

流域变换建模及其算法研究的新进展

周海芳 蒋艳凰 杨学军

(国防科技大学计算机学院, 长沙 410073)

摘要 流域变换是数学形态学中用于图象分割的一种经典方法, 虽然流域变换曾因运算量大、效率低而使得其研究工作遭到冷遇, 但也因此出现了一些新的理论和算法, 并随着并行手段的引入, 又使其重新成为研究的热点; 同时就近期许多研究成果而言, 形式化模型的多样性, 使得流域变换的定义、算法和实现, 尚缺乏统一的描述和全面的总结. 针对这一情况, 首先给出了连续域流域变换的严格数学模型和两种离散情况下典型的形式化定义; 然后分类总结了近年来, 流域变换算法实现的新进展; 最后提出了有待进一步研究的问题.

关键词 流域变换 图象分割 集水盆 分水岭 浸没 地形学距离

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2003)01-0008-07

Modeling for Watershed Transform and Development of Watershed Algorithm

ZHOU Hai-fang, JIANG Yan-huang, YANG Xue-jun

(Department of Computer Science and Technology National University of Defense Technology, Changsha 410073)

Abstract Watershed transform is a classical method of image segmentation in mathematical morphology. This method, with a wide perspective, has been applied successively into some fields like remote sensing images processing of satellite and radar, biomedical and computer vision applications. Watershed transform is a relatively time consuming task, and the study of this problem has been disesteemed by researchers for its low efficiency. But also for this reason, there appear some new theories and algorithms; moreover, with the development of parallel tools, watershed transform has an increasing attention internationally. As to the recent literatures on this field, variety of formalizing model makes the definition, algorithm and implementation of watershed lack in uniform description. Pointing to this status this paper starts with a rigorous definition of watershed transform for continuous case, followed by two kinds of representative definitions for digital case. The solution of plateau problem is discussed in detail. Then some new watershed algorithms proposed in recent years are classified and analyzed, including watershed algorithms by immersion and by topographical distance. The need to distinguish between definition, algorithm specification and algorithm implementation is pointed out. Finally we present the problems and challenges of future research, including accuracy of watershed transform, problem of over-segmentation, parallelization of algorithms and so on.

Keywords Watershed transform, Image segmentation, Catchment basins, Watershed line, Immersion, Topographical distance

0 引言

流域变换(watershed transform, 又称为分水岭变换)是数学形态学中, 用于图象分割的一种经典方

法. 它最初是由 Digabel 和 Lantuejoul 于 20 世纪 70 年代末引入图象处理领域, 用于分析简单的二值图象^[1]. 为了得到更为通用的模型, Beucher、Vincent 等人的研究工作^[2,3], 使流域变换的理论得以建立, 并在 20 世纪 80 年代大量用于灰度图象的分割问

题.最近十年中,流域变换曾因运算量大、效率低而遭冷遇,但也因此出现了一些新的理论和算法,同时由于并行手段的引入,使其重新焕发了生机,在工业、生物医学、计算机视觉、遥感等应用领域中,将可能发挥更大的作用.

但是目前许多针对流域算法研究的文章,由于形式化模型的多样性,使人们常常很难确定算法采用的是哪一种定义.有时定义采取的仅是算法描述的形式,而没有仔细区分描述和实现之间的差别,如,当给定一个描述后,其相应的实现不能与其相符时,或者在对定义进行修改,以消除不期望的结果时,这种修改往往又引入新的问题;或者虽引入优化(比如加快速度、减少内存消耗),但同时它也改变了算法的输出结果,即使这种变化在自然图象中经常不易被察觉.这些问题已不是纯粹的学术问题,因为流域算法正广泛地应用于医学、军事等图象处理领域,而且这些领域都要求尽量避免算法的副作用.

为此,本文一方面给出了连续域流域变换的严格数学模型,并归纳列举了几种离散情况下流域变换的形式化定义;另一方面,分类总结了近年来流域变换算法研究的新进展,同时分析强调了算法定义、算法描述和算法实现之间的差别,以便为今后的研究工作提供参考,并为该算法进一步的完善乃至突破性的进展提供一定的帮助.

1 流域变换建模

流域变换可以归属为基于区域的分割方法.这种方法的思想来源于地理学,即将图象看作是地形学上被水覆盖的自然地貌,图象中的每一像素的灰度值表示该点的海拔高度,其每一个局部极小值(local minima)及其影响区域称为集水盆(catchment basins),而集水盆的边界则是分水岭.通常描述流域变换有如下两种方法:第1种是“降雨法”,即当一滴雨水分别从地形表面的不同位置开始下滑,其最终将流向不同的局部海拔高度最低的区域(称为极小区),那些汇聚到同一个极小区的雨滴轨迹就形成一个连通区域,称为集水盆,对于地形上凸起的平坦高地,雨滴在其上的运动,通常认为是顺着距离高地的下降边沿最近的路径流动;第2种方法是模拟“泉涌浸没”(immersion)的过程(又称为爬山法),这是一种自下而上形成区域的方法,即首先在各极小区的表面打一个小孔,同时让泉水从小

孔中涌出,并慢慢淹没极小区周围的区域,那么各极小区波及的范围,即是相应的集水盆.这实质上是两种不同的泛洪方法,不论是降雨法,还是爬山法,不同区域的水流相遇时的界限,即是期望得到的分水岭.将其应用到图象分割中,流域变换就是指将源图象转换为一个标记图象,其中所有属于同一集水盆的点均被赋予同一个标记,同时用一个特殊的标记来标识分水岭上的点.下面给出的流域变换形式化定义,它可以看成是测地影响区骨架SKIZ^[1]的定义在灰度图象中的一般化.下面先考虑连续域的情况,然后给出两种离散情况下较典型的流域变换定义.

1.1 连续域流域变换定义

连续域中严格的流域变换定义,一般是基于距离函数的图象积分,但不同的距离函数可以得到不同的定义,这里仅讨论其中的一种.

设 $f(p)$ 为二维灰度图象函数, $F(D)$ 为连续域 $D(D \subseteq \mathbb{R}^2)$ 上的实二次连续可微的函数空间, $f \in F(D)$, 则点 $p, q \in D$ 之间的地形学距离定义为:

$$L_f(p, q) = \inf_{\gamma} \int_{\gamma} \|\nabla f(\gamma(s))\| ds$$
 其中, γ 表示域 D 中, p, q 之间所有光滑路径的集合, \inf 表示路径长度的下界. 而点 $p \in D$ 与集合 $A \subseteq D$ 之间的地形学距离定义为: $L_f(p, A) = \min_{a \in A} L_f(p, a)$. 可见, p, q 之间使 L_f 距离最短的路径是 p, q 间坡度最陡的路径.

定义1 连续域流域变换定义

设 $f \in F(D)$ 在定义域内存在极小区集合 $\{m_i\}_{i \in I}$, I 为顺序索引标记集合. 某一极小点 m_i 所代表的极小区最终形成的集水盆 $B(m_i)$ 定义为, 对于所有的 $p \in D$, p 到以极小点 m_i 代表的极小区的距离比到其他任何以 m_j 代表的极小区的距离更短

$$B(m_i) = \{p \in D \mid \forall j \in I, j \neq i: L_f(p, m_i) < L_f(p, m_j)\} \quad (1)$$

变换形成的分水岭是不属于任何集水盆的点的集合

$$W(f) = D \cap \left\{ \bigcup_{i \in I} B(m_i) \right\}^c \quad (2)$$

式(1)、式(2)在原理上体现了上述“降雨积水”的思想.

设 w_0 为一特殊标记, 且 $w_0 \notin I$, 其代表分水岭点, 那么灰度图象 f 的流域变换, 实质上是一个映射 $\lambda: D \rightarrow I \cup \{w_0\}$, 使得若 $p \in B(m_i)$, 则 $\lambda(p) = i$; 否则 $\lambda(p) = w_0$, 即图象 f 的流域变换就是将标记 w_0 赋给 D 中各点, 使得①不同的集水盆打有不同的标记; ②特殊的标记 w_0 赋给图象 f 中分水岭上的点.

1.2 离散域流域变换定义

事实上人们更感兴趣的是数字图象,这时 $D \subseteq \mathbb{Z}^2$, 数字灰度图象 $f: D \rightarrow \mathbb{N}$. 但在数字图象中, 距离函数的变化、高地(具有恒定灰值的区域)的形成及其在图象中可能大面积延伸所引发的问题, 将使得连续域流域变换的定义不能直接扩展到离散域中来, 而且这种非局部性的影响也是流域变换并行化的主要障碍. 为此, Beucher 等人首先提出了模拟“泉涌浸没”的流域算法^[2,5], 但其没有形式化定义, 且效率较低; 尔后由 Vincent 和 Soille 提出的基于浸没模拟的算法级定义^[6]是较为严格的递归定义, 由于它回避了高地问题, 故具有一定的代表性.

1.2.1 基于浸没模拟的算法级定义

设 $D \subseteq \mathbb{Z}^2$, $f: D \rightarrow \mathbb{N}$, 且 h_{\min} 和 h_{\max} 分别为图象 f 中最小和最大灰度值. 递归过程定义就是使灰度级从 h_{\min} 增长到 h_{\max} , 同时与 f 的极小区相关的集水盆也顺序地扩大. 若用 X_h 表示在 h 级计算所得的集水盆集合, 则一个在 $h+1$ 级属于阈值集 T_{h+1} 的连通域, 要么是一个新极小区, 要么是 X_h 中一个集水盆的扩展(其中, $T_h = \{p \in D \mid f(p) \leq h\}$). 对于后者应计算 X_h 在 T_{h+1} 范围内的测地影响区 $IZ^{[4]}$, 并更新为 X_{h+1} . 这里用 M_h 表示在 h 级所有极小区的集合.

定义2 基于浸没模拟的流域变换定义

$$\begin{cases} X_{h_{\min}} = \{p \in D \mid f(p) = h_{\min}\} = T_{h_{\min}} \\ X_{h-1} = M_{h-1} \cup IZ_{T_{h-1}}(X_h) \end{cases} \quad (3)$$

其中, $h \in [h_{\min}, h_{\max}]$.

分水岭则是 $X_{h_{\max}}$ 在域 D 中的补集

$$W(f) = D \setminus X_{h_{\max}} \quad (4)$$

遗憾的是, 文献[6]中最终给出的算法实现, 实际上并不完全符合这个定义, 这一点后面还要讨论.

1.2.2 基于地形学距离的定义

虽然不能将连续域的流域变换定义直接扩展到离散域中来, 但严格的数学定义会带来新的、而且高效的算法, 一方面会使算法更精确; 另一方面在硬件上更容易是实现. Meyer 等人先后提出了一些离散域的流域变换定义^[4], 但由于这些定义都回避了高地问题, 因此由这些定义提出的算法在处理高地区域时的盲目性无法避免. 本文基于上述定义重点讨论了高地问题的解决.

这里, 首先假设数字灰度图象 f 是完全下降的(lower complete), 即不存在高地, 且每个非极小区像素都有至少一个相邻像素比其灰度值更低, 后面

还将讨论如何消除这个假设.

定义像素 p 的下降坡度(lower slope) $S(p)$ 为 p 到其邻域中任何下降点的最大坡度

$$S(p) = \max_{q \in N_G(p) \cup \{p\}} \left\{ \frac{f(p) - f(q)}{d(p, q)} \right\} \quad (5)$$

其中, $N_G(p)$ 表示像素 p 在数字网格图 G 中的邻域集, 该邻域集的范围随邻接方式(4-邻接、8-邻接等)的变化而异. $d(p, q)$ 是边 (p, q) 的长度(实现时常常假设 G 中任何两相邻像素之间的距离是各向同性的, 即均设为 1, 以简化计算). 当 $p=q$ 时, 由于上式的 \max 算子定义为 0, 因此当像素 p 的邻域像素的灰度值都比像素 p 的灰度值高时, $S(p)=0$, 而像素 p 沿路径 τ 到达邻域像素 q 的代价定义为

$$c(p, q) = \begin{cases} S(p) \cdot d(p, q) & \text{当 } f(p) > f(q) \\ S(q) \cdot d(p, q) & \text{当 } f(p) < f(q) \\ (S(p) + S(q)) \cdot d(p, q) / 2 & \text{当 } f(p) = f(q) \end{cases} \quad (6)$$

定义3 离散域地形学距离定义

p, q 之间沿路径 $\tau = (p_0, \dots, p_n)$ (其中, $p_0 = p, p_n = q$) 的地形学距离定义为

$$L_f(p, q) = \sum_{i=0}^{n-1} d(p_i, p_{i+1}) c(p_i, p_{i+1})$$

则 p, q 之间地形学距离定义为: 从 p 至 q 所有路径中最短路径的地形学距离

$$L_f(p, q) = \min_{\tau \in [p, q]} L_f(p, q) \quad (7)$$

其中, $[p, q]$ 表示从 p 至 q 所有路径的集合. 相应的像素 $p \in D$ 与集合 $A \subseteq D$ 之间的地形学距离定义为 $L_f(p, A) = \min_{a \in A} L_f(p, a)$.

容易得到下面的定理:

定理1 设 $f(p) > f(q)$, 则从 p 到 q 路径 τ 是下降最快路径, 当且仅当 $L_f(p, q) = f(p) - f(q)$; 若 τ 不是下降最快路径, 则 $L_f(p, q) > f(p) - f(q)$.

这一定理意味着, 依据上面的定义, 下降最快的路径就是地形学距离最短的路径. 当引入数字图象的地形学距离定义后, 离散域分水岭变换中, 集水盆和分水岭变换的定义即与定义1相同. 同时, 定理1的一个直接结论是: $B(m_i)$ 仅是以 m_i 为代表的极小区上游像素的集合; 而分水岭则包含了所有至少两个极小区上游像素的集合, 也即从一分水岭像素到不同的极小区至少存在两条下降最快的路径. 这与“降水”模型的原理相吻合.

下面着重考虑不是完全下降的图象, 在这种图象中, 存在非极小区的内部非空的高地. 如果直接应用上述的定义, 则高地内部的像素间的地形学距离

均为0,由于这样无法正确判断这些像素之间的关系,这也是相应算法在这种情况下产生盲目性的原因,因此需要在这些像素间,定义另外的顺序关系.其直观的解决方法是计算这些像素距高地下降边沿的测地学距离^[7,8],即将源图象先转换为完全下降的图象后,再应用上面的定义(关于地形学距离和测地学距离之间关系的解释见文献[4]).

定义4 完全下降变换定义

设 f 为域 D 上的数字灰度图象,定义函数 $d(p):D \rightarrow \mathbf{N}$ 如下

$$d(p) = \begin{cases} 0 & \text{如果 } \tau_f^{\downarrow}(p) = \emptyset \\ \min_{t \in \tau_f^{\downarrow}(p)} l(t) & \text{否则} \end{cases} \quad (8)$$

其中, $\tau_f^{\downarrow}(p)$ 表示所有起点为 p , 并终止于 D 中某点 q (满足 $f(p) > f(q)$) 的下降路径的集合, $l(t)$ 表示 t 的测地学路径长度^[14].

令 $\lambda = \max_{p \in D} d(p)$, 则 f 经完全下降变换后的图象 \tilde{f} 定义如下

$$\tilde{f}(p) = \begin{cases} \lambda \cdot f(p) & \text{如果 } d(p) = 0 \\ \lambda \cdot f(p) + d(p) - 1 & \text{否则} \end{cases} \quad (9)$$

函数 $d(p)$ 对极小区像素赋值为0,而对其他像素, $d(p)$ 等于从像素 p 到比像素 p 灰度级低的像素集的最短路径长度,并称 $d(p)$ 为 p 的下降距离.若图象 f 已经是完全下降的,则 $f = \tilde{f}$. 经过完全下降变换后, L_f 即是 $D' \times D'$ 上适用的地形学距离函数(其中, $D' \times D' \setminus \{\text{minima}\}$),它不仅满足定义1的前提,且消除了高地问题.综上所述,在离散情况下,基于地形学距离的流域变换的完整定义如下:

定义5 基于地形学距离的离散域流域变换定义

结合定义1,3,4,设 \tilde{f} 为 f 完全下降变换后的图象, $\{m_i\}_{i \in I}$ 为图象 f 的极小区集合, $B(m_i)$ 为基于 \tilde{f} 的关于某一以 m_i 为代表的极小区的集水盆:

$$B(m_i) = \{p \in D \mid \forall j \in I, \tilde{f}(m_j) + L_{\tilde{f}}(p, m_j) < \tilde{f}(m_i) + L_{\tilde{f}}(p, m_i)\} \quad (10)$$

其分水岭的定义与式(2)相同.

研究中还发现:

(1) 在算法实现中,对含有高地的图象进行流域变换的分割算法,常常没有明确地指出完全下降的步骤,而是用其他方法划分高地内的像素,例如后面将提到的基于顺序队列的算法.需要注意的是,这种算法级的解决方案可能导致产生与定义5相差很

大的结果,其差别的程度与实现的精确度有关.

(2) Meyer 在文献[4]中指出,如果将图象 f 中所有极小区的值均用一最低值替换,那么分水岭将不会改变.这个结论对于定义2是正确的,而且易于证明,但是对于定义5,这个性质事实上并不成立,而且降低一个极小区像素的值,其产生的影响还可能波及到整个图象.

(3) “孤立区”的问题 流域变换的过程中,有些区域可能会完全被分水岭像素包围,按定义,它们也应该标记为分水岭像素,但在基于地形学距离的一些实现中,这些区域可能暂时地或永久地成为孤立区^[9],这也是算法实现与定义不符造成的缺陷.

2 流域变换的算法实现

总的来说,近些年提出的流域变换算法可以按上述的定义大致分为模拟泉涌淹没和基于地形学距离的算法两大类,其中又可根据具体采用的实现技术进一步细分,有些算法还依据是否产生分水岭又分为生成分水岭和不生成分水岭两种.下面对其分别作一下简要的介绍.

2.1 模拟“泉涌淹没”的流域变换算法

2.1.1 Vincent-Soille 算法

Vincent 和 Soille 给出了定义2的一种实现算法^[6].该算法分为如下两步:(1)按灰度递增的次序给像素排序,以便于在某一灰度级直接访问相关的像素;(2)淹没过程:从极小区开始,一级一级地处理.用该算法模拟泉涌时,使用一个 FIFO 队列,按照宽度优先的方式递归地分配给每一极小区及其相关的集水盆以不同的标记.例如递归到灰度级 h ,首先所有灰度值为 h 的像素被赋予标记 MASK,且其中那些已标记邻域像素的像素被插入队列;然后从队列中的这些像素开始,在 MASK 的范围内计算各集水盆的测地影响区 IZ ,如果某像素只与一个集水盆 i 的像素连通,则打上该集水盆的标记 i ;如果一个像素同时与两个集水盆相邻,则标记为分水岭像素 w_0 ;最后那些仍标记为 MASK 的像素,则属于新出现的极小区,被赋予一个新的标记.该算法的复杂度与输入图象的像素个数呈线性关系.

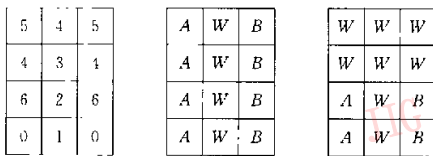
Vincent 和 Soille 虽提出了新的流域变换的定义,而且相应的算法较以前的算法在效率上有很大的提高,但事实上该算法并不能完全实现定义2的递归式,主要差别如下:

(1) 在灰度级 h , V-S 算法只将灰度值为 h 的像素纳入待处理的集合(标记为 MASK), 而定义 2 则是将所有灰度值小于、等于 h 的非集水盆像素均纳入处理集合中。

(2) 定义 2 仅传播集水盆标记, 而 V-S 算法还传播 w_0 标记的像素, 这样做是由差别(1)引起的, 因为当前灰度级不再考虑低灰度值的像素, 所以必须也在当前级确定应标记为 w_0 的像素。

(3) 为了避免 V-S 算法对于当前级在标记某一像素为 w_0 后, 可能会受另一邻域像素的影响而出现偏差, 需将标记 w_0 改为标记 i 。而这种修改在定义 2 中则是在处理更高灰度级时才完成。

这些差别就使得 V-S 算法对某些图象处理的结果与原始定义的结果不一致, 如图 1 的例子。一般认为估价一个算法实现的准确性, 应以它是否符合定义的内容为基础。另外, 虽可通过适当地修改每级递归的初始条件和标记规则来使算法完全符合定义 2, 但修改后的算法复杂度却由线性关系变为二次关系, 其原因是重复地处理分水岭像素所致。可见, 这是准确性与性能之间的一种折衷, 在实际应用中, 应适当地考虑。



(a) 源图 (b) 定义(2)的结果 (c) V-S 算法的结果

图 1 V-S 算法实现与原始定义的差别

2.1.2 区域关系图算法

从并行的角度考虑, V-S 算法的直接并行化是比较困难的, 尤其是处理存在高地的图象时, 其并行效率较低。为此, 另一种基于有向值图的方法被提出^[10], 该方法首先将图象转换为一有向值图, 且图中每个结点都存在与其值不同的相邻结点, 该图称为图象 f 的区域关系图。如果基于此图进行流域变换, 那么 V-S 算法将得到简化, 从而效率也有所提高。由于该方法消除了高地, 因此不再需要 FIFO 队列, 其具体步骤如下:

(1) 首先将输入图象看作一有向值图 $G=(V, E, f)$, 其中, V 表示由所有像素组成的结点集; E 表示由邻接关系决定的结点之间的边集; $f: V \rightarrow N, \forall p \in V, f(p)$ 表示像素 p 的灰度值; 然后将 G 转换成图象 f 的区域关系图 $\tilde{G}=(\tilde{V}, \tilde{E}, \tilde{f})$:

① 若 $v \in \tilde{V}, v$ 则代表源图象中一个灰度级为 h 的连通域, 即 $v = \{p \in V \mid p \in C_h\}$, 其中, C_h 表示灰度值为 h 的连通像素的集合, 且有 $\tilde{f}(v) = h$ 。

② $u, v \in \tilde{V}$, 若 $(u, v) \in \tilde{E}$ 当且仅当 $\exists (p \in u, q \in v: (p, q) \in E \text{ 且 } f(p) > f(q))$ 。

(2) 计算有向值图 \tilde{G} 的流域变换

(3) 先将变换后的图象按步骤(1)的逆过程还原为源图象的形式, 再将分水岭结点值置 1, 其他结点置 0。由此即得到分割后的二值图象; 然后再应用骨架化算法细化该二值图中的分水岭。

2.2 基于地形学距离的流域变换算法

依据定义 3、5, 通过求最短路径来进行流域变换的算法分支较多^[11], 且算法之间存在一些技术交叉。这里按照这类算法一般要处理的 3 个步骤, 将它们采用的技术和方法归纳分类如图 2 所示。

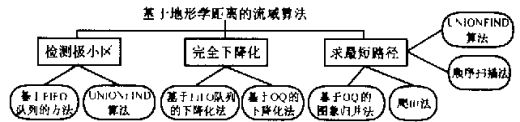


图 2 基于地形学距离的流域算法分类

2.2.1 极小区检测

极小区检测是这类算法必须经历的初始步骤, 实质上这一步是确定各个灰度级的连通域, 再在连通域中确定极小区。这一步通常采用基于 FIFO 队列的泛洪算法来实现^[12]; 另外, UNIONFIND 算法也可以计算连通域, 而事实上 UNIONFIND 算法比 FIFO 泛洪算法效率高, 而且修改后的 UNIONFIND 算法还可直接用于流域变换计算。

(1) FIFO 泛洪算法 标准的、宽度优先的泛洪算法通常使用一个 FIFO 队列来确定具有某一相同灰度值的像素连通, 而对每一个连通域则是先将一个像素放入一个空的 FIFO 队列中, 接着进行泛洪过程, 包括从队列中移出一个像素, 并打上标记; 然后插入虽具有相同灰度值、但还未被标记的邻域像素, 重复此过程, 直到队列为空为止。该算法的复杂度与图象中像素的数量呈线性关系, 而实际检测中, 还需通过一步后处理来判定哪些连通域是极小区。虽然无法再构造一个复杂度更低的算法, 但由于无法预知队列的最小长度, 所以该算法对内存的使用很不规则, 进而导致虚存利用率及并行性能的下降。

(2) UNIONFIND 算法 在该算法中, 不相交集(即各个连通域)以树的形式存储, 而所有集合

组成森林,但可以将每棵树的根作为该集合的代表。该算法中 FIND 操作是指找出与当前结点像素连通的根结点,以用于比较和合并;而 UNION 操作是指树之间的合并操作,包括路径压缩。另外,通过 Scan-line 方式的两遍扫描,先生成树,再给各树打上标记的方案,还适用于任意维、任意大小和邻接方式的图象^[1]。

2.2.2 完全下降化

若输入图象含有高地,那么首先要将其完全下降化。下降化的一种方法是根据定义 4 将源图象转换为完全下降的图象,具体可以借鉴极小区检测所使用的 FIFO 队列来传播高地像素的下降距离值;另一种方法是在流域变换的过程中,实时地计算到达高地下降边沿的距离,这可通过下一节提到的顺序队列来实现。

2.2.3 求最短路径,进行图象积分

这一步可分为有序和无序两类算法,这里有序和无序是相对像素被处理的次序而言的,其中有序是指像素被访问的次序由其距极小区的地形学距离决定,而无序则是指像素按照固定的次序被访问,即不强调先后。

(1) 有序算法

① 基于 Dijkstra-Moore 最短路径理论的数字图象积分(Integration)

算法的目的是通过计算最短地形学距离来判断某像素应该属于某个集水盆或分水岭,由于 Dijkstra-Moore 算法可以找出从一个结点到所有其他结点的最短路径,因此可以利用其来计算流域变换^[2]。算法引入标记图象 I_{lab} 和距离图象 $I_d, I_{lab}(p)$ 表示距 p 最近的极小区的标记, $I_d(p)$ 表示到 p 最近的极小区的距离值。其运算过程是从极小区像素开始,每次选取 $I_d(p)$ 最小的像素,按照代价函数(式(6))和下降坡度(式(5))的定义来更新其邻域像素在 I_d 中的值,并传播相应的标记,使得划入某集水盆的像素均处在通往其极小区的下降最快的路径上,最终的 I_{lab} 即为输出。

在算法实现时,可以通过层次式的顺序队列 OQ (Ordered Queues) 这种数据结构来保证像素的有序处理。OQ 是一个包含 n 个 FIFO 队列的优先级队列组。每一个 FIFO 队列服务于图象中的一个灰度级,且越低的灰度级具有越高的优先级^[11],初始化队列时,先存放各极小区的边沿像素,然后将灰度级为 h 的像素插入 OQ 中的优先级为 h 的 FIFO 队

列中;而 OQ 中的像素则按优先级从队列中移出,并按下面两种情况传播它的标记:将标记传给未标记的邻域像素,再将这些像素插入 OQ 中;将标记传给已在 OQ 中的邻域像素,若这些邻域像素的标记与当前传播的标记不同,则重新标记为分水岭像素。易见,利用灰度级的优先次序,算法总是将标记传播给坡度最陡的上游邻域像素。同时,对于高地内的像素,由于其中来自不同下降边沿像素的标记也以宽度优先的方式同步地传播,且 OQ 已自动地确定了高地内像素之间的顺序关系,从而不必对输入图象进行完全下降化的预处理工作。

但必须注意,算法实现中,OQ 并不是总能获得与完全下降化一致的结果,因为含有高地的图象经完全下降化后,高地内的像素均会划归给不同的集水盆而不存在分水岭像素,而 OQ 实现时,则可能因不同的传播路径在内地内相遇而产生分水岭,且其具体位置有赖于处理顺序,故一般偏向于后处理的下降边沿部分。

② 爬山法

爬山法是另一种相对简单的图象积分方法,它是利用集水盆中的点与极小区之间的测地线为最陡的下降路径这一性质,直接将标记传递给所有坡度最陡的上游邻域,而无须计算和更新距离值。如果算法不必区分分水岭,且式(5)、(6)中的 $d(p,q)$ 简化为 1,则可以更简单地用 p 的未标记邻域像素集合来取代坡度最陡的上游邻域像素集合,因为算法总是先处理最低灰度级的像素,所以 p 的未标记邻域必然在 p 的上游,而且由于该方法不区分分水岭像素,因此标记过的像素也不再被访问,这虽进一步节省了时间和内存,但缺点是因结果要依赖于处理次序而变得不够精确。

(2) 无序算法

由于这种算法不关心像素被处理的次序,因此常常使用经典的串行扫描方式。又由于算法在计算最短路径和传播标记时,仅考虑扫描方向上的像素,因此需要反复扫描,直到所有标记稳定不变为止。

3 结 语

本文综述了近年来国际上有关流域变换及其算法实现的研究成果,目的是为进一步地完善,乃至获得突破性进展提供可能和参考。研究与实践表明,流域变换的思想已成功应用于图象处理和分析领域,

并随着理论研究的深入,人们又提出了许多新的行之有效的实现算法,然而,由于数字定义与原始定义的近似程度,以及形式化定义与算法实现的差别造成了结果的不一,因此从精确和效率的角度出发,仍然有很大的研究空间。此外,还有一些相关的问题没有明确地讨论,有待进一步研究:

(1) 分水岭位置的准确性问题 通常认为流域变换中的数字距离是连续域情况的一个近似结果,它应当尽量地逼近欧几里德距离,这也是产生误差的原因之一。而 Meyer 提出应用 Chamfer 距离会得到更为精确的分水岭^[4],由此可见,研究适合于流域变换的距离公式有利于提高算法的精度。

(2) 过渡分割问题 原始形式的流域变换计算方法会由于输入图象中有过多的局部极小区而产生许多很小的集水盆,从而导致严重的过渡分割。虽已有一些方法用来弥补这个缺陷^[1],但尚待进一步完善。

(3) 算法并行化问题 效率低一直是困扰流域算法发展的问题,以致于 1998 年后,国内在这方面的研究遭到冷遇。但随着并行化手段的发展,流域算法的并行化又成为近几年国际上研究的热点^[8]。相关进展将在另文中讨论。

参考文献

- 1 Digabel H, Lantuejoul C. Iterative algorithms [A]. In: Chermant J L, Ed, Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine[M]. Stuttgart West Germany; Riederer Verlag, 1978; 85~99.
- 2 Beucher S, Lantuejoul C. Use of watersheds in contour detection [A]. In: Proc. International Work-shop on Image Processing, CCETT/IRISA[C]. Rennes, France, 1979; 17~21.
- 3 Vincent L. Graphs and mathematical morphology [J]. Signal Processing, 1989, 16(4): 365~388.
- 4 Meyer F. Topographic distance and watershed lines [J]. Signal Processing, 1994, 38(9): 113~125.
- 5 Beucher S. Watershed, hierarchical segmentation and waterfall algorithm [A]. In: Serra J, Soille P, Eds, Mathematical Morphology and its Applications to Image Processing [M]. Dordrecht, USA; Kluwer Acad. Publ., 1994; 69~76.
- 6 Vincent L, Soille P. Watersheds in digital spaces; an efficient algorithm based on immersion simulations [J]. IEEE Trans. Patt. Anal. Mach. Intell., 1991, 13(6): 583~598.

- 7 Roerdink J B, Meijster A. The watershed transform: definitions, algorithm and parallelization strategies [J]. Fundamenta Informaticae, 2000, 41: 187~228.
- 8 Moga A. Parallel watershed algorithms for image segmentation [D]. PhD thesis, Tampere University of Technology, Finland, 1997.
- 9 Dobrin B P, Viero T, Gabbouj M. Fast watershed algorithms: analysis and extensions [A]. In: Proc. of SPIE (IS&T/SPIE) Symposium on Electronic Imaging Science & Technology, Nonlinear Image Processing [C]. San Jose Convention Center CA, USA, 1994; 2180; 209~220.
- 10 Meijster A, Roerdink J B. A proposal for the implementation of a parallel watershed algorithm [A]. In: Hlavac A, Sara R, Eds, Computer Analysis of Images and Patterns (Lecture Notes on Computer Science) [M]. Berlin; Springer Verlag, 1995; 970; 790~795.
- 11 Beucher S, Meyer F. The morphological approach to segmentation; the watershed transformation [A]. In: Dougherty E R, Ed, Mathematical Morphology in Image Processing [M]. New York; Marcel Dekker, 1993; 433~481.
- 12 Meijster A, Roerdink J B. Computation of watersheds based on parallel graph algorithms [A]. In: Maragos P, Ed, Mathematical Morphology and its Applications to Image and Signal Processing [M]. Dordrecht USA; Kluwer Acad. Publ., 1996; 305~312.



周海芳 1975年生,1997年获国防科技大学计算机系工学学士学位,2000年获国防科技大学计算机学院工学硕士学位,现为国防科技大学计算机学院博士研究生。主要从事遥感图象处理算法及其并行计算方法的研究。



蒋艳凤 1976年生,1998年获国防科技大学计算机系工学学士学位,现为国防科技大学计算机学院博士研究生。主要从事图象信息挖掘和机器学习研究。

杨学军 1963年生,1990年获国防科技大学计算机系工学博士学位,教授,博士生导师,现任国防科技大学计算机学院院长。主要从事计算机体系结构、高性能计算和并行处理研究。