

区域的深度计算及其在图象渐变处理中的应用

朱述龙

(解放军信息工程大学测绘学院遥感信息工程系, 郑州 450052)

摘要 为了解决局部图象处理容易产生边界效应的问题, 可以区域中任意一点的深度值为基础, 来得到权函数的数学表达式, 然后根据权函数对图象上给定区域范围内的象素进行处理, 使区域中离区域边界较近的象素的灰度值改正量小一些, 而使区域中离区域边界较远的象素的灰度值改正量大一些, 从而实现区域图象的渐变处理. 另外, 为了克服迭代算法速度慢的缺点, 还提出了一种快速标记多边形区域内部点的深度值算法. 实验结果表明, 所提出的渐变处理方法, 在克服区域图象增强处理时容易产生的边界效应方面是十分有效的, 可以获得满意的效果.

关键词 多边形区域 深度值的计算 图象的渐变处理

中图法分类号: TP751 文献标识码: A 文章编号: 1006-8961(2000)09-0734-05

The Computation of Region Depth and Its Application to Image Feathering

ZHU Shu-long

(Department of Remote Sensing and Information Engineering Institute of Surveying and Mapping the PLA University of Information Engineering, Zhengzhou 450052)

Abstract In order to resolve the problem of image feathering in the region, the formula of weight function is presented based on the depth information of the region and every pixel in the region is processed according to the weight. As a result, the closer the pixel is to the region boundary, the less the correction of pixel value is, and the farther the pixel is to the region boundary, the more the correction of pixel value is. This method of image feathering is also tested using real images. In order to overcome the time consuming disadvantage of iterative algorithm for computation of region depth, a fast algorithm is also proposed and tested using convex, concave and cross polygon region. The experiment shows that the method proposed above is very effective and can acquire good results in image feathering of the region.

Keywords Polygon region, The computation of depth value, Image feathering

0 引言

在数字图象的交互式处理中, 渐变处理是一项十分重要的处理方式, 它有许多重要的应用. 例如由于各种因素的影响, 图象上局部范围内的影象较暗, 这时需要有一种方法将局部影象的亮度修正过来. 为了解决这个问题, 通常情况下, 是首先采用人机交互式的方法确定图象上被处理区域(以下简称区域)的范

围, 然后, 选择某种算法对该区域内的图象进行处理. 为了使处理后的图象在区域边界处不留下明显的边界效应, 一般需采用渐变处理策略, 使区域内部, 距离区域边界较近的象素的灰度值改正量小一些; 而使区域内部, 距离区域边界较远的象素的灰度值改正量大一些. 这种区域图象渐变处理的关键是: 适当给出灰度处理时, 区域范围内各象素的权值, 并且使该权值与该象素到区域边界的距离成反比, 而与该象素到区域中轴线的距离成正比. 对于规则区域(如矩形区域、

椭圆形区域等), 由于它有固定的中心, 因此权值很容易用象素到规则区域中心点的距离的倒数来表示. 但对于不规则的任意多边形区域, 由于它没有固定的几何中心, 因此权值很难用一个固定的函数来表达. 为了能够对任意多边形区域实施区域图象的渐变处理, 并使处理后的图象不出现明显的边界效应, 本文用区域中任意一点的深度值作为权函数来对图象上给定区域范围内的象素进行处理, 并提出了一种计算区域深度的快速算法. 实验结果表明, 本文提出的渐变处理方法, 在克服区域图象增强处理时容易产生的边界效应方面, 其效果是十分明显的.

1 区域深度及其快速计算

设图象 I_R 中有一个区域 R , 它的象元值为“1”; 其背景值为“0”, 背景区域记为 \bar{R} ; 则区域 R 中任意一象素到 \bar{R} 的距离定义为区域中该象素的深度值. 此处两个象素 p, q 之间的距离 $d(p, q)$ 是用邻域的层次来定义的, 若 p 和 q 为相邻象素, 则 $d(p, q) = 1$. 由于我们用邻域的层次来定义两个象素之间的距离, 因此八邻域中的距离和四邻域中的距离是不一致的. 如图 1 所示, 对于四邻域来说, 象素 p 的上、下、左、右象素到 p 的距离为 1, 而它四个斜角上的象素到 p 的距离为 2; 而对于八邻域来说, 象素 p 的上、下、左、右和四个斜角上的象素到 p 的距离都为 1.

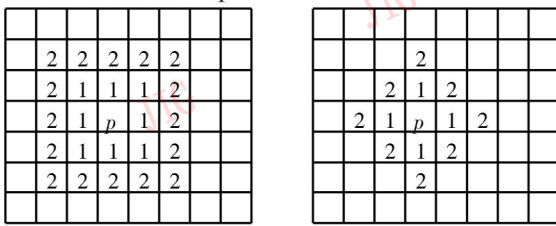


图 1 象元 p 与其邻域象元之间的距离

在邻域关系确定以后, 区域 R 中任意一象素 p 到 \bar{R} 的距离可按下列公式计算

$$I_R^{(m)}(p) = I_R^{(0)}(p) + \min | I_R^{(m-1)}(q) | \quad (1)$$

式中, q 为 p 的邻域象素, $I_R^{(m)}(p)$ 表示区域 R 经过 m 次标记以后 p 点的深度值, 其中 $I_R^{(0)} = I_R$ 为原始图象.

公式(1)是一个迭代公式, 每迭代一次, 内缩一层边界以后的区域范围内的象素值加“1”, 这样经过 m 次迭代以后, 区域中任意一点的象素值就是该象素的深度值. 由于公式(1)是一个迭代公式, 当区域 R 很大时, 完成区域深度的标记工作需要很长的时

间. 一般地, 对于最大深度为 d 的区域, 共需要迭代 $d-1$ 次. 为了解决迭代算法需要大量时间消耗的问题, 本文提出了如下的区域深度快速标记算法, 它只需对图象 I_R 扫描两次, 就可以将区域 R 中所有象素的深度值标记出来. 即

设 N_1 是 p 点 (x, y) 的八邻域中的前半邻域 (图 2 中的深灰色部分), N_2 是其八邻域中的另外一半 (图 2 中的浅灰色部分), 即

$$N_1 = \{(x-1, y), (x-1, y-1), (x, y-1), (x+1, y-1)\}$$

$$N_2 = \{(x+1, y), (x+1, y+1), (x, y+1), (x-1, y+1)\}$$

则

$$I'_R(p) = \begin{cases} 0 & p \in \bar{R} \\ \min_{q \in N_1} \{I'_R(q)\} + 1 & p \in R \end{cases} \quad (2)$$

$$I''_R(p) = \begin{cases} 0 & p \in \bar{R} \\ \min_{q \in N_2} \{I'_R(p), I''_R(q) + 1\} & p \in R \end{cases} \quad (3)$$

其中, I'_R 是对图象进行第一遍扫描后的结果, 第一遍扫描时是按照从上到下、自左而右的顺序进行扫描, 且象素 p 的邻域仅考虑 N_1 ; I''_R 是对图象进行第二遍扫描后的结果, 第二遍扫描时是按照从下到上、自右而左的顺序进行扫描, 且象素 p 的邻域仅考虑 N_2 . 上述快速算法是以八邻域距离为基础的, 当 N_1 定义为 p 点 (x, y) 的四邻域中的前半邻域, N_2 定义为四邻域中的另外一半邻域时, 即

$$N_1 = \{(x-1, y), (x, y-1)\}$$

$$N_2 = \{(x+1, y), (x, y+1)\}$$

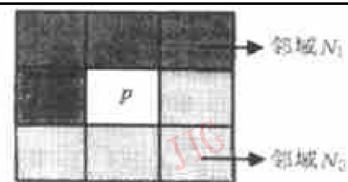


图 2 邻域 N_1, N_2 的定义

公式(2)、(3)便是以四邻域距离为基础的区域深度快速标记算法.

公式(1)所描述的迭代算法的运算量不仅与图象的大小有关, 还与图象上待标记区域的大小有关. 设图象的长、宽分别为 M 和 N , 图象上待标记区域的最大深度为 d , 则按公式(1)完成该图象上区域标记所需的运算量(以八邻域中的距离为例)为

$$(d-1) \times [M \times N \times (1 \text{ 次加法运算} + 8 \text{ 次比较运算})]$$

而按公式(2)、(3)完成该图象上的区域标记所需的运算量(以八邻域距离为例)最大为

$$\begin{aligned}
 &M \times N \times (1 \text{ 次加法运算} + 4 \text{ 次比较运算}) + \\
 &M \times N \times (4 \text{ 次加法运算} + 4 \text{ 次比较运算}) = \\
 &M \times N \times (5 \text{ 次加法运算} + 8 \text{ 次比较运算})
 \end{aligned}$$

显然,当区域的最大深度 d 很大时,则公式(1)的运算量远远大于公式(2)和(3)的运算量.因此本文提出的

区域深度标记算法较经典的迭代算法要快很多.

下面用一个例子说明两种算法的结果是一致的,其中,图3形象地描述了按公式(1)进行图象上区域标记的过程,图4形象地描述了按公式(2)和(3)进行图象上区域标记的过程,从这个例子可以看出:虽然两者的实现方法不一样,但两者的最终结果却是完全一致的.

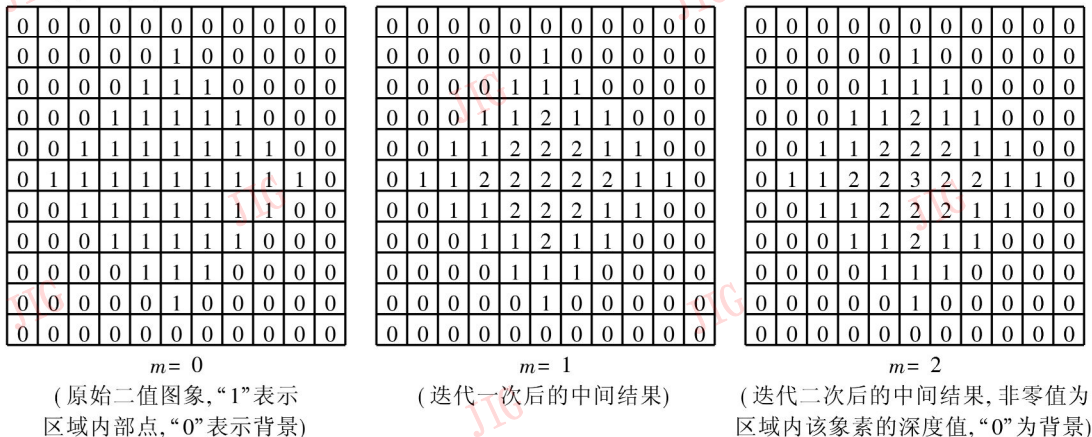


图3 按公式(1)进行图象上区域标记的过程

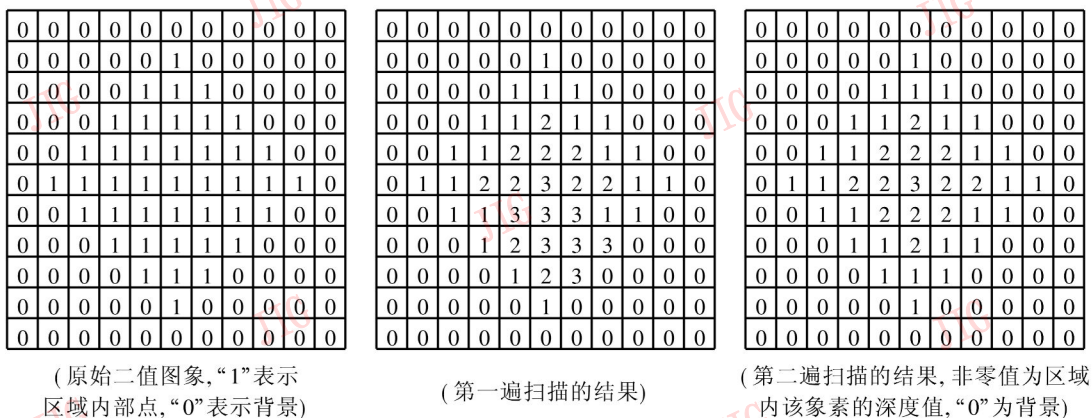


图4 按公式(2)、(3)进行图象上区域标记的过程

区域深度标记是寻找区域中轴线的关键,在中轴变换、区域细化、区域表示等处理中有着重要的应用^[1,2].

2 区域的深度信息在图象渐变处理中的应用

下面以图象增强中的均值方差变换^[3]为例,说明图象渐变处理的实现过程.

首先讨论如何确定区域图象渐变处理的范围.如图5所示,设图象上需要增强的区域为 R ,为了使增强以后的图象在区域 R 的边界 B 位置不出现灰度突变现象,需采用渐变处理方法,使图象上处理后的灰度值,在渐变处理宽度为 d 的环形区域内(图5

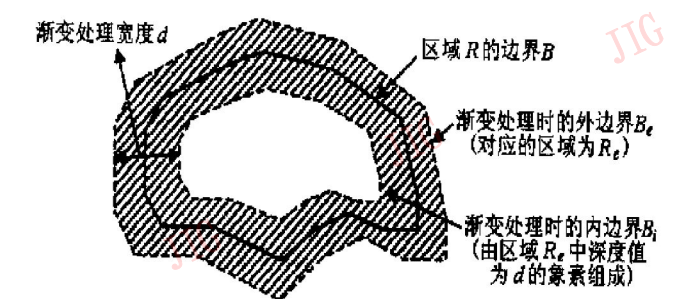


图5 渐变处理范围的确定

中晕线部分)由内边界 B_i 向外边界 B_e 逐渐变化.在用户用鼠标给出区域 R 的边界 B 以后,可采用多边形区域填充算法^[4]将由边界 B 的矢量数据确定的区域 R 转化为栅格形式的二值图象 I_R ,并且区域 R 中的象元值为“1”,背景象元值为“0”.再利用图象形态学中的膨胀运算^[5]将区域 R 膨胀成区域 R_e ,则区

域 R_e 的边界即为渐变处理时的外边界 B_e , 由区域 R_e 中深度值为 d 的像素组成的等值线, 即为渐变处理时的内边界 B_i ; 然后按式(2)和式(3)求区域 R_e 中各点的深度值; 最后以区域 R_e 中各点的深度值为基础, 来确定区域 R_e 中各点在增强处理时的权函数, 这样就可以实现区域范围内灰度渐变处理的目的. 为了节约存储量和减少计算量, 区域 R_e 中各点深度值的计算只在由 R_e 的外边界 B_e 所确定的外接矩形范围内进行.

其深度值具体计算为:

设区域中像素点 (i, j) 处的深度值为 l , 则该点在增强处理时的权值 w 定义为

$$w = \begin{cases} 1 & \text{当 } l > d \text{ 时} \\ \frac{l}{d} & \text{当 } l \leq d \text{ 时} \end{cases} \quad (4)$$

式中, d 为渐变处理宽度, 为已知值, 由用户直接给出.

我们知道, 传统的均值方差变换算法^[3]为

$$y(i, j) = \frac{\sigma_x}{\sigma_y} [x(i, j) - \mu_x] + \mu_y \quad (5)$$

式中, $x(i, j)$ 、 μ_x 、 σ_x 分别表示待处理图象上像素点 (i, j) 处的灰度值、待处理图象的灰度均值和灰度方差, $y(i, j)$ 、 μ_y 、 σ_y 分别表示处理后的图象上像素点 (i, j) 处的灰度值、处理后的图象的灰度均值和灰度方差, 其中 μ_y 、 σ_y 一般由用户采用交互式的方法给出. 若将式(5)改写为

$$y(i, j) = x(i, j) + \frac{\sigma_x}{\sigma_y} [x(i, j) - \mu_x] + \mu_y - x(i, j) \quad (6)$$

则引入权值 w 后的均值方差变换算法就为

$$y(i, j) = x(i, j) + w \cdot \left| \frac{\sigma_x}{\sigma_y} [x(i, j) - \mu_x] + \mu_y - x(i, j) \right| \quad (7)$$

上式实际上是将该像素在增强处理时的灰度值改正量, 按权值大小作相应的调整, 以便使图象灰度值的改正量在渐变处理宽度为 d 的环形区域内(图 5 中晕线部分), 由内边界 B_i 向外边界 B_e 从大到小地逐渐变化, 从而实现区域图象的渐变处理. 这种操作形象地称为“羽化处理(feathering)”.

类似地, 我们可以得到以区域深度值为基础的其他图象区域增强算法.

3 实验和结论

为了验证区域深度快速标记算法的正确性, 因而分别在图象上用鼠标画出了 3 种复杂情况的多边形, 即凸多边形、凹多边形、交叉多边形, 然后采用奇偶检校算法将多边形内部的像素灰度值变成 1, 多边形外部的像素灰度值变成 0, 最后按照式(2)、式(3)标记多边形区域的深度值, 其结果如图 6 所示. 从实验结果可以看出, 不论多边形的形状如何复杂, 本文提出的算法都可以较好地将多边形区域内部的深度值标记出来, 且该算法的运算量不随区域的大小变化而显著变化.

为了验证本文提出的图象渐变处理方法的有效性, 又用实际图象进行了实验, 图 7 为凹多边形的情况, 图 8 为交叉多边形的情况. 图 7、图 8 中(a)为待处理的图象; 图 7、图 8 中(b)是未采用渐变方法进行增强处理的结果, 此时在图象上区域边界处产生了明显的边界效应; 图 7、图 8 中(c)是采用了渐变方法进行增强处理的结果(渐变处理宽度 $d=30$ 像素), 从图上可见, 此时在区域边界处已没有明显的边界过渡. 实验表明: 本文提出的渐变处理方法, 在克服区域图象增强处理时容易产生的边界效应方面是十分有效的, 而且区域的深度值一旦标记出来以后, 权值很容易给出.



(a) 凸多边形区域

(b) 凹多边形区域

(c) 交叉多边形区域

图 6 几种多边形区域的深度信息(图中深度值的大小用亮度表示)

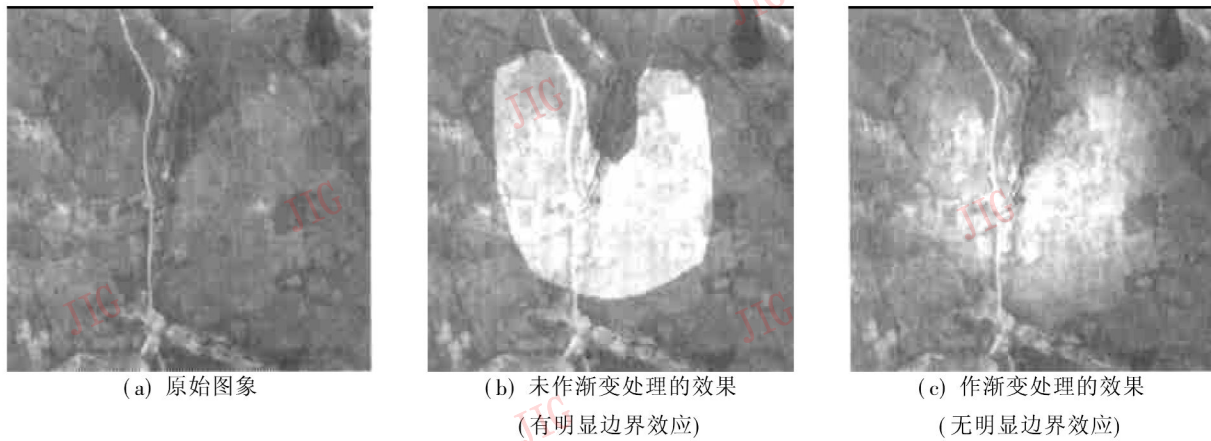


图7 凹多边形区域图象的增强处理

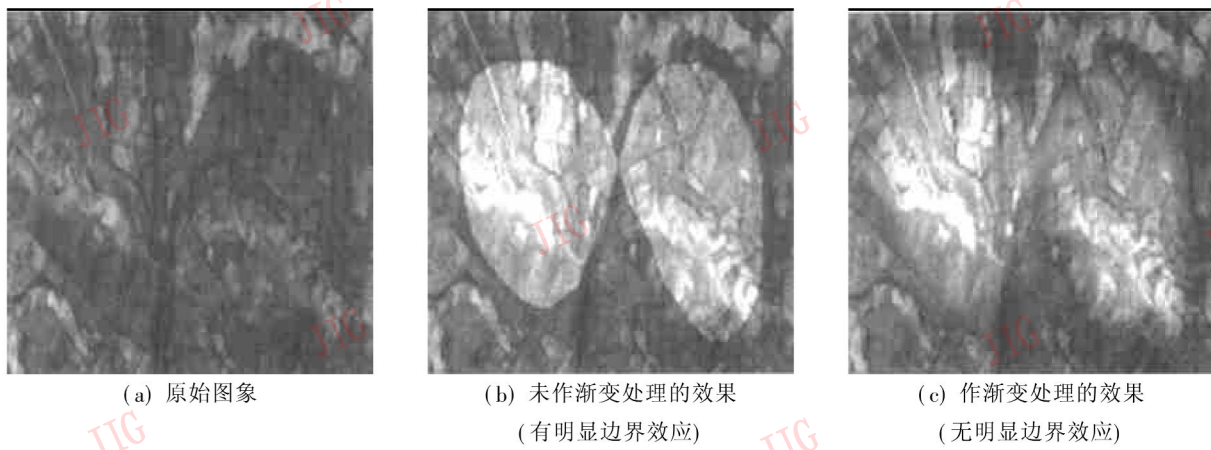


图8 交叉多边形区域图象的增强处理

参考文献

- 1 朱焱良. 计算机视觉. 浙江: 浙江大学出版社, 1997, 173~ 178.
- 2 钟玉琢、乔秉新、李树青. 机器人视觉技术. 北京: 国防工业出版社, 1994, 360~ 361.
- 3 沈邦乐. 计算机图象处理. 北京: 解放军出版社, 1995, 93.
- 4 孙家广、杨长贵. 计算机图形学. 北京: 清华大学出版社, 1997, 178~ 186.
- 5 Jisheng Song, Edward J Delp. The analysis of morphological filters with multiple structuring elements. Computer Vision, Graphics and Image Processing, 1990, 50: 308~ 328.



朱述龙 1964年生, 1997年获解放军测绘学院工学博士学位, 现为解放军信息工程大学测绘学院遥感信息工程系副教授. 曾获军队科技进步二、三等奖各1项, 公开发表学术论文近30篇, 专著2部. 主要研究方向为遥感信息处理、图象识别和数字摄影测量.