膨胀算法^[5]。
- 4) 各向同性的热传导方程数值解。

这些方法中,2)、4)类方法能得到精确的距离值,但它们不是方法复杂就是运行效率不高。因此,平时使用最为广泛的是其中的第 1 类方法。

这 4 类算法有一个共同的特点,就是它们并不直接计算点与点间的距离,因此都是计算二值图像距离图的间接方法。其中,Chamfer 法虽然简单实用,但也存在形状失真和必须整幅图像同时进行处理缺点。

文献 [6]、[7] 采用边界跟踪的方法实现了类似腐蚀膨胀的过程,在此过程中计算像素的距离值,以此提高距离变换的速度与精度取得一定成效。

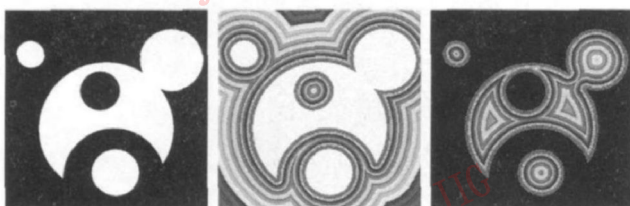
本文提出一种用带形状校正的腐蚀膨胀实现的 Euclidean 距离变换新算法。它有与 Chamfer 法相近的运行效率,同时具有能够校正形状失真,并可任意选定所需距离层数的优点。

1 距离变换及算法的改进

1.1 距离变换

距离变换有内外之分,所谓外距离变换可看作以区域 S 作为扩散源在区域的补集 S^c 中以相同速度向各个方向传播的距离波,距离波同一时刻到达的像素构成等值线,所有等值线构成距离变换图。而内距离变换则相反,可看作以区域的补集 S^c 作为扩散源在区域 S 内部以相同速度向区域中央传播的距离波形成的距离变换图^[1]。

图 1 给出一幅二值图像及其内外距离变换图,图像尺寸为 512×512 由不同尺寸的圆组成。



(a) 原二值图像 (b) 外距离变换图 (c) 内距离变换图

图 1 二值图像及其距离变换图

Fig 1 Binary image and its distance transform

1.2 算法的改进

在数字图像处理中,距离变换图是由不同距离值的等值线组成的。由此可见,距离变换与区域的腐蚀膨胀有着密切的联系。但是直到目前为止,只有将距离变换得到的距离图用于区域的腐蚀、膨胀;反之,还没有利用区域的腐蚀膨胀来生成距离变换图的算法。原因在于后者不仅效率低,而且形状还有较大的失真。

所谓点到区域的距离就是该点到区域的最短距离。确切地说,是该点到区域边界的最短距离。等值线是相同距离值的点的集合,图像范围内所有的等值线构成距离变换图。为了得到精确的距离变换图,理想的情况是等值线上的像素都能经过计算验证。粗看起来,这是一个效率极低的方法,存在组合爆炸的问题。然而,合理安排计算距离的点可以大大减少等值线形成过程中的距离计算次数以提高效率。本文提出的改进算法采用两个途径解决问题,即改进区域腐蚀膨胀算法以提高效率,再对初步结果进行误差校正以提高精度。前者已由文献 [8] 完成,本文介绍后者,即所得腐蚀膨胀结果中误差校正的方法。

2 等值线的获得与描述

2.1 区域边界的获得

由数学形态学基本运算规则可知,区域的内边界可以通过原区域与该区域腐蚀的结果相减得到。类似的有,区域的外边界可以通过该区域膨胀的结果与原区域相减得到^[5]。这样前面生成等值线的过程就转变为不断求取区域腐蚀膨胀后的边界的过程。

采用不同的结构元素所得到的边界是不同的。除了一些特殊需要外,最为常用的腐蚀膨胀用结构元素有两种,即 8 邻点与中心点组成的方形的 SQUARE(9) 结构元素与 4 邻点与中心点组成的十字形的 CROSS(5) 结构元素^[1]。

2.2 区域的线段表描述

水平线段集可以精确地表示区域,线段表则是描述水平线段集的数据结构。线段表的每个表项对应一条水平线段,它包含 Label, X_L , X_R , Y 和 $Flag$ 5 个参数,即表示上下线段连通关系的标记,左、右端点的 X 坐标,线段的 Y 坐标和表示线段特征或类型的标志^[4,9]。

图 2 给出了由 6 条水平线段组成的小区域, 表 1 给出了该区域的线段表。

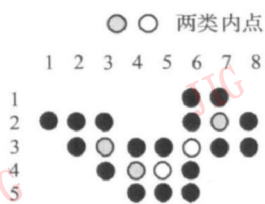


图 2 区域实例

Fig 2 Sample of A Area

表 1 区域的线段表

Tab 1 Line segment table of the region

	Label	X_L	X_R	Y	Flag
1	0	6	7	1	0
2	0	1	3	2	0
3	0	6	8	2	0
4	0	2	8	3	0
5	0	3	6	4	0
6	0	4	6	5	0

采用线段表可以实现连通区域的快速搜索算法, 其中种子填充快速算法适用于单区域提取, 线段编码则适用于全图像所有区域的标记^[10]。同时, 利用线段表表示区域还可参与区域的各种运算, 例如逻辑运算^[9]与形态学运算^[8]。

2.3 边界的线段表描述

线段表结构不但可以用来表示区域, 也可扩大范围应用于表示边界。为了便于统一处理, 线段表项的前 4 个参数与表示区域时使用相同的定义, 仅第 5 个参数 *Flag* 用于表示线段的类型。

可将表示边界的水平线段分为 5 种类型, 即左边界、右边界、中间线、顶线与底线, 它们分别用标志值 1 2 3 4 8 来表示。其中, 中间线表示处于边界凹陷部分的水平边界。

边界线表项中线段的类型标志既可用于区别实心区域与区域的边界, 又可用于边界腐蚀膨胀前控制填充其内部空腔。

表 2 给出了图 2 区域边界的线段表, 左面数据为 8 邻接边界的线段表。边界为 8 邻接时, 区域内点则为 4 邻接关系, 图中 5 个浅灰色和空心圆表示的点为内点, 它们没有包含在边界线中。右面数据为 4 邻接边界的线段表, 区域则用 8 邻接规则确定内点, 此时图中只有一个内点, 即空心圆表示的那个

点。表 2 中标出了两种邻接条件下发生变化的数据。

表 2 边界的线段表

Tab 2 Line segment table of region boundary

	8 邻接					4 邻接				
	Label	X_L	X_R	Y	Flag	Label	X_L	X_R	Y	Flag
1	0	6	7	1	4	0	6	7	1	4
2	0	1	3	2	4	0	1	3	2	4
3	0	6	6	2	1	0	6	6	2	1
4	0	8	8	2	2	0	7	8	2	2
5	0	2	2	3	1	0	2	3	3	1
6	0	4	5	3	3	0	4	5	3	3
7	0	7	8	3	2	0	6	8	3	2
8	0	3	3	4	1	0	3	4	4	1
9	0	6	6	4	2	0	6	6	4	2
10	0	4	6	5	8	0	4	6	5	8

2.4 线处理实现的腐蚀膨胀

文献 [9] 将操作对象为水平线段的运算称为线处理, 线处理的基础是同一行或上下相邻行两线段端点之间位置的比较, 前者用于逻辑运算, 后者用于连通判别与腐蚀膨胀。

直接采用传统的基于卷积方法的腐蚀膨胀运算来获取区域的内外边界效率是不高的。文献 [8] 提出了基于线处理的腐蚀膨胀快速算法极大地提高了处理的效率。与传统的卷积方法相比较运算速度一般可提高数十倍。

在本文所介绍的距离变换算法中, 整个过程只需在开始时生成区域的线段表, 仅此一次, 以后的处理都是基于线段表的运算。因此, 特别有利于处理效率的提高。

与图像的逻辑运算归结为像素与像素之间的同类逻辑运算相类似, 文献 [9] 证明了, 区域的逻辑运算可以分解为水平线上各分量同类逻辑运算结果之并。这样, 区域采用线段表表示后, 求取区域边界的减法操作就可以合并到腐蚀膨胀过程中同时进行, 由此处理的效率又能有所提高。

3 形状失真及其校正

3.1 腐蚀膨胀造成的失真

单次腐蚀、膨胀所造成的失真很小, 难以觉察。但若连续执行多次累积起来, 则可清楚地显现出来。图 3 给出了连续腐蚀膨胀 16 次的结果。

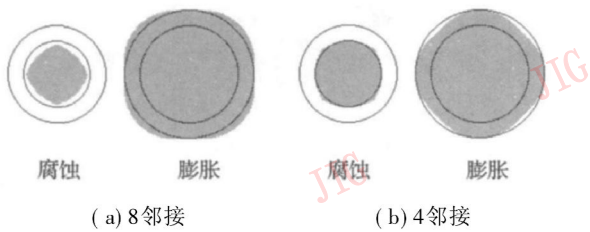


图 3 腐蚀膨胀造成的失真

Fig 3 Distortion from erosion and dilation

原始区域为半径 50 的圆,左面为连续腐蚀,右面为连续膨胀。图中浅灰色区域是腐蚀膨胀的结果,黑色圆周则为用作比较的基准。腐蚀图上基准圆半径分别为 50 与 34,膨胀图上基准圆半径分别为 50 与 66。其中半径为 50 的圆为腐蚀膨胀前的原始边界位置。

从图 3 不难得出腐蚀膨胀造成形状失真的 3 条规律:

- 1) 最终的误差小于腐蚀膨胀的次数,换句话说,不计累积作用的话,每次腐蚀膨胀造成的误差不超过一个像素。
- 2) 在水平与垂直方向失真很小,几乎没有失真,失真主要发生在对角方向。
- 3) 在 4 连通条件下,腐蚀膨胀慢于正常速度造成欠失真,即腐蚀膨胀达不到所需的要求;而在 8 连通条件下,腐蚀膨胀快于正常速度造成过失真,即腐蚀膨胀往往超出所需的要求。

由于每次腐蚀膨胀造成的误差不超过一个像素,腐蚀膨胀后及时进行校正就可使误差不致得以累积。采用 4 连通时补充未达目标的像素,采用 8 连通时消除超出目标的像素。

计算距离的两个点,一个为待校正点,在新形成的等值线上;另一个为基准点,在原始区域的边界线上。问题在于哪些点需要校正以及如何选取相应的基准点。由于误差小又采用了线段表结构,所谓校正只需修改待校正线段端点的 X 坐标,即根据需要增 1 或减 1。而基准点则只需在邻接的边界线段中间选取。

3.2 待校正点

距离变换图是数字图像,距离图上的距离值都作了取整操作。所以图中的等值线由一条条水平线段所组成。为了简化表述,这些组成等值线的水平线段在下文中简称为线段。如果得到的等值线没有误差,严格地说,误差小于相邻像素距离之半时,点到区域边界线上某一对应线段间的距离就是此点的

距离值。由于在本算法中等值线是通过腐蚀或膨胀得到的,它们的边界在形状上是相似的。此时根据几何常识,同一等值线上的点与原边界线之间分别计算距离值(浮点值),其中的最大值与最小值必定出现在它们所在线段的端点之间。

图 4 为腐蚀膨胀过程中左上方边界校正的情况。膨胀时采用边界左端点往内缩,腐蚀时采用边界右端点往内涨。图中待校正点 A 上的箭头为修正的方向。其他位置边界的情况与此相类似,为图 4(a)、图 4(b)图下面图形作镜像得到。

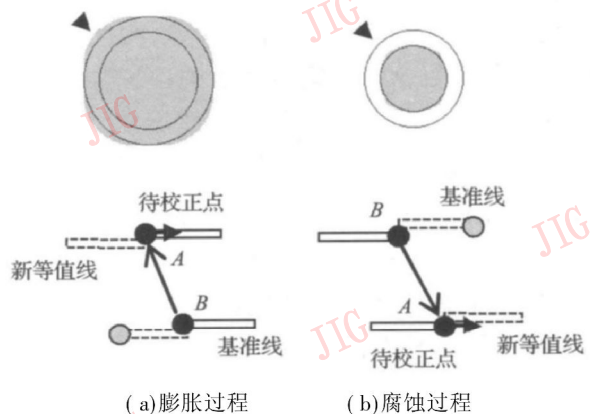


图 4 待校正点与基准点(左上方)

Fig 4 Point being regulated and the reference point (left upper)

3.3 待校正线段的编码

距离变换中计算距离的基准是 0 距离等值线,在外距离变换时它就是原区域的内边界,在内距离变换时它就是原区域的外边界。其次,距离变换图中的等值线之间虽然并非严格几何学意义上的相似图形,但仍是形状相似程度极高的准相似图形。这样,不同等值线上的线段间存在一定的对应关系。一般而言,对应线段的基准线段或者相同,或者为其相邻线段。

为了利用这种对应关系,可以对 0 距离等值线的线段进行编码。编码值为各线段在线段表中的序号,存储在各表项的 Label 参数中。在腐蚀膨胀过程中再将这些编码传递给新等值线上的对应线段。这样距离的校正可在新等值线与 0 距离等值线相同编码的线段之间进行,从而减少所需的搜索与比较时间。

编码传递的原则是,编码来自新线段关键坐标的来源线段。例如,膨胀时新线段为左边界线,则其关键点为左端点,此线段的编码就应来自当前等值

线上中下 3 条相邻线段中左端点坐标与膨胀结果相同的那一条。

3.4 基准线段的链接

等值线上的线段进行编码后可以缩小搜索范围,但还不能保证能够一次选中最佳基准点。由于误差小于 1 个像素,最佳基准点就在同编号的线段附近不会偏离太远,即集中在此线段及与其相邻的线段上。可从这些线段中寻找最短距离。这样,每个待校正点最多只需计算 3 次距离。如果再参考修正的方向,实际上只需两相邻线段之一,由此计算距离的次数可以减少到 2 次。

为此,在对原始等值线进行编码的同时还需将 0 距离等值线上线段之间的邻接关系记录下来。这种邻接关系可以通过链的结构连接起来。链接信息存放在每条线段对应表项的 *Label* 参数中,指向与它邻接的线段。链接时,左、右边界需分别进行。需要注意的是,线段的编码值存放在等值线的线段表中,而链接信息存放在区域原始边界的线段表中,两者是不同的,不能混淆。

3.5 校正的步骤

建立了上述两种数据结构之后,距离校正的过程就得以简化,即:

1) 在待校正的等值线上取出一条线段,由它的 *Label* 值找到原始边界上的对应线段,计算它们端点之间的距离,判别是否符合要求。

2) 若不符合要求,则通过原始边界的链结构找到与后者邻接的另一线段,继续进行距离计算并进行比较与校正。

3) 用较小距离线段的编码修改等值线线段原来的 *Label* 值,使得在后续过程中避免累积误差。

4 Euclidean 距离变换新算法

4.1 边界的收缩与扩展

为了与区域的腐蚀膨胀相区别,特将边界到其内外邻接边界的变换称为边界的收缩与扩展。这样,内距离变换时区域的外边界是 0 距离等值线,内距离变换的各等值线就变成由此线不断收缩而得到;外距离变换时区域的内边界是 0 距离等值线,外距离变换的各等值线就变成由此线不断扩展而得到。等值线收缩与扩展的中间过程还是区域的腐蚀与膨胀,只是变换过程的输入与输出都取距离等值线而已。

4.2 距离变换的步骤

以外距离变换为例介绍其变换步骤。

1) 建立工作单元

建立 3 个线段表和一个与图像尺寸相同的灰阶位图。一个线段表用于存放作为基准的内边界线,另外两个线段表用作存放等值线线段表的工作单元,分别存放源与目标数据。位图用于存放距离图像,开始时先清为零。

2) 生成初始线段表

(1) 生成区域的线段表。

(2) 生成区域内边界的线段表作为基准线。

(3) 复制内边界线段表到源工作线段表。

3) 初始线段表进行编码

(1) 源工作线段表的每个表项按序进行编码,序号记录在表项的 *Label* 标记中,此时的源工作线段为 0 距离等值线。

(2) 建立水平线段之间的链接关系,链接信息存放在原始边界线段表每条线段对应表项的 *Label* 参数中,指向与它邻接的线段。

4) 等值线扩展循环

(1) 由当前等值线出发扩展得到外层等值线。为了延续前后联系,等值线扩展时将前一等值线上线段的 *Label* 标记传递到后一等值线相应线段。

(2) 对新等值线按上节介绍的方法进行校正。

(3) 根据校正后的线段表结构的等值线在距离变换位图中用距离值绘制像素形式的等值线。

(4) 对等值线线段表进行裁剪并作结束判别。

(5) 两个工作线段表源与目标位置互换,重复循环中的步骤 (1) ~ (4),直至结束。

内距离变换的过程除了以原始区域的外边界作基准进行等值线收缩外,基本步骤与外距离变换相类似,但是具体处理方法还是有所不同。距离变换结果实例如图 1(b) 与图 1(c)。

5 新算法的效率与效果

5.1 效率分析

由于每个需要计算距离值的像素都在某一条等值线上,并且计算距离的次数不超过 2 次。因此,当图像尺寸为 $n \times n$ 时,本算法在最坏情况下计算距离的次数为 $O(n^2)$ 。

另外,本算法的效率还取决于腐蚀膨胀算法的

效率。而基于线段表的腐蚀膨胀算法的效率取决于图像的内容,特别适用于块状区域,即同一扫描线上只有较少由对象像素构成的线段的区域。在图像分割中,一般情况下待处理的图像多数能满足这一要求。

5.2 与 Chamfer 算法相比较

为了比较各种距离变换算法的运行效率与处理效果,图像尺寸为 512×512 , 其中有一个直径为 340 的圆形区域,采用 Intel Core 2 Duo E7200 CPU, 时钟频率 2.53 GHz, 内存 2 GB 的计算机。图 5 为不同算法的处理结果,表 3 为本文所涉 3 种距离变换方法进行内外距离变换的运行时间。

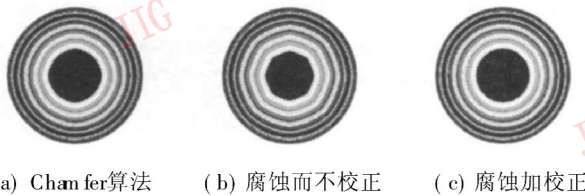


图 5 内距离变换图的形状失真

Fig 5 Distortion of inner distance transform

表 3 不同距离变换算法的运行时间

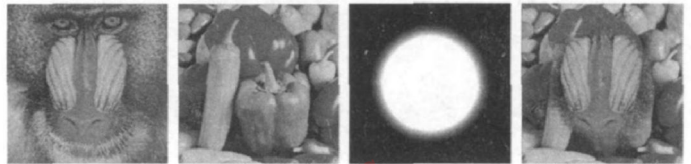
Tab 3 Operation time of different distance transforms /ms

	Chamfer 算法	腐蚀膨胀法	
	(5 × 5 模板)	不带校正	带校正
内距离变换	9.2	4.9	6.7
外距离变换	15.1	8.8	12.2

从表 3 可以看出,3 种方法中不作校正的腐蚀膨胀算法速度最快,几乎比采用传统 5×5 模板的 Chamfer 算法快一倍,但形状失真稍大,如图 5(a) 与图 5(b) 所示。而带校正的腐蚀膨胀算法在运行速度方面较 Chamfer 算法略有提高,而在形状保持方面则有了很大的改进,如图 5(c) 所示。

5.3 图像的羽化合成

文献 [2] 将距离变换用到了图像合成上,用于制作任意形状区域不同羽化宽度的蒙板。羽化蒙板由区域的内、外距离变换图经灰度变换后叠加而成。图 6 给出了羽化合成的原始图像、羽化蒙板与合成图像,表 4 为程序的运行时间。合成图像是幅面 512×512 的真彩色图像,圆形区域的直径为 340,羽化宽度为边界内外各 20 个像素。



(a) 源图像 (b) 目标图像 (c) 羽化蒙板 (d) 合成图像

图 6 图像的羽化合成

Fig 6 Image feather and blend

表 4 图像羽化合成算法的运行时间

Tab 4 Operation time of different feather algorithms /ms

	Chamfer 算法	腐蚀膨胀法	
	(5 × 5 模板)	不带校正	带校正
生成羽化蒙板	25.8	1.9	3.0
羽化合成	28.7	5.6	6.5

从表 4 中可以看出,本文介绍的距离变换在图像合成方面更为有效。因为 Chamfer 算法距离计算必须整幅图像一起进行,制作蒙板所需时间超过内、外距离变换所需时间之和。在较小羽化宽度时效率很低,故不适用于图像合成应用。而用腐蚀膨胀方法制作蒙板所需时间与羽化宽度有关,羽化宽度愈小时间愈节省。

与 Photoshop 软件羽化方法的不同在于本算法可用于任意形状的区域,而前者则需要将选区的边界先拟合成为光滑的规则曲线,然后再进行羽化处理^[11]。

6 结 论

本文提出一种用带形状校正的腐蚀膨胀实现的 Euclidean 距离变换新算法。该算法的特点是采用新的数据结构——线段表来表示区域与边界。对于用线段表表示的区域作腐蚀膨胀比用卷积型腐蚀膨胀效率提高数十倍。由于每次伸缩的误差不超过一个像素,只需校正新等值线各水平线段的端点。其次,边界的水平线段先按顺序进行编码并在等值线的伸缩过程中进行传递,距离值的计算在新等值线与边界线相同编码水平线段的端点间进行,每个像素距离计算次数的平均值小于 2。与传统基于局部距离累加的 Chamfer 算法相比较,本方法在保真度与处理效率两方面都有很大提高,距离误差不超过一个像素。

新的距离变换算法也可用于数字图像的合成,优点是生成羽化蒙板时形状保真度高并且运行速度

快。特别适用于任意形状区域可选宽度边界条带上的羽化处理。

线段表是一种描述区域与边界的新型数据结构,具有很强的区域与边界描述能力与灵活的操作特性。它不但可以用于快速提取连通区域,还可用于区域的腐蚀膨胀,以及用于区域的逻辑运算。加上本文介绍的距离变换,已在图像分割领域起到越来越大的作用。以上算法都属于线处理,相信随着应用范围的拓展,线段表和线处理会与传统的链码、点处理、邻域处理一样成为图像处理领域重要的数据结构与处理算法。

参考文献 (References)

- [1] Bovik A. Handbook of Image and Video Processing Second Edition [M]. Beijing Publishing House of Electronics Industry, 2006 45-53, 601-605
- [2] Lu Zong-qi, Zhu Yu. Principle and implementation of image blend [C] // Proceedings of the 14th ICIG. Beijing Tsinghua University Press, 2008 100-104 [陆宗骥, 朱煜. 图像合成原理及其实现 [C] /第十四届全国图象图形学学术会议论文集. 北京:清华大学出版社, 2008 100-104]
- [3] Castleman K R. Digital Image Processing[M]. Beijing Tsinghua University Press, 1998 475-477.
- [4] Lu Zong-qi. C/C++ Programming of Image Processing [M]. Beijing Tsinghua University Press, 2005 246-250, 319-326 [陆宗骥. C/C++ 图像处理编程 [M]. 北京:清华大学出版社, 2005 246-250, 319-326]
- [5] Gonzalez R C, Woods R E. Digital Image Processing [M]. 2nd ed. Beijing Publishing House of Electronics Industry, 2002 519-549
- [6] Liu Xiang-bin, Zou Bei-ji, Sun Jianguang. Fast euclidean distance transform based on contour tracking [J]. Chinese Journal of Computer, 2006, 29 (2): 317-323 [刘相滨, 邹北骥, 孙家广. 基于边界跟踪的快速欧氏距离变换算法 [J]. 计算机学报, 2006, 29 (2): 317-323]
- [7] Lu Ji-wen, Zhang Er-hu. A fast algorithm for complete euclidean distance transformation based on contour scanning [J]. Journal of Image and Graphics, 2006, 11 (6): 872-876 [鲁继文, 张二虎. 基于围线扫描的快速完全欧氏距离变换算法 [J]. 北京:中国图象图形学报, 2006, 11(6): 872-876]
- [8] Lu Zong-qi, Zhu Yu. Fast algorithm of mathematical morphology erosion and dilation operation [C] // Proceedings of the 13th ICIG. Beijing Tsinghua University Press, 2006 306-311. [陆宗骥, 朱煜. 数学形态学腐蚀膨胀运算的快速算法 [C] /第十三届全国图象图形学学术会议论文集. 北京:清华大学出版社, 2006 306-311]
- [9] Zhu Y, Lu Z, Liu M, et al. The Line Segment Table — A Fast Region Description Algorithm [EB]. Journal of RealTime Image Processing, 2008, Online DOI:10.1007/s11554-008-0089-6
- [10] Lu Zong-qi, Zhu Yu. Fast algorithm of Line segment Encoding and seed filling based on line segment table [J]. Journal of Image and Graphics, 2009, 14(3): 499-504 [陆宗骥, 朱煜. 使用线段表实现线段编码与种子填充快速算法 [J]. 北京:中国图象图形学报, 2009, 14(3): 499-504]
- [11] Adobe Corporation. Adobe Photoshop CS [CP]. 2003