

# 基于最近邻有向图的遥感图像快速分割算法

崔宾阁<sup>1,2</sup> 孟翱翔<sup>1</sup>

(山东科技大学信息科学与工程学院 青岛 266510)<sup>1</sup> (国家海洋局第一海洋研究所 青岛 266061)<sup>2</sup>

**摘要** 针对现有的区域生长算法没有考虑到区域之间最近邻关系的有向性这一问题,提出了基于最近邻有向图的遥感图像快速分割算法。首先使用分水岭算法对遥感图像进行初次分割,然后在分割得到的区域对象基础上建立最近邻有向图。在区域生长过程中,沿着有向边形成的路径合并相邻的区域对象。当所有合并完成后重构区域对象的最近邻有向图,进行下一轮合并,直至区域数目不再变化。该方法避免了每次合并一个区域对象就重新计算新的邻居关系,从而降低了计算复杂度。实验结果表明,该方法分割结果比较合理,与其他几种方法相比运行效率明显提高。

**关键词** 遥感图像分割,最近邻有向图,区域生长

中图分类号 TP391.41 文献标识码 A

## Fast Remote Sensing Image Segmentation Algorithm Based on Nearest Neighbor Direct Graph

CUI Bin-ge<sup>1,2</sup> MENG Ao-xiang<sup>1</sup>

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China)<sup>1</sup>

(First Institute of Oceanography of the State Oceanic Administration, Qingdao 266061, China)<sup>2</sup>

**Abstract** The existing region growing algorithms do not take into account the direction of the nearest neighbor relations, which results in frequent rebuilt of the neighbor relations. In this paper, a fast algorithm for remote sensing image segmentation was proposed based on nearest neighbor directed graph. First of all, a remote sensing image was segmented using the watershed algorithm, and then a nearest neighbor directed graph was established on the basis of the region objects of the previous segmentation. In the region growing phrase, the adjacent region objects were merged along the directed edges. When the first round is finished, the nearest neighbor directed graph should be rebuilt, and the second round of region growing is initiated. This process repeats until the region number is no longer changed. This method avoids recalculating the neighbor relations whenever a merge happens, which reduces the computational complexity. The experimental results show that the algorithm proposed in this paper is more reasonable, more efficient compared with the other three algorithms.

**Keywords** Remote sensing image segmentation, Nearest neighbor directed graph, Region growing

## 1 引言

遥感图像分割是遥感图像分析和解译中的关键技术之一,对有效利用遥感数据、进一步进行信息提取与目标识别等工作具有重要意义。常用的分割算法可分为基于阈值的算法、基于边缘检测的算法、基于区域的算法等<sup>[1,2]</sup>。基于阈值的分割是通过设定不同的特征阈值,把图像像素分为具有不同灰度等级的目标和背景的区域<sup>[3]</sup>。基于边缘检测的算法是利用区域间灰度的不连续性来确定区域的边界位置<sup>[4]</sup>。基于区域的分割算法是利用图像空间特性进行分割的算法。区域生长算法是常用的基于区域的分割算法之一。区域生长是根据区域对象间的相似性质,从某一对象开始,将相邻的同质对象合并到当前对象中并逐步增长,直至没有可以合并的对象为止。在没有先验知识可以利用时,区域生长方法可以取得最佳的性能,经常用来分割比较复杂的图像<sup>[5-8]</sup>。

近年来,图论的概念不断被引入图像分割算法中。区域生长算法中的区域对象形成的最近邻图(Nearest Neighbor Graph, NNG)或者区域邻接图(Region Adjacent Graph, RAG)被提出用于控制对象合并<sup>[6-10]</sup>。Haris K等提出的全局最优法每次合并时总是选择NNG中总合并代价最小的顶点对<sup>[6]</sup>,合并完立即重建NNG,重复合并和重建等过程,直至最小全局合并代价大于给定阈值时算法终止。Adams R等提出了种子扩散的概念,即通过种子顶点的扩散进行区域生长<sup>[7]</sup>。选择RAG中的种子顶点不断与邻域合并,直至合并代价超过给定阈值转向下一个种子点的合并,在合并过程中,需要不断更新种子点的邻域关系。当所有种子点都不能再进一步合并时算法终止。汪闽等提出了光谱形状组合法<sup>[8]</sup>。在一轮合并循环中,按顺序每个种子点和最近邻合并后就转到下一个种子点,合并后将新顶点登记在一个临时NNG中,等所有种子点都处理完后根据临时NNG重建新NNG,然后在

到稿日期:2012-12-11 返修日期:2013-03-28 本文受国家自然科学基金(40906094)资助。

崔宾阁(1979-),男,博士,副教授,硕士生导师,主要研究方向为数据库、遥感数字图像处理,E-mail:cuibinge@yahoo.com.cn;孟翱翔(1987-),男,硕士生,主要研究方向为遥感数字图像处理。

此基础上重复合并,直至当所有顶点都不能再进一步合并时算法终止。

上述方法都没有考虑邻接图中边的方向问题,即不能够表达区域对象最近邻关系的有向性。例如,甲的最近邻是乙,但乙的最近邻不一定是甲。因而,全局最优法和种子扩散法每次都需要计算顶点与其所有邻接顶点的光谱异质性,花费时间较长。光谱形状组合法虽然每次只合并最近邻顶点,但是它需要频繁重建 NNG 图,因而合并速度也较慢。针对上述问题,本文提出了一种基于最近邻有向图的遥感图像快速分割算法,该算法根据最近邻关系将 NNG 中的边赋予了方向,形成最近邻有向图(Nearest Neighbor Direct Graph, NNDG)。由于预先存储了区域对象的合并方向,算法将根据区域合并路径实现区域对象的连续合并,避免了反复重建最近邻有向图,只有在所有对象均不能进一步合并时进行一次整体重建即可,因而提高了对象的合并效率。

本文首先描述了图像分割和 NNDG 的基本概念,并对 NNDG 中的顶点类型进行了划分;然后描述了区域对象的合并规则,在此基础上提出了最近邻有向图合并算法,并通过一个实例剖析了区域对象合并的具体步骤。最后,利用一幅航空遥感图像对包含本文算法在内的 4 种算法进行了分割效率和分割效果两方面的对比分析。

## 2 最近邻有向图分割算法原理

### 2.1 基本概念定义

**定义 1(图像分割)** 令集合  $R$  代表整个图像区域,对  $R$  的分割可看作将  $R$  分成若干个满足以下 5 个条件的非空子集(子区域) $R_1, R_2, \dots, R_n$ <sup>[1]</sup>:

- (1)  $\bigcup_{i=1}^n R_i = R$ ;
- (2) 对所有的  $i$  和  $j, i \neq j$ , 有  $R_i \cap R_j = \emptyset$ ;
- (3) 对  $i=1, 2, \dots, n$ , 有  $P(R_i) = \text{true}$ ;
- (4) 对  $i \neq j$ , 有  $P(R_i \cup R_j) = \text{false}$ ;
- (5) 对  $i=1, 2, \dots, n, R_i$  是连通的区域。

其中  $P(R_i)$  为作用于所有集合  $R_i$  中元素的逻辑谓词,如光谱特征相似; $\emptyset$  代表空集。

**定义 2(NNDG 最近邻有向图)** NNDG 可以用一个二元组描述:

$$NNDG = (V, VR) \quad (1)$$

式中,  $V$  是由区域对象组成的顶点集合,  $V = \{v_i | 0 \leq i \leq m, m \geq 0\}$ , 其中  $m$  为区域对象个数;  $VR$  是所有由顶点  $v$  到顶点  $v_{adj}$  的有向边组成的集合,  $VR = \{\langle v, v_{adj} \rangle | v, v_{adj} \in V\}$ , 其中  $v_{adj}$  是  $v$  的出边邻接顶点。

定义有向边  $\langle v, v_{adj} \rangle$  的权重  $H(v, v_{adj})$  为区域对象  $v$  和  $v_{adj}$  之间的光谱异质性。

在区域对象合并过程中,区域对象之间的关系不断变迁,因此 NNDG 的结构并不是固定的,而是存在一定的生命周期。定义所有对象与最近邻交互的最大持续时间  $T$  为 NNDG 的生命周期。

**定义 3(NNDG 时间序列  $S_{NNDG}$ )** 由一系列不同时间段的 NNDG 组成:

$$S_{NNDG} = \{NNDG_{T_i} | T_i \in T\} \quad (2)$$

式中,  $T = \{T_i | T_i = [t_i, t_{i+1}), 0 \leq i \leq n, n \geq 0\}$ ,  $n$  为 NNDG 演

变次数。  $T_i$  的时间宽度由  $NNDG_{T_i}$  中顶点的合并时间决定。

图论中顶点的出度是指顶点指向其它顶点的数量,由于 NNDG 中每个顶点只可能有一个最近邻,因此所有顶点的出度均为 1;入度  $d_v$  是根据指向当前顶点的顶点个数来确定的,公式为:

$$d_v = InDegree(v) \quad (3)$$

**定义 4(顶点类型)**

$$V_0 = \{v | d_v = 0 \& Circle(v, v_{adj}) = \text{false}, \forall v \in V\} \quad (4)$$

式中,  $V_0$  中顶点的入度  $d_v$  为 0,  $Circle(v, v_{adj})$  是顶点  $v$  与  $v_{adj}$  是否构成回路的逻辑谓词,此处不和出边邻接点构成回路。  $V_0$  中的顶点在合并路径中属于初始顶点。

$$V_N = \{v | d_v = N \& Circle(v, v_{adj}) = \text{false}, \forall v \in V, N \geq 1\} \quad (5)$$

顶点的入度  $d_v$  大于或者等于 1,并且和出边邻接点不构成回路。  $V_N$  的顶点在合并路径中属于中间顶点。

$$V_{Loop} = \{v | d_v = N \& Circle(v, v_{adj}) = \text{true}, \forall v \in V, N \geq 1\} \quad (6)$$

顶点入度  $d_v$  大于或者等于 1,并且和出边邻接点构成回路。  $V_{Loop}$  类型顶点在合并路径中属于中间顶点或结束顶点。

### 2.2 算法原理

基于最近邻有向图的遥感图像快速分割算法的基本思想是利用由形态学分水岭算法生成的区域对象构建最近邻有向图 NNDG,并根据 NNDG 中顶点之间有向边的方向和顶点类型指导区域对象的合并过程。合并过程采用一种标记方式,即如果某顶点与邻接点满足合并条件,那么将该点(或者其邻接点)标记为对方顶点。逻辑上被合并到一起的两个顶点在物理上依然相对独立,各对象之间拓扑关系没有改变。当所有对象不能再进一步合并时,计算剩余对象之间的拓扑关系,重建 NNDG,然后在此基础上继续合并,直至最后区域对象个数不再改变时算法结束。

通过 2.1 节定义 4 可知,在 NNDG 中有 3 种类型的顶点。根据入度大小以及是否构成回路,本文定义以下 3 种合并规则:

**规则 1**  $v \in V_0$ , 根据出边邻接点  $v_{adj}$  是否被其它顶点合并,分两种情况考虑:(a)  $v_{adj}$  未被合并,则将  $v_{adj}$  合并至  $v$ ;(b)  $v_{adj}$  被合并到  $v_{adj}'$ ,则将  $v$  合并到  $v_{adj}'$ 。

**规则 2**  $v \in V_N$ :  $v$  作为中间顶点,其合并状态在处理前存在两种情况:(a)  $v$  未被合并,按照  $v \in V_0$  类型顶点的合并规则处理;(b)  $v$  被合并到  $v'$ ,根据出边邻接点  $v_{adj}$  是否被合并,分两种情况考虑:

- ①  $v_{adj}$  未被合并,将  $v_{adj}$  合并至  $v'$ ;
- ②  $v_{adj}$  被合并到  $v_{adj}'$ ,考察  $v'$  与  $v_{adj}'$  是否相同。如果  $v'$  与  $v_{adj}'$  相同,那么合并终止;如果不相同,将  $v'$  合并到  $v_{adj}'$ 。

**规则 3**  $v \in V_{Loop}$ : 处理过程与  $v \in V_N$  类似,但是该顶点与其出边邻接点  $v_{adj}$  构成回路,因而必须终止合并过程。考察合并路径:  $v \rightarrow v_{adj} \rightarrow v' \rightarrow v_{adj}' \rightarrow v_{adj} \dots$ 。如果  $v_{adj}$  被合并到  $v$ ,则在执行  $v_{adj} \rightarrow v$  时可以发现  $v_{adj}$  与  $v$  已经合并,所以此时应终止合并过程。

### 2.3 最近邻有向图合并算法描述

**算法 1** 基于 NNDG 的遥感图像快速分割算法

输入参数: 遥感图像  $R$ ; 分割阈值  $S$

输出参数:分割后的遥感图像  $R'$

算法步骤:

- (1)使用分水岭算法将  $R$  分割成若干个区域对象,用  $R^{(0)}$  表示: $R^{(0)} = \{R_1^{(0)}, R_2^{(0)}, \dots, R_{n_0}^{(0)}\}$ ,其中  $n_0$  为  $R^{(0)}$  集合中区域对象个数。初始化循环计数器变量  $i=0$ ;
- (2)计算  $R^{(0)}$  中每个区域对象的邻域表、最近邻和像元光谱均值等属性,构建  $NNDG_0$ ,更新  $NNDG$  时间序列  $S_{NNDG}$ ;
- (3)遍历集合  $R^{(0)}$  中的所有区域对象。对于每个区域对象,根据 2.1 节定义 4 判断该对象在  $NNDG_0$  中所属顶点类型,然后根据 2.2 节中的相应规则进行合并;
- (4)循环计数器变量  $i=i+1$ ,计算合并后的区域对象个数  $n_i$ ,如果  $n_i = n_{i-1}$ ,说明合并前后区域对象个数并没有变化,转到步骤(5)输出分割结果;否则返回步骤(2);
- (5) $R' = R^{(i)} = \{R_1^{(i)}, R_2^{(i)}, \dots, R_{n_i}^{(i)}\}$ ;
- (6)输出  $R'$ ,算法结束。

## 2.4 区域对象合并过程示例

下面通过一个实例描述基于  $NNDG$  的遥感图像分割算法的执行过程。图 1 是遥感图像经过分水岭算法分割后生成的区域对象分布示意图。由该图中的区域对象生成的  $NNDG$  中包含所有 3 种类型的顶点,其对说明本算法的合并过程具有代表性。由图 1 创建的  $S_{NNDG}$  序列中的第一个  $NNDG_0$  如图 2 所示。表 1 是  $NNDG_0$  中区域对象的属性表。

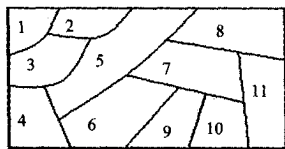


图 1 区域对象分布示意图

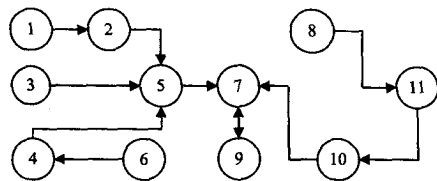


图 2 由图 1 所示区域对象生成的  $NNDG_0$

表 1 区域对象属性列表

顶点编号	顶点邻域表	最近邻顶点	顶点类型	权重 $H$ 与 阈值 $S$ 比较
1	2,3	2	$V_0$	$H < S$
2	1,3,5	5	$V_N$	$H > S$
3	1,2,4,5	5	$V_0$	$H < S$
4	3,5,6	5	$V_N$	$H < S$
5	2,3,4,6,7,8	7	$V_N$	$H < S$
6	4,5,7,9	4	$V_0$	$H < S$
7	5,6,8,9,10,11	9	$V_{Loop}$	$H < S$
8	5,7,11	11	$V_0$	$H < S$
9	6,7,10	7	$V_{Loop}$	$H < S$
10	7,9,11	7	$V_N$	$H > S$
11	7,8,10	10	$V_N$	$H < S$

区域对象的合并过程描述如下:

第一轮合并:

顶点 1 的合并路径为  $1 \rightarrow 2 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $1 \rightarrow 2$  合并: $H(1,2) < S$ ,将 2 合并到 1; $2 \rightarrow 5$  合并: $H(2,5) > S$ ,顶点 1 合并过程终止。

顶点 2 的合并路径为  $2 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $2 \rightarrow 5$  合并:由于 2 和 5 已经试合并过,因此合并过程终止。

顶点 3 的合并路径为  $3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $3 \rightarrow 5$ : $H(3,5) < S$ ,将 5 合并到 3; $5 \rightarrow 7$ : $H(5,7) < S$ ,将 7 合并到 5,由于 5 已被合并到 3,因此将 7 合并到 3; $7 \rightarrow 9$ : $H(7,9) < S$ ,同理将 9 合并到 3。 $9 \rightarrow 7$ :由于 7 和 9 已经合并过,合并过程终止。

顶点 4 的合并路径为  $4 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $4 \rightarrow 5$ : $H(4,5) < S$ ,将 4 合并到 3; $5 \rightarrow 7$ :5 和 7 已经合并过,合并过程终止。

顶点 5 的合并路径为  $5 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $5 \rightarrow 7$ :由于 5 和 7 已经合并过,因此合并过程终止。

顶点 6 的合并路径为  $6 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $6 \rightarrow 4$ : $H(6,4) < S$ ,将 6 合并到 3; $4 \rightarrow 5$ :4 和 5 已经合并过,合并过程终止。

顶点 7 的合并路径为  $7 \rightarrow 9 \rightarrow 7$ 。 $7 \rightarrow 9$ :7 和 9 已经合并过,合并过程终止。

顶点 8 的合并路径为  $8 \rightarrow 11 \rightarrow 10 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $8 \rightarrow 11$ : $H(8,11) < S$ ,将 11 合并到 8; $11 \rightarrow 10$ : $H(11,10) < S$ ,将 10 合并到 8; $10 \rightarrow 7$ : $H(10,7) > S$ ,顶点 8 合并过程终止。

顶点 9 的合并路径为  $9 \rightarrow 7 \rightarrow 9$ 。 $9 \rightarrow 7$ :7 和 9 已经合并过,合并过程终止。

顶点 10 的合并路径为  $10 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $10 \rightarrow 7$ :10 和 7 已经试合并过,合并过程终止。

顶点 11 的合并路径为  $11 \rightarrow 10 \rightarrow 7 \rightarrow 9 \rightarrow 7$ 。 $11 \rightarrow 10$ :11 和 10 已经合并过,合并过程终止。至此,第一轮合并结束。

第一轮合并结束后对区域对象重新编号,分布如图 3 中所示。

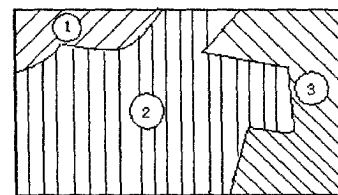


图 3 第一轮循环后各对象所属区域

第一轮合并后剩下 3 个区域对象,它们之间的拓扑关系需重新计算。重建后的  $NNDG_1$  如图 4 所示。

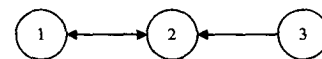


图 4  $NNDG_1$  示意图

第二轮合并:

同理进行第二轮合并。第二轮合并后所剩区域对象个数与第二轮合并之前的区域对象个数相等,整个合并过程至此结束,输出分割结果。

## 3 实验与结果分析

### 3.1 数据来源与实验平台

为检验本算法对遥感图像分割的效率和效果,本文进行了如下实验。实验数据是分辨率为 0.88 米的航空遥感图像,尺寸为  $1400 \times 900$ ,如图 5 所示。本文统一实验平台为 Windows XP sp3 操作系统,系统配置为 2.0GHz 双核 CPU,2GB 内存。本文提出的算法以及对比较算法均是在 Visual Studio 2010 上使用 C++ 语言实现。



图5 0.88米高分辨率航空遥感图像

### 3.2 实验过程描述

本实验对引言中提到的3种分割算法<sup>[6-8]</sup>以及本文提出的算法进行了分割效率和分割效果两方面的对比,分割阈值统一设置为400。对于分割结果的好坏,目前并没有一种公认的客观评价方法,一般采用的是专家评价方法。本文除了采用专家目视评价方法外,还引入文献<sup>[11]</sup>中提出的均值方差指标来进行评价。其中,均值是指组成某个对象的所有像元光谱的平均值,而均值方差则是指整幅图像中所有对象均值的方差。

在面向对象的图像分析中,图像对象类型的差异对信息提取的精度有很大影响。根据组成对象的像元类别不同,对象可分为纯对象、混合对象等多种对象<sup>[11]</sup>。

**纯对象:**当组成图像对象的所有像元属于同一地物类别时,称之为纯对象。此类对象中像元具有一致性,光谱差异小,因此方差较小。

**混合对象:**对象内像元属于不同的地物类别,光谱差异大,因此方差较大。

当图像中纯对象增多时,相邻对象间的光谱差异增大,整幅图像的均值方差增大;相反,当图像中混合对象增多时,相邻对象间的光谱差异降低,整幅图像的均值方差变小。一般来说,对于同一阈值下不同算法分割出的图像,均值方差越大代表该分割算法分割效果越好。各算法分割效果如图7中所示,分割效率和分割精度对比分别如图7和图8中所示。

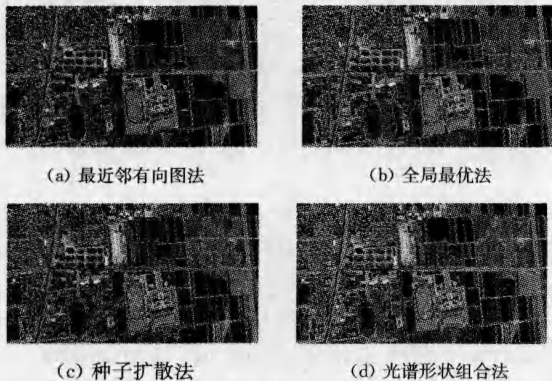


图6 4种算法分割效果对比图

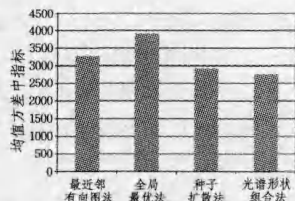
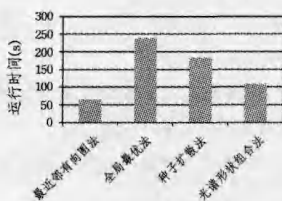


图7 4种算法分割时间对比

图8 4种算法分割精度对比

### 3.3 实验结果分析

图6(a)中采用了本文提出的基于最近邻有向图的图像分割算法,可以看出中间和右部的农田、操场以及广场均被完

整地分割出来。植被和农闲地、两个操场的跑道和田径场均被正确分开,且边界清晰、光滑。图像左下角和中部上方的厂房(蓝色区域)分割效果也比较好。由于图像左下角部分地物分布较为杂乱,各种算法的分割结果都较为破碎,边界比较弯曲。图6(b)中采用的是全局最优分割算法,可以看出该算法对于农田、操场、广场等内在尺度较大的地物的分割效果也较好。图6(c)中采用的是种子扩散分割算法,图6(d)中采用的光谱形状组合分割算法,它们对中间操场的分割效果较差,过分割现象严重。图7(c)中图像右部部分农田分割得比较破碎,并且边界标记弯曲,与原始图像中的农田边界基本不符。

通过以上分析,可以看出4种算法对结构简单并且内在尺度较大的地物分割结果较好,如对农田、操场的分割。但是种子扩散法和光谱形状组合法过分割现象较为明显,整体分割结果含杂点、碎斑较多。部分地物存在较大程度的误分割,比如农田、操场。本文算法是按照最近邻的顺序合并对象,并且在总体合并代价局部最小的基础上实现的,合并过程与其它3种方法相比简单直接,但是分割结果中存在一定的过合并现象;4种算法对于结构比较复杂的地物分割结果均不太理想,比如图像左上角的居民楼以及农村民居等,但是全局最优法和最近邻有向图法仍表现较好。

从图7中的图像分割时间上看,全局最优法耗时最长,达到240秒;种子扩散法也达到170秒左右;光谱形状组合法分割时间在110秒左右;本文算法花费时间在60秒左右,效率是其它3种算法的2~4倍。

从图8中的图像分割效果定量评价上看,全局最优法分割结果均值方差最高,达到3800左右;其次是本文提出的算法,均值方差在3280左右;其他两种算法的均值方差均低于3000。本文算法由于采用的合并代价并非全局最小,因此从分割效果上看离全局最优化尚有一定的差距,但是比种子扩展法和光谱形状组合法要好。

从实验中可以得出如下结论,本文算法大幅提高了分割效率,同时也牺牲了一定的分割效果。因此下一步工作将是引入一些启发式规则和优化算法,使得合并代价接近全局最小,分割效果逼近全局最优。

**结束语** 文中提出了一种基于最近邻有向图的遥感图像快速分割算法。该算法避免了反复重建最近邻有向图的高耗时操作,使得合并速度大大提高。实验结果表明,与传统区域生长算法相比,该算法可以获得更高的分割效率;专家目视评价该算法分割结果合理,比大多数算法表现更佳。本文还引入均值方差指标对不同算法的分割结果进行定量化评价,证明了本文算法分割结果的有效性和合理性,算法的分割结果可以满足高分辨率遥感图像分类的需要。

### 参考文献

- [1] 林开颜,吴军辉,徐立鸿.彩色图像分割方法综述[J].中国图像图形学报,2005,10(1):1-10
- [2] Wu Jian. Study on Segmentation of Color Remote Sensing Image [J]. Procedia Engineering, 2012, 29: 3312-3316
- [3] Manikantan K, Arun B V. Optimal Multilevel Thresholds based on Tsallis Entropy Method using Golden Ratio Particle Swarm Optimization for Improved Image Segmentation [J]. Procedia Engineering, 2012, 30: 364-371

- [4] Priya G G L, Domic S. Edge Strength Extraction using Orthogonal Vectors for Shot Boundary Detection [J]. Procedia Technology, 2012, 6: 247-254
- [5] Kang C-C, Wang W-J, Kang C-H. Image segmentation with complicated background by using seeded region growing [J]. AEU-International Journal of Electronics and Communications, 2012, 66(9): 767-771
- [6] Haris K, Efstratiadis S N, Maglaveras N, et al. Hybrid Image Segmentation Using Watersheds and Fast Region Merging [J]. IEEE Transaction on Image Processing, 1998, 7(12): 1684-1699
- [7] Adams R, Bischof L. Seeded Region Growing [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(6): 641-647
- [8] 汪闽, 万其明, 张大筹, 等. 光谱、形状特征结合的多精度图像分割算法与应用 [J]. 地球信息科学学报, 2010, 12(2): 261-268
- [9] Ding Jun-di, Ma Ru-ning, Chen Song-can, et al. A Fast Directed Tree Based Neighborhood Clustering Algorithm for Image Segmentation [J]. Lecture Notes in Computer Science, 2006, 4233: 369-378
- [10] Araújo A R F, Costa D C. Local adaptive receptive field self-organizing map for image color segmentation [J]. Image and Vision Computing, 2009, 27(9): 1229-1239
- [11] 黄慧萍, 等. 面向对象图像分析中的尺度问题研究 [D]. 北京: 中国科学院研究生院, 2003

(上接第 251 页)

误差分别比 SVM、Kalman 模型低了 6.91% 和 4.26%; 高峰时段预测中, 组合预测的相对误差分别比 SVM、Kalman 模型低了 10.10% 和 7.87%; 从表 3 中可以看到, 3 种预测方法的均等系数都在 0.90 以上, 且组合预测方法的均等系数均高于其余两种单项预测模型。

此外, 本文还将组合预测方法与文献[18]中提出的基于 K 近邻的预测方法作了比较, 该文中采用的是 10 号检测器 2006 年全年流量数据, 交通规律与本文中 2012 年数据一致, 文中预测了 2007 年 1 月 1 日到 1 月 5 日共 5 天的交通流量。表 4 记录了文献[18]与本文方法的对比结果, 可以看出, 本文提出的组合预测模型在全天预测中, 比基于 K 近邻预测方法误差降低了 11.44%, 在高峰时段的预测中, 误差降低了 17.98%。因此, 本文提出的组合预测方法是令人满意的。

表 4 文献[18]基于 K 近邻方法与本文组合预测方法误差对比

评价指标	文献[18]预测方法	本文预测方法	误差降低率
全天修正相对误差	14.86%	13.16%	11.44%
高峰时段相对误差	9.29%	7.62%	17.98%

**结束语** 不同的单项预测方法在不同时段的预测效果不一样, 本文提出的组合预测模型旨在智能地选取针对某一时段相应的更准的单项预测方法。其思想简单直观: 通过过去几个时段流量实测值与单项预测方法预测值之间的误差平方和及相关系数两重标准, 来决定下个时段采用何种预测方式, 消除了传统组合预测模型中, 通过改善某种拟合误差进行线性规划得出各单项预测模型所占权重而造成推广性不强的弊端。此外, 支持向量机与卡尔曼滤波两种方法的结合既包含了实时动态变化的趋势, 也潜在包含了历史各时段的流量规律。实验表明, 组合预测模型效果令人满意, 具有普遍性。作者下一步的主要工作就是: 在单步预测的基础上, 将组合预测模型运用到交通流量多步预测上, 为基于预测的导航系统提供服务。

## 参 考 文 献

- [1] 邵春福, 熊志华, 姚智胜. 道路网短时交通需求预测理论、方法及应用 [M]. 北京: 清华大学出版社, 2011
- [2] 许伦辉, 傅惠. 交通信息智能预测理论与方法 [M]. 北京: 科学出版社, 2009
- [3] 宋驰, 沈国江. 短时交通流预测模型综述 [J]. 自动化博览, 2012(6): 84-87
- [4] 陈华友, 盛昭瀚, 刘春林, 等. 基于向量夹角余弦的组合预测模型的性质研究 [J]. 管理科学学报, 2006, 9(2): 1-8
- [5] 孙李红, 沈继红. 基于相关系数的加权几何平均组合预测模型的性质 [J]. 系统工程理论与实践, 2009, 29(9): 84-91
- [6] 李斌, 郝涛, 史明华, 等. 基于支持向量机的交通流组合预测模型 [J]. 天津工业大学学报, 2008, 27(2): 73-76
- [7] 韩冬梅, 牛文清, 杨荣, 等. 线性与非线性最优组合预测方法的比较研究 [J]. 情报科学, 2007, 25(11): 1672-1678
- [8] 张大斌, 张景广, 彭森, 等. 基因表达式编程在组合预测建模中的应用 [J]. 系统工程理论与实践, 2012, 32(3): 568-573
- [9] Wu Chun-hsin, Wei Chia-chen, Chang Ming-hua, et al. Travel time prediction with support vector regression [J]. IEEE Transaction on Intelligent Transportation systems, 2004, 5(12): 276-281
- [10] Theja P V V K, Vanajakshi L. Short Term Prediction of Traffic Parameters Using Support Vector Machines Technique [C] // Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on. Goa, India, 2010: 70-75
- [11] 龚珊, 尹相勇, 朱爱华, 等. 基于浮动车的路段行程时间卡尔曼滤波预测算法 [C] // 2008 第四届中国智能交通年会论文集. 2008: 1-6
- [12] Chang Ming-wei, Lin Chi-hjen. Leave-one-out Bounds for Support Vector Regression Model Selection [J]. Neural Computation, 2005, 17(5): 1188-1222
- [13] Kim K J. Financial Time Series Forecasting Using Support Vector Machines [J]. Neurocomputing, 2003, 55(3): 307-319
- [14] Kalman R. A new approach to linear filtering and prediction problems [J]. Journal of Basic Engineering, 1960, 82(01): 35-46
- [15] 陆如华, 徐传玉, 张玲, 等. 卡尔曼滤波的初值计算方法及其应用 [J]. 应用气象学报, 1999, 12(3): 63-67
- [16] 杨兆升, 王媛, 管青, 等. 基于支持向量机方法的短时交通流量预测方法 [J]. 吉林大学学报: 工学版, 2006, 36(6): 881-884
- [17] 沈国江, 王啸虎, 孔祥杰, 等. 短时交通流量智能组合预测模型及应用 [J]. 系统工程理论与实践, 2011, 31(3): 561-568
- [18] 张涛, 陈先, 谢美萍, 等. 基于 K 近邻非参数回归的短时交通流预测方法 [J]. 系统工程理论与实践, 2010, 30(2): 376-384