

二维栅格图象距离图的 的计算与分析

李建平 史济民

(计算机与自控系)

〔摘要〕 本文着重讨论和分析了二维栅格图象各种距离图的计算算法以及运算量、遍历次数和误差值等。作者首先定义了几个概念,包括距离和距离图,然后提出了用于计算N-邻居距离图和对角线距离图的算法A和用于计算八边形距离图的算法B,并用二组遍历蒙片直观地描述这两个算法。最后,对各种距离图的计算量、遍历次数和误差值进行了详细的比较。

关键词: 细化; 距离图; 栅格图象; 路径; 邻居; N-序列

在二维或高维栅格图象的某些计算机处理算法中,都不同程度地牵涉到栅格间距离的概念。G. Borgefors 曾经从直观意义上定义并且分析了几种不同类型的距离以及它们与欧氏距离之间的误差,但对各种距离的计算算法以及相互之间的比较情况,都未做详细的研究和进一步论述。尤其是,在大多数的图象处理中,主要关心的是栅格离其最近周界外栅格的距离即距离图的概念。因此,本文着重探讨计算几种典型距离与距离图的算法,并就有关计算量及误差等指标进行比较。

1. 基本概念及其记号

用 Z^2 表示由所有整数二元组组成的二维栅格图象空间,用 $F = \{f(\vec{v})\}$ 表示在 Z^2 中的一幅二值离散栅格图象,其中 $\vec{v} \in Z^2$ 即 $\vec{v} = (i, j)$, 表示 Z^2 中的某一栅格。下面定义几个在 Z^2 中的有关图象和距离的概念。

1.1 直接邻居和对角邻居

对于图象 $F = \{f(\vec{v})\}$ 中的栅格 $\vec{v} = (i, j)$, 称栅格 $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, $(i, j+1)$ 为它的直接邻居; 称栅格 $(i-1, j-1)$, $(i+1, j+1)$, $(i-1, j+1)$, $(i+1, j-1)$ 为它的对角邻居。

1.2 毗邻类型 N 和 N-序列

在 Z^2 中, 一个毗邻类型 N 是指这样一个对 (P, d) , 其中 P 是 Z^2 的一个有限

本文于1988年6月22日收到。

子集, d 是定义在 P 上的一个算术函数。而 N -序列是指一个或几个毗邻类型的序列, 记为 q , 即假设 N_0, N_1, \dots, N_{t-1} 是 Z^2 上的 t 个毗邻类型, 则:

$$q = N_0 N_1 N_2 \cdots N_{t-1}, \text{ 其中 } N_i = (P_i, d_i), i \in \{0, t-1\}$$

称 t 为 N -序列 q 的周期, 用 $|q|$ 表示。

1.3 路径和路径长度

对于图象 $F = \{f(\vec{v})\}$ 中的两个栅格 \vec{u} 和 \vec{v} , 如果存在如下的一个栅格序列:

$$X: \vec{x}_0 (= \vec{u}), \vec{x}_1, \dots, \vec{x}_m (= \vec{v})$$

使得下列条件成立:

$$\vec{x}_{i+1} - \vec{x}_i \in P_{\text{mod}(i,t)}, i \in \{0, m-1\}, \text{mod 为求余运算,}$$

则称 X 是由 N -序列 q 决定的、周期为 t 的、从 \vec{u} 到 \vec{v} 的多毗邻类型路径, 简称路径。从 \vec{u} 到 \vec{v} 的所有这样的路径集合记作 $PATH_q(\vec{u}, \vec{v})$ 。

路径长度是指路径中每对相邻栅格间的距离之和, 用 $L_q(X)$ 表示如下:

$$L_q(X) = \sum_{i=0}^{m-1} d_{\text{mod}(i,t)}(\vec{x}_{i+1} - \vec{x}_i)$$

1.4 距离和距离图

对于 $F = \{f(\vec{v})\}$ 中的栅格 \vec{u} 和 \vec{v} 间的距离, 记作 $D_q(\vec{u}, \vec{v})$, 定义如下:

$$D_q(\vec{u}, \vec{v}) = \min\{L_q(X): X \in PATH_q(\vec{u}, \vec{v})\}$$

如果 $PATH_q(\vec{u}, \vec{v})$ 为空集, 则 $D_q(\vec{u}, \vec{v}) = \infty$ 。

对于图象 $F = \{f(\vec{v})\}$ 和一个 N -序列 q , 图象 F 的距离图可用 $D = \{D(\vec{v})\}$ 表示。式中的 $D(\vec{v})$ 表示图象 F 中的栅格 \vec{v} 离其最近周界外栅格的距离, 这类距离在图象细化处理中经常使用。在二维栅格图象中, 且有较大使用价值的距离图有 N -邻居距离图、对角线距离图和八边形距离图。

2. N-邻居距离图

N -邻居距离图有 N_0 -邻居距离图和 N_1 -邻居距离图两种, 分别由 N -序列 $q_0 = N_0 = (P_0, d_0)$ 和 $q_1 = N_1 = (CP_1, d_1)$ 定义, 其中:

$$P_0 = \{(-1, 0), (1, 0), (0, 1), (0, -1)\}$$

$$d_0(\vec{p}) = 1, \vec{p} \in P_0$$

$$P_1 = \{(-1, 0), (1, 0), (0, 1), (0, -1), (1, 1), (1, -1), (-1, 1), (-1, -1)\}$$

$$d_1(\vec{p}) = 1, \vec{p} \in P_1$$

先考查 N_1 -距离图。下文算法 A 用于计算 N_1 -距离图。

算法 A :

(1) 计算初始距离图 $D^{(0)} = \{D^{(0)}((i, j))\}$

$$D^{(0)}((i, j)) = \begin{cases} C & \text{如果 } f((i, j)) = 1 \\ 0 & \text{如果 } f((i, j)) = 0 \end{cases}$$

(2) 计算中间距离图 $D^{(1)} = \{D^{(1)}((i, j))\}$

$$D^{(1)}((i, j)) = \min\{D^{(0)}((i, j)), D^{(1)}((i-1, j-1)) + d_2', \\ D^{(1)}((i-1, j)) + d_1', D^{(1)}((i-1, j+1)) + d_2', \\ D^{(1)}((i, j-1)) + d_1'\}$$

(3) 计算中间距离图 $D^{(2)} = \{D^{(2)}((i, j))\}$

$$D^{(2)}((i, j)) = \min\{D^{(1)}((i, j)), D^{(2)}((i+1, j+1)) + d_2', \\ D^{(2)}((i+1, j)) + d_1', D^{(2)}((i+1, j-1)) + d_2', \\ D^{(2)}((i, j+1)) + d_1'\}$$

其中 $d_1' = d_2' = 1$, 常数 C 是比距离图 D 中任一值都大的整数。

作者已证明 $D^{(2)}$ 就是由 N -序列 $q_1 = N_1$ 所决定的距离图, 但由于证明占用篇幅较大, 在此, 仅用图 1 中的向前和向后遍历蒙片来阐述这一算法。在图 1 所示的遍历蒙片中, 值为 0 的栅格称为蒙片 0 栅格, 其余栅格的值表示该栅格与蒙片 0 栅格之间的距离值即 d_1' 和 d_2' 。为了求得图象 $F = \{f(\vec{v})\}$ 的距离图, 首先用向前遍历蒙片, 从图象左上角开始, 从左到右、从上到下进行向前遍历。在遍历的过程中, 当蒙片覆盖图象 F 的某一栅格区域时, 就计算出所有被覆盖的图象栅格值与蒙片栅格值之和, 并取极小值, 作为被蒙片 0 栅格所覆盖的图象栅格值, 这一栅格常被称为图象的当前栅格。这样计算出的栅格值, 实际上是该栅格离图象左上角最近周界外栅格的距离, 即形成了中间距离图 $D^{(1)}$ 。当向前遍历完成以后, 接着用向后遍历蒙片, 从图象的右下角开始, 从右到左、从下到上, 对 $D^{(1)}$ 进行一次向后遍历。由这次向后遍历所计算出的栅格值, 是当前栅格离图象右下角最近周界外栅格的距离。将这个距离与上次遍历所得的距离进行比较并取较小值, 便获得当前栅格离其最近周界外栅格的距离, 从而获得距离图, 即 $D^{(2)}$ 就是由 N -序列 q 决定的图象 F 之距离图。

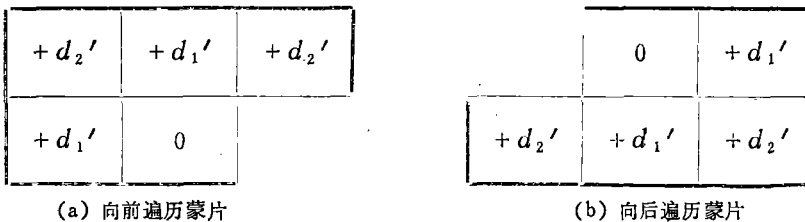


图 1 二维栅格图象距离图计算的遍历蒙片

由上可知, 在计算 N_1 -距离图时, 蒙片中的 d_1' 和 d_2' 为 1, d_1' 表示当前栅格与其直接邻居之间的距离值, 而 d_2' 表示当前栅格与其对角邻居之间的距离值。这些蒙片以及蒙片栅格值完全由 N -序列 $q_1 = N_1$ 决定, 即由 P_1 和 d_1 决定。

现在考查 N_0 -距离图, 该距离图由 N -序列 $q_0 = N_0$ 决定, 即只考虑一个栅格的直接邻居, 不考虑其间接邻居, 也即 $d_1' = 1, d_2' = \infty$ 。因此仍然可以使用算法 A 来计算 N_0 -距离图, 只不过此时 $d_2' = \infty$ 。同样也可用图 1 的遍历蒙片来阐述计算 N_0 -距离图的算法, 只不过此时不考虑由 d_2' 所标记的蒙片栅格。

不难发现, N_0 -距离图和 N_1 -距离图是有差异的, 不同的 N -序列将得出不同的距离值, 研究欧氏距离与不同 N -序列所决定的距离值间的误差, 对选择一种合适的距离图是非常重要的。欧氏距离与 N_0 -邻居距离和 N_1 -邻居距离的最大误差分别为 $-0.59M$ 和 $0.41M$, 其中 M 代表图象的大小。可见, N_0 -邻居距离总比欧氏距离大, 而 N_1 -邻居距离总比欧氏距离小。

3. 对角线距离图

对角线距离图其距离由 N -序列 $q_2 = N_2 = (P_2, d_2)$ 定义, 其中:

$$P_2 = \{(0, 1), (0, -1), (1, 0), (-1, 0), (1, 1), (1, -1), (-1, 1), (-1, -1)\}$$

$$d_2(\vec{p}) = 1, \text{ 如果 } \vec{p} \in \{(0, 1), (0, -1), (1, 0), (-1, 0)\}$$

$$d_2(\vec{p}) = \sqrt{2}, \text{ 如果 } \vec{p} \in \{(1, 1), (1, -1), (-1, 1), (-1, -1)\}$$

由定义不难看出, 我们仍然可以利用算法 A 来产生对角线距离图, 不过此时 $d_1' = 1, d_2' = \sqrt{2}$ 。类似地, 也可以用图 1 的遍历蒙片来阐述这一算法。

对角线距离图的最大误差值为 $-0.09M$ 。更进一步考察, 假设 $d_1' = 1$, 通过研究最大误差值的极小值, 可以求出最优的 d_2' , 使得最大误差为最小值。值得注意的是, 当 $d_1' = 1$ 时, d_2' 宜取 $[1, 2]$ 之间的数, 并可算得这样的最优值 $d_2' = 1.351$ 。这种最优的对角线距离, 其最大误差值为 $\pm 0.06M$ 。

在实际应用中, 为了避免浮点数运算, 常常取 d_1' 和 d_2' 为整数值。一组接近最优值 1.351 的整数 d_1' 和 d_2' 为 $d_1' = 3$ 和 $d_2' = 4$, 其最大误差值为 $0.08M$ 。这种整数对角线距离常称为对角线 (3, 4) 距离, 其相应的距离图则称为对角线 (3, 4) 距离图。

4. 八边形距离图

N_0 -邻居距离总是大于欧氏距离, 而 N_1 -邻居距离又总是小于欧氏距离。如果把这两种距离结合起来, 也即路径依次取前面定义过的 P_0 和 P_1 中的邻居, 这样所得到的距离就是所谓的八边形距离。其距离值由 N -序列 $q_3 = N_0 N_1$ 定义, 相应的距离图称为八边形距离图。下文的算法 B 就是用来计算栅序格图象 F 的八边形距离图 $D = \{D(\vec{v})\}$ 的。

算法 B :

(1) 计算初始距离图 $D^{(0)} = \{D^{(0)}(\vec{v})\}$

$$D^{(0)}((i, j)) = \begin{cases} C & \text{如果 } f((i, j)) = 1 \\ 0 & \text{如果 } f((i, j)) = 0 \end{cases}$$

(2) 依次计算出中间距离图 $D^{(2)}$, $D^{(1)}$, $D^{(3)}$, $D^{(4)}$

$$D^{(1)}((i, j)) = \min\{D^{(0)}((i, j)), 1 + D^{(1)}((i-1, j)), \\ 1 + D^{(1)}((i, j-1)), 1 + D^{(1)' }((i-1, j-1))\}$$

$$D^{(2)}((i, j)) = \min\{D^{(1)}((i, j)), 1 + D^{(2)}((i+1, j)), \\ 1 + D^{(2)}((i, j+1)), 1 + D^{(2)' }((i+1, j+1))\}$$

$$D^{(3)}((i, j)) = \min\{D^{(2)}((i, j)), 1 + D^{(3)}((i-1, j)), \\ 1 + D^{(3)}((i, j+1)), 1 + D^{(3)' }((i-1, j+1))\}$$

$$D^{(4)}((i, j)) = \min\{D^{(3)}((i, j)), 1 + D^{(4)}((i+1, j)), \\ 1 + D^{(4)}((i, j-1)), 1 + D^{(4)' }((i+1, j-1))\}$$

其中 $D^{(K)' }((i, j))$, $K \in \{1, 4\}$, 定义如下:

$$D^{(K)' }((i, j)) = \begin{cases} D^{(K)}((i, j)), & \text{如果 } D^{(K)}((i, j)) \text{ 是奇数} \\ C, & \text{如果 } D^{(K)}((i, j)) \text{ 是偶数, } C \text{ 同算法 A} \end{cases}$$

同理, 在此只用图 2 所示的遍历蒙片来阐述算法 B。首先, 用遍历蒙片 (a), 从左到右、从上到下遍历初始距离图 $D^{(0)}$ 。在遍历的过程中, 分别进行求和并取极小值运算, 且将该极小值作为当前栅格的栅格值。完成向前遍历后, 得到中间距离图 $D^{(1)}$ 。然后用遍历蒙片 (b), 从右到左、从下到上遍历 $D^{(1)}$, 产生 $D^{(2)}$ 。以下再继续用遍历蒙片 (c)、(d) 分别进行从右到左、从上到下的向前遍历和从左到右、从下到上的向后遍历, 最终获得 $D^{(4)}$ 。作者已证明 $D^{(4)}$ 就是由 N -序列 $q_3 = N_0 N_1$ 所决定的距离图。值得指出的是, 对每个标有 a 的蒙片栅格, 在进行遍历时, 对被它覆盖的栅格不一定都加 1, 只有当这个被覆盖的栅格为奇数值时, 才能加 1。否则就加上无穷大数。这点反映了由 N -序列 $q_3 = N_0 N_1$ 决定的路径, 在某些步上只能按上、下、左、右四个方向进行移动, 而不能在对角线方向上移动。

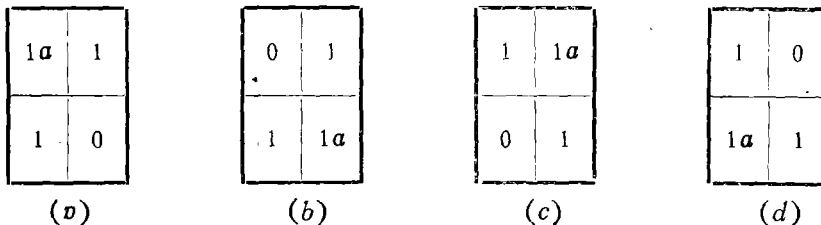


图 2 八边形距离图的遍历蒙片

欧氏距离与八边形距离的最大误差值为 $0.118 M$ 。可见, 八边形距离的误差处于 N_0 -邻居距离和 N_1 -邻居距离之间。

5. 各种距离图距离的比较

下表列出了几种典型距离的误差和相应距离图的计算量。从表中可知, 对角线 (3, 4) 距离误差较小, 但此时距离图中的距离值都相应被扩大了 3 倍, 对较大的图象其距离值将变得很大, 而处理较大的整数有时是不现实的。如果相邻栅格间距离只能取 1, 则八边形距离误差较小, 但需要 4 次遍历。求值最快的距离类型是 N_0 -邻居距离, 但误

差较大。因此，在具体的应用领域中，应根据不同的要求，采用不同的距离类型，使用不同的距离图。

二维栅格图象各种典型距离比较

距离类型	运算量 (求和次数)	遍历次数	最大误差
N_0 -邻居距离	$M \times 4$	2	-0.586M
N_1 -邻居距离	$M \times 8$	2	0.414M
对角线距离	$M \times 8$	2	-0.090M
最优对角线距离	$M \times 8$	2	$\pm 0.064M$
对角线 (3, 4) 距离	$M \times 8$	2	0.081M
对角线 (2, 3) 距离	$M \times 8$	2	-0.314M
八边形距离	$M \times 16$	4	0.118M

参 考 文 献

1 G Borgefors. Distance transformations in arbitrary dimensions. *Comp. Vision, Graphics and image processing*, 1984; (27) : 321~345
2 P E Danielsson. Euclidean distance mapping. *Comp. GIP*, 1980; (14) : 227~248
3 A Rosenfeld and J I Pfaltz. Distance functions on digital pictures. *Patt. Reco.*, 1968; (1) : 33~61
4 M Yamashita. Distance functions defined by variable neighborhood sequences. *Patt. Reco.*, 1983; (5) : 509~513

(编辑 李耀莉)

The Calculation and Analysis for the Distance
Graph about 2-Dimensional Raster Image

Li jianping Shi jimin

ABSTRACT

This paper focuses on various algorithms for the distance graphs about 2-D raster image and their amount of calculation, the number of pass and difference from Euclidean distance. First, some definitions such as distance and distance graph etc., are given by the present writer. It is then proposed that algorithm A be used to calculate the N-neighborhood's or chamfer's distance graphs and algorithm B be used to calculate the octagon's distance graphs. And at the same time two groups of masks are utilized to describe these algorithms. At last, the amount of calculation, the number of pass and the difference between the Euclidean distance and the computed distances for some distance graphs are compared.

Key Words: thinning, distance graph, raster image, path, neighborhood, N-Sequence