

一种顾及障碍物的欧氏距离变换方法*

张青年

(中山大学地理科学与规划学院, 广东 广州 510275)

摘要: 提出了一种顾及障碍物的欧氏距离变换方法。该方法基于栅格扫描方式进行距离变换, 逐行逐列传播距离值, 并在距离传播过程中进行可见性检测, 从而使距离传递路径避开障碍物, 得到一种反映实际通行距离的欧氏距离图像。该算法思路简单直接, 不涉及任何复杂数据结构, 具有线性时间复杂度。实验表明, 该算法计算出的距离图像准确性高, 优于 ArcGIS 基于距离累计的算法, 适合于点、线、面 3 种形态的源和障碍物的欧氏距离变换。

关键词: 欧氏距离变换; 障碍物; 可见性

中图分类号: P208 **文献标志码:** A **文章编号:** 0529-6579 (2013) 01-0130-06

Euclidean Distance Transform by Raster Scan in Obstacle Space

ZHANG Qingnian

(School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: An algorithm of Euclidean distance transform was proposed by raster scan in obstacle space. It propagates distances line by line, checking the visibility of target pixels, and calculating the distance by finding a shortest way from a source to the target keeping away from obstacles. The algorithm is straightforward for distance calculation, involving in no complicated data structure, and having a linear computational complexity. Case study showed that the calculated distances by this algorithm were accurate, better than the results by the distance accumulation-based algorithm, and suitable for sources and obstacles in any types of shapes, including point, line and area objects.

Key words: Euclidean distance transform; obstacle; visibility

距离变换是一种二值图像处理技术, 被广泛应用于图像处理和模式识别等领域。它将二值图像中的像素区分为特征(源)和背景两种类型, 计算背景像素的距离值而产生一幅距离图像^[1]。在距离变换相关研究中提出了大量的距离变换算法, 这些算法主要集中在提高距离变换的精确性、简单性和时间效率等方面^[2-13]。这些算法通常都没有考虑障碍物的影响, 其距离变换是在一个没有障碍物的空间中进行的。但在实际的地理空间中, 通常有河流、山丘等地物存在, 对距离的传递起到阻隔作用, 即障碍物两侧的两个点之间的通行距离并不是直线距离。因此, 考虑障碍物影响的距离变换算

法才可能得到实际通行距离, 该类算法的应用领域包括可视域分析、目标分割、距离/厚度测量、路径规划、影响区域范围等^[13]。目前只有少数学者研究了顾及障碍物的距离变换算法^[13-17], 而且他们的算法也较复杂, 都采用了由源向外逐个圈层传递距离的方式和较复杂的像元可见性判断方法。文献[14-15]最早在距离变换中考虑障碍物, 采取逐圈层传递距离方式。Coerjolly 等^[16]提出了一种基于可见性分析的距离变换方法, 基于角度排序判断像元可视性。Cárdenes 等^[13]提出了一种基于阻隔点(Occlusion points)探测的距离变换算法, 依据阻隔点相对于障碍物的最大最小遮蔽角判断后

* 收稿日期: 2012-07-14

基金项目: 国家自然科学基金资助项目(40971210)

作者简介: 张青年(1968年生), 男, 博士, 副教授; E-mail: zqzsu@163.com

续距离传递过程中像元的可见性。这些算法复杂性高，软件实现困难。除此之外，ArcGIS 软件也实现了一种距离变换算法，通过累计距离传递路径上各个像元中心点之间距离的和得到距离值。这种算法原理简单，但若源像元与目标像元不在同一行、同一列或同一条对角线上，其累计距离值与二者间的欧氏距离有一定偏差，并且误差随传递距离的增加而增大。本文从提高算法简单性角度出发，提出了一种顾及障碍物的距离变换算法，基于栅格扫描方式传递距离，并采用改良的直线栅格化方式判断像元可见性，算法简单，具有线性时间复杂度，所得到的距离准确度较高。

1 基于栅格扫描的欧氏距离变换原理

以 8 邻居距离变换为例，基于栅格扫描的欧氏距离变换基本过程是^[2]：首先自上而下扫描栅格平面，分别依据左、左上、上和右上侧邻居像元的距离值计算目标像元的距离值，用 4 个新距离值和目标像元的当前距离值中的最小值更新目标像元的距离值；然后自下而上扫描栅格平面，依据右、右下、下和左下侧邻居像元的距离值计算目标像元的距离值，用 4 个新距离值和目标像元的当前距离值中的最小值更新目标像元的距离值。

在距离变换过程中，可以采取累计距离的方式计算目标像元到源的距离（图 1），可表示为：

$$d_{ij} = d_{uv} + \Delta d$$

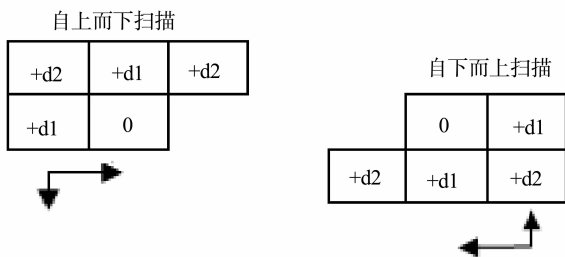


图 1 栅格扫描与距离增量

Fig. 1 Distance offset in raster scan algorithm

其中， d_{ij} 为目标像元 P_{ij} 的距离， d_{uv} 为邻居像元 P_{uv} 的距离， Δd 为 P_{ij} 与 P_{uv} 之间的距离。目标像元与其上、下、左、右 4 个邻居像元之间的距离为 1，与左上、右上、左下、右下 4 个邻居像元之间的距离为 $\sqrt{2}$ 。

但这种累计距离计算方式只在源的上、下、左、右及左上、右上、左下、右下 8 个方向上的距离传递路径是直线，得到准确的直线距离；在其他方向上的距离传递路径都是一条从源到目标像元的

折线，该折线长度比二者之间的直线距离大。设某位置到源的行距和列距的最大值为 k ，其最大误差方向的误差为 $0.089k^{[17]}$ 。

在计算空间中存在障碍物的情况下，只需限定障碍物像元不参与距离比较，该算法仍可有效地进行距离变换。但随着离源的距离增加，其累积误差不断增大。这种算法被 ArcGIS 用于计算成本距离，适用于小幅图像的距离计算。

更准确距离计算方法是依据偏移量计算像元到源的直线距离。在距离变换过程中，记录各像元到源的行偏移量和列偏移量，可以视为一个向量，依据距离向量计算直线距离。目标像元到源的偏移量依据其邻居像元的偏移量来计算，即

$$a_{ij} = a_{uv} + \overrightarrow{P_{ij}P_{uv}}$$

其中， a_{ij} 为目标像元 P_{ij} 的距离向量， a_{uv} 为邻居像元 P_{uv} 的距离向量， $\overrightarrow{P_{ij}P_{uv}}$ 为从 P_{ij} 到 P_{uv} 的距离向量。如图 2 所示，目标像元到左、左上、上和右上侧邻居像元的距离向量分别为 $(1, 0)$ 、 $(1, 1)$ 、 $(0, 1)$ 和 $(-1, 1)$ ，目标像元到右、右下、下和左下侧邻居像元的距离向量分别为 $(-1, 0)$ 、 $(-1, -1)$ 、 $(0, -1)$ 和 $(1, -1)$ 。采用距离向量方式能够得到准确的欧氏距离。

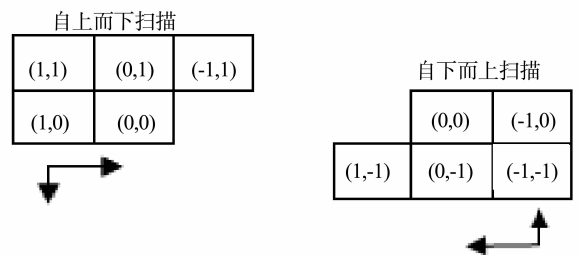


图 2 栅格扫描与距离矢量

Fig. 2 Distance vector in raster scan algorithm

采用偏移量计算距离的方式不适合于存在障碍物的情形。在目标像元到源的连线上有障碍物时，源到目标像元的距离传递路径应为一條绕过障碍物的折线，此时直线距离所代表的直线传递路径显然不符合实际情况。

2 顾及障碍物的欧氏距离变换方法

2.1 算法框架

如前所述，在目标像元到特征像元（源）的连线上存在障碍物时，目标像元被障碍物阻隔，源到目标像元的距离传递路径不再是一条直线。本文在基于距离矢量的栅格扫描算法中引入“虚拟

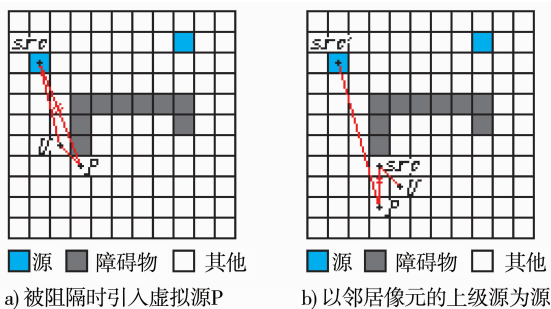
源”，允许距离传递路径发生转折，在被障碍物阻隔的区域中继续进行距离传递。虚拟源是被障碍物阻隔的一个非障碍物像元。为区别起见，将原有特征像元称为初始源。

算法的主要过程如下：

1) 初始化。假定栅格平面大小为 $M \times N$ ，设置 4 个二维数组 $e[M, N]$ ， $a[M, N]$ ， $b[M, N]$ 和 $s[M, N]$ 。 $e_{i,j}$ 记录像元 $P_{i,j}$ 是否以自身为源， a_{ij} 和 b_{ij} 分别记录像元 $P_{i,j}$ 的源（初始源或虚拟源）的列坐标和行坐标， $s_{i,j}$ 记录像元 $P_{i,j}$ 到初始源的距离， $1 \leq i \leq M$ ， $1 \leq j \leq N$ 。如果像元 $P_{i,j}$ 是初始源，令 $a_{ij} = i$ ， $b_{ij} = j$ ， $e_{i,j} = 1$ ， $s_{i,j} = 0$ ；否则令 $e_{i,j} = 0$ ， $s_{i,j} = 2 \times \max(M, N)$ 。

2) 以栅格平面的左上角为起点，行号递增，列号递增，自上而下顺序访问各像元 $P_{i,j}$ 。若 $P_{i,j}$ 不是障碍物像元，则依据其左、左上、上和右上侧邻居像元依次更新其源坐标和距离 a_{ij} 、 b_{ij} 、 $e_{i,j}$ 、 $s_{i,j}$ 。不失一般性，记邻居像元为 U 。

首先判断 U 是否为障碍物像元，若是则直接跳过，不参与距离比较和更新。



源，更新 e_{ij} ， a_{ij} ， b_{ij} ， s_{ij} 。令 $e_{ij} = 0$ ， $a_{ij} = src_x(U)$ ， $b_{ij} = src_y(U)$ ， $s_{ij} = s_{new}$ 。

若 $src(U)$ 不能直达 P (图 3a)，则计算经过 U 到达 P 的距离 s_{new} 。令 $s_{new} = s(U) + \Delta d$ 。其中， Δd 为 P 与 U 之间的直线距离。若 s_{new} 小于 P 的距离值 s_{ij} ，则以 $src(U)$ 为上级源，更新 e_{ij} ， a_{ij} ， b_{ij} ， s_{ij} 。令 $e_{ij} = 1$ ， $a_{ij} = i$ ， $b_{ij} = j$ ， $s_{ij} = s_{new}$ 。即引入 P 为虚拟源。

3) 以栅格平面的右下角为起点，行号递减，列号递减，自下而上顺序访问各像元 $P_{i,j}$ 。若 $P_{i,j}$ 不是障碍物像元，则依据其右、右下、下和左下侧邻居像元依次更新 $e_{i,j}$ 、它的源的坐标 a_{ij} 和 b_{ij} 及距离 $s_{i,j}$ 。

4) 重复执行步骤 2) 和 3) 若干次，直到距离没有变化为止。

2.2 可见性判断

上述算法的一个核心过程就是可见性判断，即源到障碍物之间是否有障碍物。Coeurjolly et al. [16] 基于障碍物的角度排序进行可见性测试；Cárdenes et al. [13] 则记录每个 LNHP 对应的障碍物的遮蔽角范围并将落在遮蔽角范围内的像元判定为不可见。本文基于直线栅格化算法直接判断目标的可见性。

直线栅格化有八方向栅格化、恒密度栅格化、全路径栅格化等多种算法。其中八方向栅格化得到的直线最细，每行或每列只有 1 个像元被涂黑；恒密度栅格化得到的直线粗细均匀，相邻行或列之间有程度相似的重复；全路径栅格化则将直线经过的每个像元都涂黑。

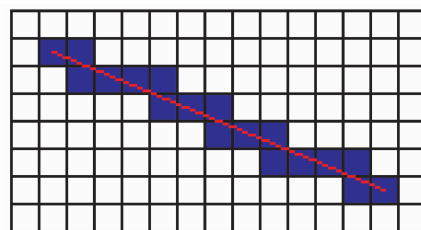


图 4 全路径栅格化

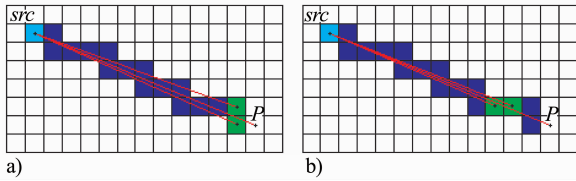
Fig. 4 Full path rasterization

图 3 依据邻居像元 U 更新目标像元 P 的距离
Fig. 3 Update distance of P according to neighbor U

然后判断 U 的源 $src(U)$ 的上级源 $src(src(U))$ 是否可直达目标像元 P (图 3b)。如果 P 到 $src(src(U))$ 的连线上没有任何障碍物像元，则 $src(src(U))$ 可直达 P ，依据它们的行列坐标计算二者之间的直线距离 s_{new} 。若 s_{new} 小于 P 的距离值 s_{ij} ，则以 $src(src(U))$ 为源，更新 e_{ij} ， a_{ij} ， b_{ij} ， s_{ij} 。令 $e_{ij} = 0$ ， $a_{ij} = src_x(src(U))$ ， $b_{ij} = src_y(src(U))$ ， $s_{ij} = s_{new}$ 。其中 $src_x(src(U))$ 和 $src_y(src(U))$ 分别表示 $src(src(U))$ 的列坐标和行坐标。

若上级源 $src(src(U))$ 不能直达 P ，则判断 $src(U)$ 是否可直达 P (图 3a)。若 $src(U)$ 可直达 P ，依据它们的行列坐标计算二者之间的直线距离 s_{new} 。若 s_{new} 小于 P 的距离值 s_{ij} ，则以 $src(U)$ 为

显然，可以依据全路径栅格化算法判定目标像元的可见性。即，在全路径栅格化过程中，从源开始检测位于路径上的每一个像元，只要有一个像元是障碍物像元，则目标像元不可见；否则继续检测下一个像元，如果路径上的所有像元都不是障碍物像元，则目标像元位于源的可见区域内。

图 5 确定像元 P 的可见性Fig. 5 Check the visibility of pixel P

本文基于全路径栅格化原理提出一种高效的可见性判断算法，不需要检测直线经过的每一个像元是否为障碍物。其算法原理是从目标像元 P 开始向源 src 反向栅格化，判断最初两行（或列）像元的可见性即可。设该直线的跨度为 m 行 n 列，如果 $m < n$ ，逐列检测像元是否需要涂黑，否则逐行检测像元是否需要涂黑。如果需要涂黑的多个像元的源就是 src ，则可直接判定目标像元 P 在 src 的可见域内，而不必遍历直线上的所有像元并检测其是否为障碍物。以图 5 为例，从 P 到 src 的连线跨 6 行 13 列，从 P 开始对连线上的像元进行检测，每个列最多检测 2 个像元。具体算法过程如下：

1) 如果连线上某个列有 2 个像元并且它们的源都是 src ，则 P 在 src 的可见域内（如图 5a 所示），退出检测过程；

2) 如果连线上某个列只有 1 个像元并且它的源是 src ，继续检测连线上的下一列的所有像元的源是否为 src ，如果下一列全部像元的源都是 src ，则 P 在 src 的可见域内（如图 5b 所示），退出检测过程；

3) 如果连线上遇到一个障碍物像元，则 P 不在 src 的可见域内，退出检测过程；

4) 其他情况下继续向前进行检测。

可以预见，如果 P 在 src 的可见域内，其连线上靠近 P 处一定有源为 src 的像元，因此检测过程很快就会结束。即使连线长、经过的像元多，也只需检测靠近 P 的数个像元。

2.3 算法效率分析

分析上述算法框架可以发现，其基本过程是栅格扫描，每个像元被扫描 2 次，每次与 4 个邻居像元对应的距离值进行比较。设像元个数为 n ，则距离比较和赋值的次数为 $8n$ 。参与比较的距离值是依据邻居像元计算出来的，在距离计算过程中要执行可见性判断。本文采取直线栅格化算法判定可见性，它是一个双重循环算法，但实际上并不需要检测连线上的所有像元是否为障碍物，而只需检测靠近目标像元一端的有限个像元，因此该可见性判断

算法是常量阶的。另外，在目标像元被障碍物遮挡层次较深的情况下，扫描过程需要重复多次才能完成距离传播，但重复的次数可以视为一个常数。因此，该距离变换算法总体上的时间复杂度为 $O(n)$ 。

3 实验结果及分析

为了验证上述方法的可行性，本文构造了一幅 255 行 318 列的图像进行了距离变换实验，计算结果见图 6a。图 6a 中有 3 个源，其中 O_1 和 O_2 为点状源， O_3 为面状源；有 3 个面状障碍物 B_1 、 B_2 和 B_3 。从图 6a 可以看出，从源向外，距离逐渐增大，距离等值线之间的间距相等（5 个栅格距离）。点源周围的距离等值线浑圆，呈现为同心圆结构；面源周围的等值线也是一种圈层结构，其延伸趋势与面源轮廓线一致。距离等值线圈层结构在遇到障碍物 B_1 和 B_2 之后遭到破坏，在障碍物背后继续进行距离传播，并在障碍物端点处形成新的圈层结构（例如在 B_1 背后靠近两端处）。在遇到第二重障碍物 B_3 之后，圈层结构再次被破坏，在 B_3 背后靠近两端处形成新的圈层结构。

由于在障碍空间中逐像元计算距离真值难度很大，无法将本文算法计算结果与距离真值进行对比。这里改用与 ArcGIS 软件计算结果比较的方式进行验证。目前顾及障碍物的距离变换算法很少且算法复杂，在实际工作中通常利用 ArcGIS 软件进行距离变换，因此 ArcGIS 软件计算结果具有一定代表性。从前述分析可知，ArcGIS 累计距离算法中的距离传递路径和本文算法中的距离传递路径都绕过了障碍物，计算出的距离值都大于实际距离值。因此，计算出的距离较小的算法更有效（其结果更接近距离真值）。将 ArcGIS 计算的距离与本文算法计算的距离相减，得到距离差值图，见图 6b。可以发现，本文算法计算的距离值更小。据第 1 节欧氏距离变换原理分析可知，累计距离在源周围的上、下、左、右、左上、左下、右上和右下共 8 个方向上是没有误差的，而在其他方向上都有距离误差，而且在这 8 个方向中的相邻两个方向的角平分线上误差最大。在图 6b 中，两种计算结果在点状源 O_1 和 O_2 周围的“米”字形方向上距离差为 0，在面状源 O_3 周围的叠加“米”字形方向上距离差也为 0，表明本文算法在源周围的 8 个方向上没有误差。在源周围的“米”字形图案的相邻两条分岔的角平分线方向上，ArcGIS 计算的距离值相对于本文算法的距离值增加最多，这与 ArcGIS

距离累计算法的误差在各方向上的分布一致, 表明本文算法在 8 方向之外其他方向上的距离值也接近于真值。在障碍物背后, 距离值最小的像元作为新的“源”继续传递距离, 两种算法的距离差在障碍物背后形成新的不完整的“米”字形图案。由于两种算法在新的“源”位置上存在距离差,

“源”周围的 8 个方向上也存在与“源”位置上相等的距离差, 其他方向上的距离差更大。此外, 两种算法计算结果的距离差随着与源的距离的增加而变大, 两种算法的最大距离差为 15.20。其原因是, ArcGIS 距离累计算法的误差与到源的距离正相关, 而本文算法的误差与空间距离无关。

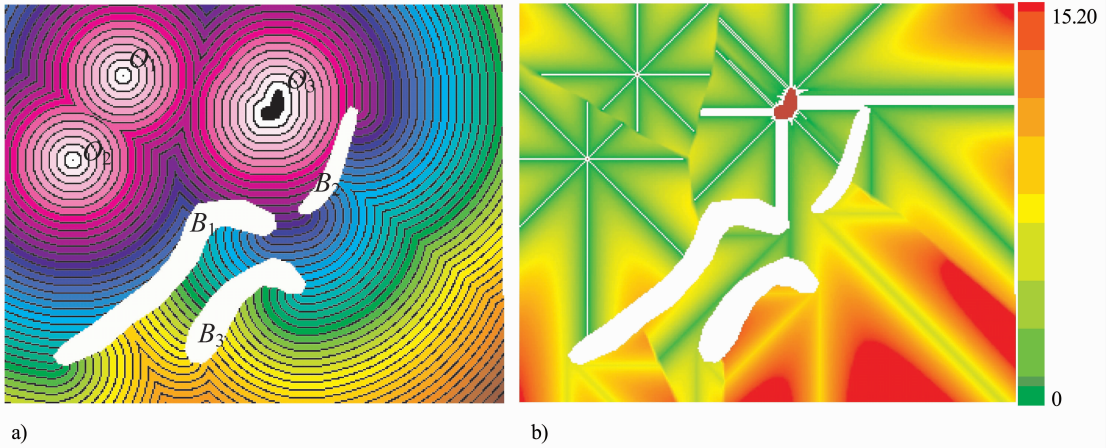


图 6 基于栅格扫描的欧氏距离变换实验

Fig. 6 Case study of raster scan based euclidean distance transformation

虽然 ArcGIS 计算结果不是距离真值图, 无法分析本文算法计算结果的最大误差, 但我们可以依据算法原理对最大误差作一个初步分析, 见图 7。依据本文算法, 源 s 在传递距离到 p 时遇到障碍物, 经过虚拟源 s' 之后继续传递距离到 q 。这条路径上, 障碍物前方的 sp 段和后方的 $s'q$ 段为直线距离, 都没有误差。但绕过障碍物的 $ps'q$ 段没有紧贴障碍物的边沿 uw (uw 为障碍物底边的两个角), 有一定的误差。这里分 spu 和 $us'q$ 两段来分析。当 s 离 p 很远时, 折线 spu 与直线 su 几乎重合, 其长度差可忽略不计; 反之, 则有一定差距, 当 s 和 p 为对角线相邻时两者的长度差最大, 为 $1 + \sqrt{0.5} - \sqrt{2.5} = 0.126$ 。类似地, 如果 $s'q$ 位于 uw 的同一侧 (例如目标像元为 q'), 则 $us'q'$ 的长度小于 uwq' , 应取 $us'q'$ 长度作为真距离, 此时 $us'q'$ 没有误差; 反之, 当 s' 和 q 位于 uw 的两侧时, $us'q$ 的长度大于 uwq , 存在一定误差, 并且 q 与 s' 对角线相邻时, 误差达到最大, 为 $2\sqrt{0.5} - 1 = 0.414$ 。因此, 障碍物两侧的最大误差和为 0.540。将障碍物遮挡的区域称为阴影区, 阴影区内最大误差为 0.540, 非阴影区内无误差。若距离传递路径经过 n 个障碍物, 第 n 个障碍物背后的阴影区内的最大误差不超过 $0.540n$ 。

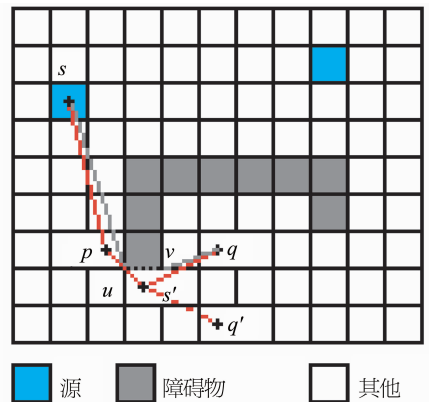


图 7 误差分析

Fig. 7 Error analysis

4 结 论

通过以上分析和实验, 可以得出以下结论:

- 1) 在距离传播过程中进行可见性检测, 并在目标不可见时引入“虚拟源”, 确保了在障碍物背后继续进行有效的距离传播, 所得距离是从初始源出发绕过障碍物的折线距离。
- 2) 在可见性检测确定上级源可见的情况下, 直接以当前的源的上级源为中继向外进行距离传播, 从而避免了传播路径经过当前的源而发生不必

要的转折, 使所得距离值更接近于最短可通行距离。

3) 本文算法的距离准确性较高, 优于 ArcGIS 距离累计算法, 在距离传递路径经过的第 n 个障碍物背后的阴影区内最大误差为 $0.540n$, 非阴影区内无误差。

4) 本文算法简单, 不涉及桶排序等任何复杂的数据结构, 易于理解和实现, 适合于点、线、面 3 种形态的源和障碍物的距离变换。

5) 本文算法的时间复杂度为 $O(n)$, 计算效率高。

参考文献:

- [1] ROSENFELD A, PFALTZ J. Sequential operations in digital pictures processing [J]. Journal of the ACM, 1966, 13(4): 471 - 494.
- [2] BORGEFORS G. Distance transformations in digital images [J]. Computer Vision, Graphics and Image Processing, 1986, 34: 344 - 371.
- [3] PAGLIERONI D W. A unified distance transform algorithm and architecture [J]. Machine Vision and Applications, 1992, 5(1): 47 - 55.
- [4] SAITO T, TORIWAKI J. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications [J]. Pattern Recognition, 1994, 27: 1551 - 1565.
- [5] FABBRI R, COSTA L D F, TORELLI J C, et al. 2D Euclidean distance transform algorithms: A comparative survey [J]. ACM Computing Surveys, 2008, 40(1): 1 - 44.
- [6] 陈峻. 完全欧几里德距离变换的最优算法 [J]. 计算机学报, 1995, 18(8): 611 - 616.
- [7] 王征旋, 李文辉, 庞云阶. 基于围线追踪的完全欧氏距离变换算法 [J]. 计算机学报, 1998, 21(3): 217 - 222.
- [8] 任勇勇, 潘泉, 张绍武, 等. 基于围线分层扫描的完全欧氏距离变换算法 [J]. 中国图象图形学报, 2011, 16(1): 32 - 36.
- [9] LUCET Y. New sequential exact Euclidean distance transform algorithms based on convex analysis [J]. Image and Vision Computing, 2009, 27(2): 37 - 44.
- [10] 徐达丽, 任洪娥, 徐海涛, 等. 基于链码技术的距离变换改进算法 [J]. 计算机工程与应用, 2009, 45(25): 176 - 178.
- [11] 陆宗骥, 朱煜. 用带形状校正的腐蚀膨胀实现 Euclidean 距离变换 [J]. 中国图象图形学报, 2010, 15(2): 294 - 300.
- [12] GUSTAVSON S, STRAND R. Anti-aliased Euclidean distance transform [J]. Pattern Recognition Letters, 2011, 32: 252 - 257.
- [13] C RDENES R, ALBEROLA-L PEZ C, RUIZ-ALZOLA J. Fast and accurate geodesic distance transform by ordered propagation [J]. Image and Vision Computing, 2010, 28: 307 - 316.
- [14] LANTUEJOL C, MAISONNEUVE F. Geodesic methods in quantitative image analysis [J]. Pattern Recognition, 1984, 17: 177 - 187.
- [15] PIPER J, GRANUM E. Computing distance transformations in convex and nonconvex domains [J]. Pattern Recognition, 1987, 20(6): 599 - 615.
- [16] COEURJOLLY D, MIGUET S, TOUGNE L. 2D and 3D visibility in discrete geometry: an application to discrete geodesic paths [J]. Pattern Recognition Letters, 2004, 25: 561 - 570.
- [17] 胡鹏, 游链, 杨传勇, 等. 地图代数 [M]. 2 版. 武汉: 武汉大学出版社, 2006.