

基于 ABTM 的城市建筑物合并算法

钱海忠 武芳 谭笑 邓红艳

(信息工程大学测绘学院, 郑州 450052)

摘要 大比例尺城市建筑物合并算法的研究一直是制图综合的重点。为此,分析了目前面状建筑物合并算法存在的不足,提出了把 Agent 技术与 TIN 技术、聚类技术相结合的算法 (ABTM),以提高城市建筑物合并的速度、智能化和自动化水平。首先对地图数据进行聚类分析,以对地图数据进行分区域处理;然后把约束 Delaunay 三角剖分技术运用到本算法中来,提出了一种对 Delaunay 三角形分类的方法,结合 Agent 技术,定义了一种以 Delaunay 三角形为基础的 Agent 层次概念,并对不同层次的 Agent 定义了具有生命周期的数据结构。依据 Agent 的层次结构、TIN 图形计算模型和制图综合约束条件,提出了一种基于 Agent 和 TIN 技术的建筑物自动合并新算法。最后给出了综合实例,并对结果进行了量化分析。分析结果表明,本算法具有比其他类似算法更高的整体性能。

关键词 地图综合 聚类分析 合并 Delaunay 三角形 Agent

中图分类号: P238 **文献标识码**: A **文章编号**: 1006-8961(2005)10-1224-10

The Algorithm for Merging City Buildings Based on ABTM

QIAN Hai-zhong, WU Fang, TAN Xiao, DENG Hong-Yan

(Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450052)

Abstract The city-building merging algorithm is important in automated generalization of large-scale map. After discussing the shortcomings of current algorithms, a new method based on the technique of agent, TIN and clustering is put forward, which is expected to improve the operation speed, intelligentized and roboticized level of generalization. First, a clustering analysis with map data is done in order to divide whole data into several parts. Second, this algorithm regards Delaunay triangle as its basic technology, and takes a new classifying criterion to it. Third, based on the TIN classifying, an agent layer is expressed. And for every different agent layers, different agent life cycles and data structures are defined. Finally, as the final goal, a new algorithm is evolved from the techniques of agent, TIN technique of graphics calculation, and the map generalization constrains. An example and relevant analysis are presented in the end. The result shows that this algorithm is more excellent than others not only in functions but also in the speed, intelligence and automatization level. On the other hand, some modification to the algorithm is discussed.

Keywords map generalization, clustering analysis, merge, Delaunay triangulation, Agent

1 引言

地理信息的自动综合问题目前仍是困扰 GIS (地理信息系统) 及地图学界实现空间数据自动处理与可视化的国际难题^[1]。GIS 的迅速发展和广泛应用,使得自动综合的地位越来越明显和突出。当 GIS 中地理空间数据的比例尺缩小后,目标间不可

避免地产生了冲突,因此必须对地理空间数据进行制图综合处理,以满足显示和生产的需要。在城市 GIS 广为建立的今天,自动创建更小分辨率的空间数据库和相应的地形图已成为一个突出的问题。在迫不得已的情况下,人们采取了人机交互方式来应付急需^[2]。城市 GIS 中的主体是面状建筑物,随着图形比例尺的缩小,建筑物之间产生了压盖、相邻等冲突,因此需要通过一系列的综合操作比如删除、合

基金项目:国家自然科学基金项目(40471115)

收稿日期:2004-09-27;改回日期:2005-03-04

第一作者简介:钱海忠(1976~),男,讲师。现为信息工程大学测绘学院地图学与地理信息工程专业博士研究生。主要研究方向为地理信息系统与辅助决策系统、地图自动综合、GIS 下多尺度空间数据表达等。E-mail:qianhaizhong2005@163.com

并等来解决比例尺缩小后图形的清晰性问题。而其中对面状建筑物的合并问题,是整个城市 GIS 中制图综合的关键之一和难点所在。

在数字环境下,由于街道、街区等地理实体的位置及其相互关系的识别要靠相应的数据结构来支持,而当前的 GIS 中尚缺少上述所需的完备数据结构^[2]。由于空间数据的复杂性,数据结构的完备性主要体现在地图综合算法应具有灵活处理复杂地理数据的能力。但简单的数据模型对于多用途的综合任务其功能是很有限的^[3]。

TIN 的出现,对实现灵活而复杂的地图综合数据模型是一次很大的帮助。Delaunay 三角剖分算法是计算几何中十分重要的研究领域,因其强大的空间探测与分析能力,导致了与 Delaunay 三角网相关的算法层出不穷。由于 Delaunay 三角网能将地理实体纳入到三角网描述体系中,通过三角网中的点、边、三角形及各种关系,对地理实体进行形式化的描述,因此对地图综合的各种操作十分有利。

但近年来,随着技术的发展,图形算法的智能化要求越来越高。TIN 技术虽然几何处理功能非常强大,但面对智能化的挑战,仍满足不了地图综合的需求。而 Agent 技术的出现,较好地填补了这一空缺^[4]。Agent 研究最早起源于分布式人工智能(DAI),目前已成为人工智能中研究的核心问题之一。Agent 研究人员目前达成的共识是:Agent 是设计来完成某类任务的,能在一定环境中自主发挥作用的、有生命周期的计算实体,具有自主性、反应性、能动性、学习性、通信性、移动性等特性^[5]。因此,TIN 技术与 Agent 技术在功能上能够相互弥补,把这两项技术进行交叉结合,是一项可取的措施。况且,广义的 Agent 概念可以衍生为继第四代面向对象的编程技术后的,具有更高层次和内涵的第五代面向 Agent 实体的编程技术^[6],在地图综合数据模型的实现上更为直观和简单。本文正是采用了这种基于 Agent 的 TIN 技术(agent based TIN model-ABTM)进行了大比例尺城市建筑物合并算法的研究与开发。

2 城市建筑物合并的特点

城市建筑物面状轮廓与其他要素相比,明显具有其特殊性。对面状建筑物而言,具有总体上成群分布、形状规则、坐落有序等特点。在 FDDM 模型

中,自动化制图综合算子的设计应充分考虑面向目标的几何特征、地理特征,建筑物(群)多边形的矩形化特点,使其化简合并与一般多边形综合处理不同^[7];在大比例尺地图自动化综合中,建筑物(群)的合并与化简是一个重要问题。建筑物目标的边界主要由一些垂直线段构成,建筑物多边形可以看作是一系列矩形的并、差结果,这使得综合时应充分考虑其形状轮廓的矩形直角化特点^[8]。而面状居民地在位置分布和相互形态关系上主要表现为错位情况、凸状情况、凹状情况和复合情况等 4 种类型^[9]。特别注意的是,用 Delaunay 三角形进行面状要素的合并时,主要考虑的是对边界三角形的处理,而对内部三角形可以不加考虑,因此,具有“外表相似性”的面目标采用 Delaunay 三角剖分进行合并时可以采用相似的处理过程。这一规律的发现可以将面状要素的合并分解为有限可穷举的几类问题进行处理,从而大大简化了面状要素的综合难度^[9]。

3 基于 ABTM 的建筑物合并模型

基于 ABTM 合并模型的总体思路是:首先对待合并的图形目标进行聚类分析,然后对每个类中的目标进行基于约束 Delaunay 的三角剖分,并把 Agent 技术运用到约束 TIN 模型中来,形成 Agent 的 3 个层次。把城市建筑物综合的约束条件作为底层技术支持,再考虑建筑物的特殊性,在 Agent 的每个层次上进行相应的图形综合操作,最后进行直角化处理,从而完成建筑物的合并过程。

3.1 对建筑物目标聚类

聚类分析是研究多要素事物分类问题的数量方法。之所以对地图上待综合的目标进行聚类,是因为一方面可以得到聚类信息,对目标分布情况的认识进一步加强;另一方面通过聚类,每个类被认为是独立的,类间不存在任何综合的依赖性,从而把对整个区域的综合转化为对每个类单独的综合,综合速度大大加快。

算法中,需进行两个层次的聚类:第 1 个层次的聚类是采用道路网、河流等对地图目标进行划分,即一个类不能跨越道路和河流,以防止综合后的建筑物出现跨越道路或河流的错误结果,因此需要利用道路网和河流对待综合的区域进行分割,把建筑物划分为相互独立的集合。该步骤是对目标进行的第一次聚类,被称为区域意义上的聚类。第 1 次聚类

结束后,可以从宏观上对待合并的区域目标进行约束和控制。

第 2 个层次上的聚类是对第 1 个层次聚类得到的结果再进行几何意义上的聚类,即按照几何条件进行聚类。这里采用欧氏距离聚类的方法。设两个多边形 i 和 j , 分别拥有 p 个和 q 个内点, 采用多边形内点间的最小距离作为多边形间聚类的指标, 即

$$d_{ij} = \min \sqrt{(x_{im} - x_{jn})^2 + (y_{im} - y_{jn})^2} \quad (1)$$

($0 < m < p, 0 < n < q$)

如果 $d_{ij} < l_{\min}$, 则这两个目标将被聚到一个聚类中。这里 l_{\min} 为制图综合中要求的最小距离变量, 与制图比例尺有关。一般制图比例尺为系列比例尺, 比如 2.5 万、5 万、10 万、25 万、50 万、100 万等。不同比例尺对目标间的最小距离有不同的要求, 表 1 是不同比例尺地形图对集、镇及街区的距离要求参数。

表 1 各比例尺对目标间距离的要求

Tab.1 Different min length for different scale

比例尺	最小距离要求 (mm)
1:25 000	0.3
1:50 000	0.3
1:100 000	0.3
1:250 000	0.3
1:500 000	0.5
1:1 000 000	0.6

图 1 是城市街区进行两次聚类后的结果。



图 1 依道路聚类的结果

Fig. 1 The result of clustering

3.2 基于聚类的 Delaunay 三角剖分及其分类

通过聚类, 把地图数据分成了一系列的数据块, 然后在聚类的基础上分别对各个数据块进行约束 Delaunay 三角剖分。而如何快速、高效地构建

Delaunay 三角网, 一直是众多学者研究和关注的焦点。迄今为止已出现了不少成熟的算法, 主要有分割-合并算法、逐点插入法和三角网生长法等。后两种方法由于时间复杂度差及效率较低, 目前较少采用; 前种方法高效但算法复杂, 而其中基于格网的分割-合并技术则相对简单且有效, 为此采用了这种基于格网分割的 Delaunay 三角网划分技术。

首先对数据进行格网分割, 然后对各个格网内的数据分别进行构网, 最后把格网进行合并, 并且把格网边缘的三角形进行局部优化处理。之所以需要处理, 主要是格网边缘的三角形在构网时没有考虑到其他格网数据的影响, 因此格网合并后其边缘可能出现不符合 Delaunay 三角形条件的三角形。而解决此问题的关键技术是交换四边形对角线 (图 2)。即已知四边形 4 个顶点 N_1, N_2, N_3, N_4 的坐标分别为 $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, 设 $a_1 = \angle N_1 N_3 N_2, a_2 = \angle N_1 N_4 N_2$, 由余弦定理得:

$$\sin(a_1 + a_2) = ((x_1 - x_3)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_3)) \times ((x_2 - x_4)(x_1 - x_4) + (y_2 - y_4)(y_1 - y_4)) + ((x_1 - x_3)(x_2 - x_3) + (y_1 - y_3)(y_2 - y_3)) \times ((x_2 - x_4)(y_1 - y_4) - ((x_1 - x_4)(y_2 - y_4))) \quad (2)$$

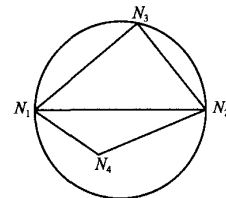


图 2 对角线交换示意图

Fig. 2 Sketch map of changing diagonal

N_1, N_2, N_3 确定一个圆, 根据圆准则可做如下判断:

- (1) 当点 N_4 位于圆周外, 即 $(a_1 + a_2) < \pi$ 时, 对角线不需要交换。
- (2) 当点 N_4 位于圆周上, 即 $(a_1 + a_2) = \pi$ 时, 对角线任意选择。
- (3) 当点 N_4 位于圆周内, 即 $(a_1 + a_2) > \pi$ 时, 对角线需要交换。

因此, 可根据 $\sin(a_1 + a_2)$ 的符号来决定是否交换对角线。当 $\sin(a_1 + a_2) > 0$ 时, 不交换对角线; 当 $\sin(a_1 + a_2) = 0$ 时, 可交换也可不交换对角线; 当 $\sin(a_1 + a_2) < 0$ 时, 必须交换对角线。

但是建筑物的合并过程需要把建筑物的轮廓边

强制作为 Delaunay 三角形的边,因此需要对三角网进行约束边处理。这里采用了约束线段嵌入的迭代算法^[10],即如图 3 所示为 Delaunay 三角网,设当前处理线段为 P_iP_j ,由与 P_iP_j 相交的三角形组成的区域称为 P_iP_j 的三角形影响域 $M_T = \{T_1, T_2, \dots, T_n\}$,而由 M_T 中三角形的外围边组成的多边形为影响多边形 $Q = \{P_1, P_2, \dots, P_i\}$ 。对于 Q ,有如下结论:

(1) Q 是一简单多边形,且 P_iP_j 为 Q 的一条对角线,即 $P_iP_j \in Q$,从而 P_iP_j 把 Q 分成 Q_L 和 Q_R 两部分,且 Q_L 和 Q_R 也为简单多边形。

(2) 可分别对 Q_L 和 Q_R 进行三角剖分及优化(这里指的优化即是图 2 所示的优化算法)。

(3) 对 $P_k \in Q$,若 P_k 为到 P_iP_j 的最近点($P_k \neq P_i, P_k \neq P_j$),则一定有 $P_iP_k \in Q, P_kP_j \in Q$ 。

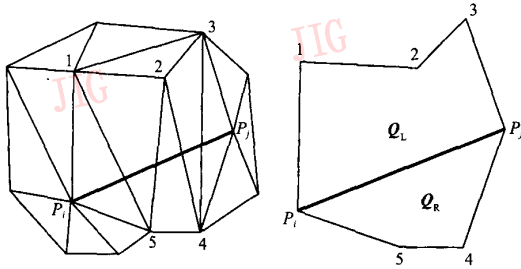


图 3 约束线段的嵌入算法
Fig. 3 Algorithm of inseting constrain line

在聚类的基础上分别对各个数据块进行约束 Delaunay 三角剖分后,得到了约束 Delaunay 三角网。根据算法需要,对三角形进行如下分类:

(1) 按归属分

I 类三角形:面目标内部的三角形。

II 类三角形:面目标外部的,且其 3 个顶点属于同一个面的三角形。

III 类三角形:面目标外部的,且其 3 个顶点分属于两个面的三角形。

IV 类三角形:面目标外部的,且其 3 个顶点分属于 3 个面的三角形。

(2) 按位置分

边界三角形:三角形的 3 条边中至少有一条边没有相邻三角形。

内部三角形:三角形的 3 条边都有相邻三角形。

对于边界三角形,把具有相邻三角形的边称为内边,不具有相邻三角形的边称为外边。图 4 是三角形划分的分类说明示例。

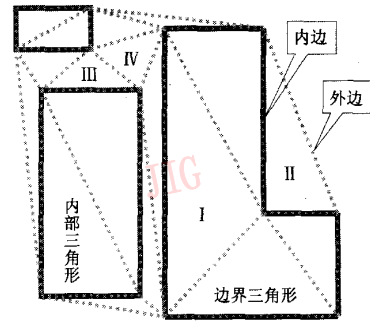


图 4 三角形分类说明
Fig. 4 Illustration of triangle classification

3.3 基于 Delaunay 三角划分的 ABTM 层次定义

从三角形的定义可以看出,三角形是建筑物合并算法中最基本的综合单元。把 Agent 概念引入到综合中来,并以 Delaunay 三角形为基础,一步步进行由微观到宏观的归纳,可以把 Agent 的结构分 3 层来定义,即 Agent 的单元结构、个体结构和群体结构。

Agent 群:地图目标中的一个聚类被称为一个 Agent 群。

Agent 个体:一个地图目标被称为一个 Agent 个体。

Agent 单元:组成 Agent 群的一个三角面被称为一个 Agent 单元。

从上述定义可以看出,Agent 群、Agent 个体和 Agent 单元之间存在着明显的隶属关系,即多个 Agent 单元组成一个 Agent 个体,而多个 Agent 个体和 Agent 单元组成了一个 Agent 群。其层次关系可用图 5 表示。

图 5 中的根节点为一个 Agent 群,相对应的是一个面多边形集合,如图 5 中的 a_1 ,这是一个聚类,而图 5 中的 a_2 则是经过了约束 Delaunay 三角剖分后的结果。一个 Agent 群可分为多个 Agent 个体,如

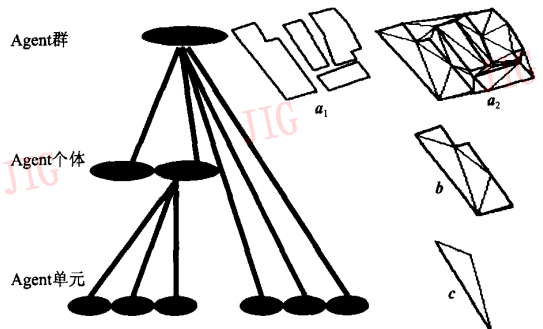


图 5 Agent 的层次结构示意图
Fig. 5 Layer structure of Agent

一个多边形面状目标图 5 中的 b , 而一个 Agent 个体又被划分为多个 Agent 单元, 图 5 中的 c 即是图 5 中的 b 中的一个三角形。

3.4 基于 ABTM 层次概念的 Agent 数据结构

根据 ABTM 的层次概念, 可以定义 Agent 的层次结构, Agent 群结构、个体结构和单元结构如下:

```
class GMC_AgentGroup
{
private:
    GMF_GetConnectedTris(); //获取其相关的三角形信息
    GMF_GetClusteringInfo(); //获取其被聚类信息
    GMF_GetConnectedSide(); //获取目标间相连的边界线
    GMF_GetSideInfo(); //获取其边界信息
    GMF_Evaluate(); //评估其需要综合的程度
    GMF_GetResult(); //得到合并后的面
    .....
protected:
    GMF_ExecuteListen(); //自动执行 Agent 实时探测的信息
    GMF_Kill(); //注销自己(生命周期强行结束)
public:
    GMF_Start(); //Agent 生命周期开始
    GMF_End(); //Agent 生命周期结束
    GMF_Listen(); //Agent 侦听外界变化
    GMF_Communicate((void) *); //Agent 群与外界交互
};

class GMC_AgentObj
{
private:
    GMF_GetConnectedTris(); //获取其相关的三角形
    GMF_GetTopologicInfo(); //获取其相邻目标的信息
    GMF_GetClusteringInfo(); //获取其被聚类信息
    GMF_GetSideInfo(); //获取其边界信息
    GMF_Evaluate(); //评估其是否能被合并
    .....
protected:
    GMF_ExecuteListen(); //自动执行 Agent 实时探测的信息
    GMF_Kill(); //注销自己(生命周期强行结束)
public:
    GMF_Start(); //Agent 生命周期开始
    GMF_End(); //Agent 生命周期结束
    GMF_Listen(); //Agent 侦听外界变化
    GMF_Communicate((void) *); //Agent 个体与外界交互
};

class GMC_AgentUnit
{
```

```
private:
    GMF_GetClassInfo(); //获取其分类信息(属于哪类三角形)
    GMF_GetTopologicInfo(); //获取其相邻三角形的信息
    GMF_GetControlArea(); //获取其相邻的所有三角形所组成的区域
    GMF_GetImportance(); //获取其自身的重要性
    GMF_GetClusteringInfo(); //获取其被聚类信息
    GMF_Evaluate(); //判断三角形的种类
    .....
protected:
    GMF_ExecuteListen(); //自动执行 Agent 实时探测的信息
    GMF_Kill(); //注销自己(生命周期强行结束)
public:
    GMF_Start(); //Agent 生命周期开始
    GMF_End(); //Agent 生命周期结束
    GMF_Listen(); //Agent 侦听外界变化
    GMF_Communicate((void) *); //Agent 单元与外界交互
};
```

地图数据输入后, 首先进行按道路和河流为约束条件的聚类分析, 把地图数据聚成若干个类; 然后把每个类的数据作为一个 Agent 群来对待, 调用接口函数 GMF_Start() 开始运行 Agent, 该接口调用 GMC_AgentGroup 类中的私有函数来获取 Agent 群进行面状目标合并所必需的综合信息, 同时 GMF_Listen() 接口函数开始监听环境的变化与需求, 并调用 GMF_ExecuteListen() 函数执行 Agent 群的动作, 而 GMF_Communicate() 则负责同其他 Agent 或外界环境的交流与沟通。如果发现 GMF_Evaluate() 函数提出该数据不满足面合并的综合条件, 则用 GMF_Kill() 强行结束生命周期, 否则在所有功能完成后运行 GMF_End() 结束 Agent 的生命周期。

从 Agent 群结构可以看出, 该结构具有生命周期开始和结束、自主运行、主动侦听外界需求与变化、并作出相应动作、与外界交互等功能, 而其私有函数则主要用于获取 Agent 运行所必需的生存环境信息。

一个 Agent 群体对应于多个面目标, 而一个 Agent 个体对应于一个面目标, 因此一个 Agent 群包含多个 Agent 个体。当 GMC_AgentGroup.GMF_Start() 开始并初始化完成后, 所有 Agent 个体的 GMF_Start() 接口函数被启动, 该接口调用 GMC_AgentObj 类中的私有函数以获取该层次 Agent 运行所必需的环境信息, 同时 GMF_Listen() 函数开始监听环境的变化

与需求,并调用 GMF_ExecuteListen() 函数执行 Agent 的请求和反应,而 GMF_Communicate() 则负责同其他 Agent 或外界环境的交流与沟通。如果发现 GMF_Evaluate() 提出不满足面合并的综合条件(比如该 Agent 个体处于中间位置,不拥有边界三角形等),则用 GMF_Kill() 强行结束生命周期,否则在所有功能完成后运行 GMF_End() 结束 Agent 的生命周期。

一个 Agent 个体对应一个面目标,一个 Agent 单元对应于一个三角面,因此一个 Agent 个体包含多个 Agent 单元。当 GMC_AgentObj. GMF_Start() 开始并初始化完成后,所有 Agent 单元的 MGF_Start() 接口函数被启动,该接口调用 GMC_AgentUnit 类中的私有函数以获取该层次 Agent 运行所必要的环境信息,同时 GMF_Listen() 函数开始监听环境的变化与需求,并调用 GMF_ExecuteListen() 函数执行 Agent 的请求和反应,而 GMF_Communicate() 则负责同其他 Agent 或外界环境的交流与沟通。如果发现 GMF_Evaluate() 提出不满足面合并的综合条件(比如该 Agent 单元为 I、II 类三角形或内部三角形等),则用 GMF_Kill() 强行结束生命周期,否则在所有功能完成后运行 GMF_End() 结束 Agent 的生命

周期。

3.5 基于 ABTM 模型的建筑物自动合并算法步骤

在定义了 Agent 的层次概念后,原本采用 TIN 技术进行面目标的合并算法就可以拓宽为基于 ABTM 的算法。由于 Agent 具有生命周期,且可以随时终止,而生命周期的结束意味着该 Agent 所对应的地图目标被删除,从而对面目标的合并演化为对 Agent 单元的处理过程。

首先,对原始数据进行聚类,生成满足要求的数据类。然后,对数据类进行约束 Delaunay 三角构网,按照 Agent 的层次结构把相应的 Delaunay 三角网数据信息赋予 Agent。即数据类信息赋予 Agent 群,数据类中的每一个面目标赋予每个 Agent 个体,而每个三角面信息赋予 Agent 单元。设原始数据被聚为 n 个类,则相应地有 n 个 Agent 群产生。设初始时 $i=0$ 执行以下步骤:

(1) 对第 i 个 Agent 群进行综合需求情况的自动评估,判断其是否需要进行综合(图 6(a)),如果需要,则进入第 2 步,否则结束该 Agent 的生命周期, $i=i+1$,重新执行第 1 步。如果 $i=n$,则执行第 5 步。

判断 Agent 群是否需要综合,主要察看其属性

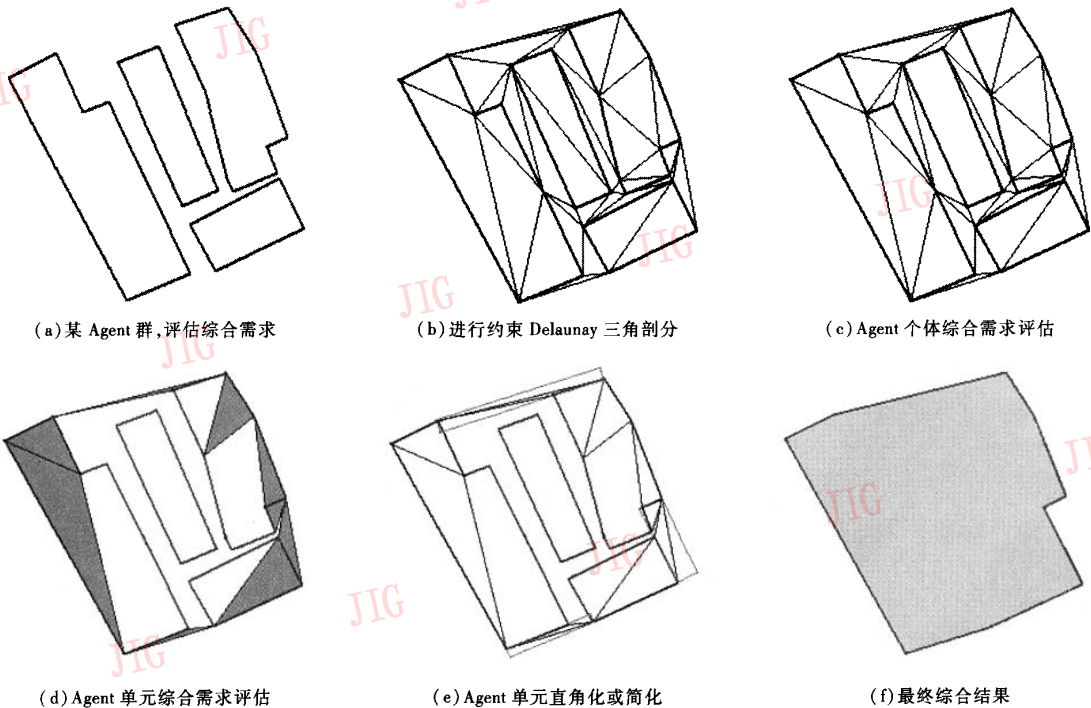


图 6 基于 ABTM 算法进行城市建筑物合并过程分解
Fig. 6 The process of ABTM algorithm used in city building

条件、精度条件、面积、子 Agent 个数和几何距离条件等。例如针对 1:5 万数据,可以定义类似表 2 的

综合约束条件。如果不能满足表 2 的约束条件,则表明该 Agent 不具有综合的条件。

表 2 综合的约束条件

Tab.2 The constrain of generalization

约束条件	综合目标
属性条件约束	是编图规范允许综合的目标
精度条件约束	综合前后精度差异不能过大
面积约束	目标应该具有最小尺寸
子 Agent 个数	子 Agent 个数应该大于 1 个
几何距离约束	目标间的最大距离不应过大
拓扑条件约束	目标间不能相交
	$m_bCanGeneralized = TRUE$
	Hausdorff distance < 20m (from initial position)
	$m_Area > 300m^2$
	$m_SubAgentNum > 1$
	$M_MaxLength_of_edgt < 20m$
	$Ojb1 \cap Ojb2 = \emptyset$

(2) 第 i 个 Agent 群中的每个 Agent 个体自动评估综合需求情况,判断其是否需要参与综合(图 6(c))。把满足综合条件的 Agent 个体保留下来,把不满足综合条件的 Agent 个体结束生命周期,进入第 3 步。

判断 Agent 个体是否需要参与综合,主要察看其是否拥有边界三角形。没有边界三角形的 Agent 个体处于综合框架的内部,并不对综合结果造成影响,可结束该 Agent 的生命周期,以减少算法的复杂度。

(3) 每个 Agent 个体中的 Agent 单元自动评估综合需求情况,判断其是否参与综合(图 6(d))。把需要参与综合的 Agent 单元保留下来,而不满足综合条件的 Agent 单元结束生命周期,进入第 4 步。

判断 Agent 单元是否需要参与综合,主要察看其属于哪一类三角形(表 3)。

表 3 Agent 单元自动评估

Tab.3 Auto-evaluate of Agent cell

Agent 单元的类型	是否结束生命周期
I 类三角形 + 边界三角形	否
I 类三角形 + 内部三角形	是
II 类三角形	否
III 类三角形 + 边界三角形	否
III 类三角形 + 内部三角形	是
IV 类三角形 + 边界三角形	否
IV 类三角形 + 内部三角形	是

(4) 针对每个仍具有生命的 Agent 单元,进行三角形的简化和直角化(图 6(e))。最后把所有 Agent 单元的外边有序连接,从而得到完整的综合结果(图 6(f))。这样,第 i 个 Agent 群综合完成, $i = i + 1$,重新执行第 1 步。如 $i = n$,则执行第 5 步。

具体来说,对属于 IV 类或 II 类的边界三角形,直接采用三角形的外边作为合并后目标轮廓的一部分,即 Agent 个体之间直接用三角形的外边相连,作

为新目标的轮廓。对属于 III 类的边界三角形,需要进行三角形的形态分析,以决定采用简化还是直角化处理。

而 III 类三角形的 3 个顶点中有两个属于同一个 Agent 个体,且把这两个顶点的连线称为基准边,则以基准边为矩形的一边方向,做三角形的最小外接矩形。设矩形的长为 l , 宽为 w , 则矩形的形状因子为

$$f_{shap} = \alpha \times (l/w) \quad (3)$$

式中 f_{shap} 为矩形的形状因子, α 为权重,可以根据不同要求设置不同的值,其值越大,则直角化的程度越小。把矩形的形状因子作为判断三角形进行简化或直角化的阈值。如采用如下阈值进行判断:

$$f_{shap} = \begin{cases} > 5 & \text{简化} \\ \leq 5 & \text{直角化} \end{cases} \quad (4)$$

(α 取值为 1)

所谓的直角化指用矩形的直角边代替原来三角形的边,并把直角边作为合并后面轮廓的一部分(图 7),而所谓的简化指直接采用矩形的对角线来替代三角形,并作为合并后面轮廓的一部分(图 8)。



图 7 三角形的直角化
Fig.7 Squaring triangle

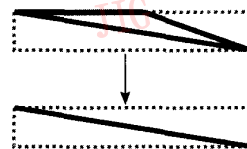


图 8 三角形的简化
Fig.8 Simplify triangle

矩形的形状因子 f_{shap} 为矩形的长和宽的相对比

例,故与制图综合比例尺无关。

(5)结束。

4 实验及结果分析

本算法的基础数据比例尺为 1:50000,目标比

例尺为 1:100000。图中有居民地及附属设施、陆地交通、水域/陆地等 3 层图形要素。按本算法实现以后的合并效果如图 9,其综合指标见表 4。图 10 是对更大范围的 1:10000 大比例尺数据综合 1:25000 图的结果。图中主要采用道路网对城市建筑物进行约束,利用本算法进行合并。



(a)综合前



(b)综合后

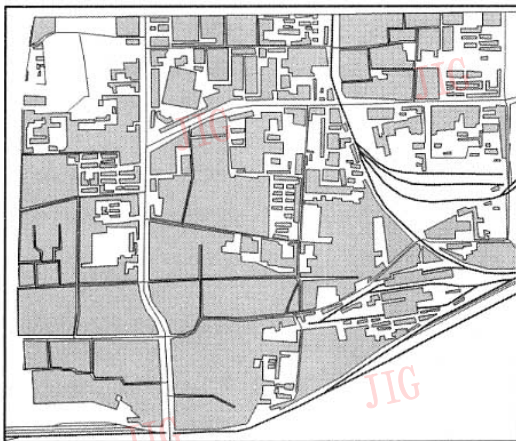
图 9 基于 ABTM 算法的建筑物合并示例(1:50000 数据)

Fig.9 An example of merging city buildings with ABTM algorithm(data of 50000)

表 4 综合指标

Tab.4 Generalization targets

综合指标	数量	综合指标	数量
面状建筑物个数	48 个	三角形直角化次数	7 次
一次聚类个数(区域意义上的聚类)	15 个	三角形简化次数	21 次
二次聚类个数(几何意义上的聚类)	30 个	合并后建筑物个数	30 个
Agent 群/ Agent 个体/Agent 单元	30/48/338 个	综合前后建筑物面积比	13:14
参加合并的有效 Agent 群	13 个	算法速度	<1/100s
合并次数	13 次		



(a)综合前



(b)综合后

图 10 基于 ABTM 算法对更大区域建筑物合并示例(1:10000 数据)

Fig.10 An other example of merging more city buildings with ABTM algorithm(data of 10000)

从表 4 的综合指标可以看出,该算法在综合过程中第 1 次聚类只是依据水系和陆地交通等线划要素进行区域分割,而第 2 次聚类才是真正几何意义上的聚类分析,并直接影响着参加合并的有效 Agent 群的个数,而参加合并的有效 Agent 群直接影响合并的次数和综合后建筑物的数量以及综合前后建筑物面积比。而基于 ABTM 的合并模型对合并后建筑物轮廓的形状、三角形直角化和简化的次数产生直接影响。同时合理利用聚类和 ABTM 多线程技术,可以大大地提高算法的速度。

5 结 论

目前面状建筑物合并算法主要有两类:一类是基于矢量数据而言的,典型算法比如基于 TIN 的面要素合并模型;另一类是基于栅格数据而言的,典型算法比如基于数学形态学的算法。基于 TIN 的面要素合并模型主要采用基于三角形的纯数学算法,因此智能化程度不高,一方面需要进行大量的计算,另一方面只能进行手工交互式合并操作,自动化程度低。而基于数学形态学的方法针对图像采用了膨胀、腐蚀、开运算、闭运算、击中、薄化和厚化等运算,算法速度与区域大小有关,与区域内目标数量无关,但对处理大区域图像,计算量成倍增长,严重影响算法的速度;同时,基于数学形态学的栅格方法对多边形边缘处理后存在毛刺现象,其对图形处理质量依赖于硬件设备,故对建筑物合并后的图形质量会造成一定的影响。而基于 ABTM 的城市建筑物算法考虑到了上述算法的不足,采用 Agent 技术和 TIN 技术相结合,使得算法的智能性得到很大提高;采用区域聚类算法,对大区域、大面积和海量数据进行区域划分,然后对每个局部区域进行小数据量的运算,使得运算时间由对全部数据一次性处理的指数级增长转变为对每个数据块处理时间的简单相加,算法速度极大地提高,并且,本算法能够对数据进行自动聚类,并自动进行合并操作,实现了算法的自动化;同时,本算法采用矢量数据形式,算法执行后的图形质量不受硬设备的影响。

通过以上实验及其分析,可以看出基于 ABTM 的面合并算法集成了 TIN 技术和 Agent 技术的特长,相比之下,具有以下优点:

(1)用约束 Delaunay 三角剖分技术,使得城市建筑物目标之间拓扑关系的探测更为方便,空间处

理能力大大加强。而 Agent 技术运用到 TIN 中来,无疑给其带来了更大的灵活性和功能上的增强。

(2)Agent 技术的自主性、反应性、能动性、通信性,以及 Agent 的生命周期等特点,使得本算法的实现更为直观,算法的智能性大大提高。ABTM 算法首次赋予了地图综合算法以生命实体的特征,对地图综合沿着智能化、自动化道路前进开辟了新的方向。

(3)对海量数据进行综合和多比例尺进行表达的难度之一是速度问题。聚类方法的使用,使得对海量数据的处理转化为对分块数据的处理,处理速度大大加快;同时,由于 Agent 技术采用了多线程并行处理的特点,也大大提高了算法的处理速度。

(4)采用聚类分析和约束条件分析,使得综合结果的正确率得到了较大的提高。对边界三角形的简化和直角化处理,充分考虑了大比例尺城市建筑物的特点,保持了城市的结构特征。

当然,本算法也具有一些考虑不周之处。比如在进行数据聚类的时候,只考虑了道路和河流作为约束条件,而其他地理要素比如植被、境界与政区、工农业与社会文化设施等对城市建筑物的图形合并也具有一定的影响,本文中并没有把这些因素考虑到。另外,对大比例尺中存在较多的建筑物中空空白区域的处理,比如四合院中的空地等,也没有考虑。这些是本算法今后需要改进的地方。

参考文献 (References)

- 1 Wang J Y. Theory of general map generalization [M]. Beijing: Mapping Publishing Company, 1992:1~20. [王家耀. 普通地图制图综合原理[M]. 北京:测绘出版社,1992:1~20.]
- 2 Wu H H. About the city building auto-generalization in GIS condition [J]. Transaction of Wuhan Surveying and Mapping University, 2000, 25(3):196~202. [毋河海. GIS 环境下城市平面图形的自动综合问题[J]. 武汉测绘科技大学学报,2000, 25(3):196~202.]
- 3 Wu F. Multiple Representation and auto-generalization for special data[M]. Beijing: Publishing Company of PLA, 2003:4~14. [武芳. 空间数据的多尺度表达与自动综合[M]. 北京:解放军出版社,2003:4~14.]
- 4 Qian H Z, Wang J Y. Agent technology of special information system [J]. Mapping Science, 2004, 29(1):10~12. [钱海忠,王家耀. 空间信息系统中的 Agent 技术[J]. 测绘科学,2004, 29(1):10~12.]
- 5 Alessandro Cecconi. Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping [D]. Zurich: University Zurich,2003.
- 6 Paul Hardy, Melanie Hayles, Patrick Revell. Clarity-A New

- Environment for Generalization Using Agents, Java, XML, and Topology [EB/OL]. [http://www. geo. unizh. ch/ICA/docs/paris2003/papers03. html](http://www.geo.unizh.ch/ICA/docs/paris2003/papers03.html), 2003-04-14
- 7 Ai T H. Research on Data Modal and Method Based on City Map Database' Generalization[D]. WuHan: WuHan Survey and Mapping University, 2000. [艾廷华. 城市地图数据库综合的支撑数据模型与方法的研究[D]. 武汉:武汉测绘科技大学, 2000.]
- 8 Guo R Z, Ai T H. Merging and simplifying city region buildings in map generalization[J]. Transaction of WuHan Survey and Mapping University, 2000, **25**(1):25 ~ 30. [郭仁忠, 艾廷华. 制图综合中建筑物多边形的合并与化简[J]. 武汉测绘科技大学学报, 2000, **25**(1):25 ~ 30.]
- 9 Qian H Z, Wu F. Region's merging method based on Delaunay relationships[J]. Transaction of Surveying and Mapping Institute, 2001, **18**(3):10 ~ 13. [钱海忠, 武芳. 基于 Delaunay 三角关系的面状要素合并方法[J]. 测绘学院学报, 2001, **18**(3):10 ~ 13.]
- 10 Liu X J, Gong J Y. A Delaunay's modified dividing algorithm with constrained data condition [J]. Transaction of Surveying and Mapping, 2001, **30**(1):82 ~ 88. [刘学军, 龚健雅. 约束数据域的 Delaunay 三角剖分与修改算法[J]. 测绘学报, 2001, **30**(1):82 ~ 88.]