引文格式：赵龙, 罗永亮. 一种基于转移边折叠的TIN模型简化方法研究[J]. 地理空间信息, 2023, 21(4): 1-4. doi:10. 3969/j.issn. 1672-4623.2023.04.001

一种基于转移边折叠的TIN模型简化方法研究

赵　龙[1]，罗永亮[2]
（1. 中铁二院工程集团有限责任公司，四川 成都 610031；2. 四川旷谷信息工程有限公司，四川 成都 610031）

摘　要：边折叠算法是简化TIN模型的主要方法之一。依据边折叠算法的特点，深入分析边折叠算法的缺陷及其产生的问题，并总结出这些问题产生的原因，通过预折叠的方式找出并标记模型中的特殊边。当对特殊边折叠的时候，通过转移折叠目标的方式来简化模型，并利用三角形品质量化简化质量。实验证明，这种转移边折叠的方式不仅可以提高简化后模型的三角形品质，还可以减弱简化模型的纹理变形和避免模型法向量计算错误，缩小简化模型和原始模型的视觉差距。

关键词：边折叠算法；TIN；三角形品质；转移边折叠算法
中图分类号：P208　　**文献标志码：**B　　**文章编号：**1672-4623（2023）04-0001-04

Research on TIN Simplification Method Based on Transfer Edge Collapse Algorithm

ZHAO Long[1], LUO Yongliang[2]
(1. China Railway Eryuan Engineering Group Co., Ltd., Chengdu 610031, China;
2. Sichuan Crungoo Information Engineering Co., Ltd., Chengdu 610031, China)

Abstract: Edge collapse algorithm is one of the main methods to simplify TIN model. According to the characteristics of edge collapse algorithm, we deeply analyzed the defects and problems of edge collapse algorithm, summarized the causes of these problems, and found and marked special edges by the way of pre-collapsing. When collapsing special edges, we simplify the model by transferring edges, and measured the quantization by triangular quality. Experimental result shows that thisalgorithm can not only improve triangular quality, but also reduce the texture distortion, avoid errors of normal vector calculation, and reduce the optical difference between simplified models and original models.

Key words: edge collapse algorithm, TIN, triangular quality, transfer edge collapse algorithm

目前针对TIN模型的简化方法可分为顶点聚类法[1-2]、区域合并法[3]、几何元素消除法。几何元素消除法又可分为顶点删除法[4]、边折叠法[5]和面折叠法[6]，而边折叠法由于其算法简单，效率高而被广泛使用。该算法也是目前模型简化算法的主流算法，针对边折叠的改进算法包括基于Voronoi图重构的UDSM边折叠简化算法[7]，顾及角度误差的边折叠算法[8]和基于CPU并行加速的边折叠算法[9-10]，基于Memoryless算法的边折叠简化算法[11]，这些算法大部分是针对代价函数、新定点的位置和算法的效率展开的，对于简化后三角形的质量却少有提及。本文根据边折叠算法存在的问题，提出了一种基于转移边折叠的TIN模型简化算法，该算法可规避在折叠过程中生成突兀边、重复边和重复三角形，避免了对边界的简化。通过实验，与传统的边折叠算法相比，该算法可提高三角形的品质。

1　边折叠算法

1.1　边折叠算法简介

边折叠算法是将原始地形网格中的每条边作为最小简化单元的一种算法，该算法将要折叠的边化简为一个点，并将与该边共顶点的三角形拉伸到一个点上，如图1所示。

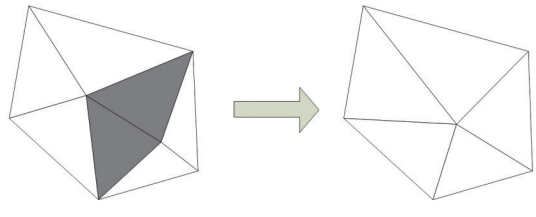


图1　边折叠算法

从图1可知，当对某条边进行折叠的时候，则与该边相邻的三角形均被拉伸，这些相邻的三角形组成了边的变化影响范围U。在U之内，假设将边e折叠

收稿日期：2022-04-03；修回日期：2023-02-22。
第一作者简介：赵龙（1983—），硕士研究生，高级工程师，研究方向为三维GIS和BIM融合和铁路工程测量方向，E-mail：305024828@qq.com。

到点 $\boldsymbol{V}=[\boldsymbol{v}_x, \boldsymbol{v}_y, \boldsymbol{v}_z, 1]^{\mathrm{T}}$，则折叠该边代价可以由点 $\boldsymbol{V}$ 到 U 集合中所有三角形的距离的平方之和来表示，若三角形所在的平面 p 的方程为 $\mathrm{a}x+\mathrm{b}y+\mathrm{c}z+\mathrm{d}=0$，则点 $\boldsymbol{V}$ 到平面 p 的距离为：

$$\Delta(\boldsymbol{V},\mathrm{p})=\frac{\left|\mathrm{a}\boldsymbol{v}_x+\mathrm{b}\boldsymbol{v}_y+\mathrm{c}\boldsymbol{v}_z+\mathrm{d}\right|}{\mathrm{a}^2+\mathrm{b}^2+\mathrm{c}^2} \tag{1}$$

则折叠到 $\boldsymbol{V}$ 点的代价为：

$$\Delta(\boldsymbol{V})=\sum_{j}^{n}\left(\Delta\left(\boldsymbol{V},\mathrm{p}_j\right)\right)^2 \tag{2}$$

式中，$\boldsymbol{V}$ 为矢量；$\mathrm{p}_j\in U$，令 $\mathrm{a}_j^{\,2}+\mathrm{b}_j^{\,2}+\mathrm{c}_j^{\,2}=1$，其中 a_j，b_j，c_j 为平面方程 p_j 上 x,y,z 的系数，则可以将公式（1）写成向量乘积的形式为：

$$\Delta(\boldsymbol{V})=\sum_{\bar{\mathrm{p}}_j}^{n}\left(\mathrm{p}_j^{\mathrm{T}}\boldsymbol{V}\right)^2=\boldsymbol{V}^{\mathrm{T}}Q\boldsymbol{V} \tag{3}$$

其中：

$$\mathrm{p}_j=[\mathrm{a}_j,\mathrm{b}_j,\mathrm{c}_j,\mathrm{d}_j]$$

$$Q=\sum_{\mathrm{p}_j}^{n}\begin{bmatrix}\mathrm{a}_j^{\,2} & \mathrm{a}_j\mathrm{b}_j & \mathrm{a}_j\mathrm{c}_j & \mathrm{a}_j\mathrm{d}_j\\ \mathrm{a}_j\mathrm{b}_j & \mathrm{b}_j^{\,2} & \mathrm{b}_j\mathrm{c}_j & \mathrm{b}_j\mathrm{d}_j\\ \mathrm{a}_j\mathrm{c}_j & \mathrm{b}_j\mathrm{c}_j & \mathrm{c}_j^{\,2} & \mathrm{c}_j\mathrm{d}_j\\ \mathrm{a}_j\mathrm{d}_j & \mathrm{b}_j\mathrm{d}_j & \mathrm{c}_j\mathrm{d}_j & \mathrm{d}_j^{\,2}\end{bmatrix}=\begin{bmatrix}\mathrm{q}_{11} & \mathrm{q}_{12} & \mathrm{q}_{13} & \mathrm{q}_{14}\\ \mathrm{q}_{21} & \mathrm{q}_{22} & \mathrm{q}_{23} & \mathrm{q}_{24}\\ \mathrm{q}_{31} & \mathrm{q}_{32} & \mathrm{q}_{33} & \mathrm{q}_{34}\\ \mathrm{q}_{41} & \mathrm{q}_{42} & \mathrm{q}_{43} & \mathrm{q}_{44}\end{bmatrix}$$

将公式（3）展开得到公式（4）。

$$\Delta(\boldsymbol{V})=\boldsymbol{V}^{\mathrm{T}}Q\boldsymbol{V}=\mathrm{q}_{11}\boldsymbol{V}_x^2+2\mathrm{q}_{12}\boldsymbol{V}_x\boldsymbol{V}_y+2\mathrm{q}_{13}\boldsymbol{V}_x\boldsymbol{V}_z+2\mathrm{q}_{14}\boldsymbol{V}_x+\mathrm{q}_{22}\boldsymbol{V}_y^2+2\mathrm{q}_{23}\boldsymbol{V}_y\boldsymbol{V}_z+2\mathrm{q}_{24}\boldsymbol{V}_y+\mathrm{q}_{33}\boldsymbol{V}_z^2+2\mathrm{q}_{34}\boldsymbol{V}_z+\mathrm{q}_{44} \tag{4}$$

在对TIN模型进行简化时，利用公式（4）计算TIN中每条边的代价，然后利用该代价为所有边排序，代价越低的边优先简化，代价高的边则保留。

1.2 $\boldsymbol{V}$ 的计算方法

$\boldsymbol{V}$ 的取值应该使得公式（5）最小，为此，可以对公式（5）中的 $\boldsymbol{V}_x$，$\boldsymbol{V}_y$，$\boldsymbol{V}_z$ 求偏导数，并使得导数为0，即：

$$\frac{\partial\Delta(\boldsymbol{V})}{\partial\boldsymbol{V}_x}=\frac{\partial\Delta(\boldsymbol{V})}{\partial\boldsymbol{V}_y}=\frac{\partial\Delta(\boldsymbol{V})}{\partial\boldsymbol{V}_z}=0$$

列出线形方程组得：

$$\begin{bmatrix}\mathrm{q}_{i11} & \mathrm{q}_{i12} & \mathrm{q}_{i13} & \mathrm{q}_{i14}\\ \mathrm{q}_{i21} & \mathrm{q}_{i22} & \mathrm{q}_{i23} & \mathrm{q}_{i24}\\ \mathrm{q}_{i31} & \mathrm{q}_{i32} & \mathrm{q}_{i33} & \mathrm{q}_{i34}\\ 0 & 0 & 0 & 1\end{bmatrix}\boldsymbol{V}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

则：

$$\boldsymbol{V}=\begin{bmatrix}\mathrm{q}_{i11} & \mathrm{q}_{i12} & \mathrm{q}_{i13} & \mathrm{q}_{i14}\\ \mathrm{q}_{i21} & \mathrm{q}_{i22} & \mathrm{q}_{i23} & \mathrm{q}_{i24}\\ \mathrm{q}_{i31} & \mathrm{q}_{i32} & \mathrm{q}_{i33} & \mathrm{q}_{i34}\\ 0 & 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \tag{5}$$

1.3 边折叠算法存在的问题

1）$\boldsymbol{V}$ 落在边的影响范围之外。如图2所示，如果折叠后的点 $\boldsymbol{V}$ 落在边e的影响范围之外，则该折叠操作就会破坏三角形网格的拓扑结构，造成简化质量低下，影响简化后模型的视觉效果。

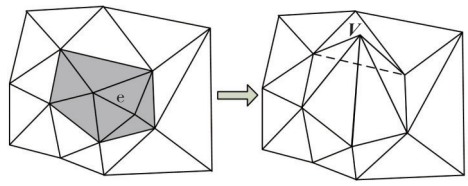


图2 $\boldsymbol{V}$ 点落在边的影响范围之外

2）突兀边的产生。当把边ab折叠到a点的时候，会产生ad、ac和cd 3条突兀边，突兀边的产生将会影响模型简化的效果，造成模型法向量紊乱，因此在简化过程中应尽量避免突兀边的产生，如图3所示。

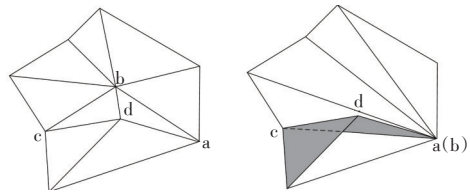


图3 产生突兀边

3）产生重合边和三角形。图4中说明了几种会生成重合边和三角形的情况，图4a中三角形f10和三角形f3有一条共同边，当将边ab折叠时，三角形f3和f10就会重合，继而生成若干重合的三角形和边。图4b中三角形f7和f6也有一条共同的边，当将边ab折叠时，三角形f7和f6也会重合，继而生成若干重合的三角形和边。图4c中综合了2种情况，即三角形f10和三角形f3有共同边，三角形f7和f6也有共同边。当将边ab折叠时，三角形f3和f10以及f6和f7均重合，继而生成若干重合的三角形和边。与突兀边类似，当在简化过程中产生了重合边或三角形的时候，也会影响模型的简化效果，造成模型的法向量紊乱，在简化过程中应该尽量避免此种现象的产生。

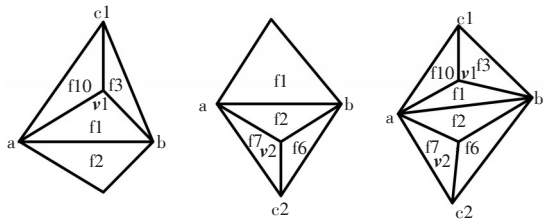
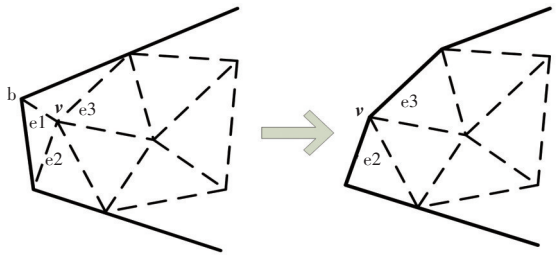


图4 产生重合边和三角形

1.4　边界处理

在对TIN模型进行简化时，通常会遇到边界问题，边界问题处理得是否得当会直接影响到整个模型的简化效果，很多情况下要求对模型简化要保持其边界轮廓的不变性，因此对边界问题的处理也是模型简化中的一个重要问题。

边界问题的处理首先要确定哪些边是边界边，可以根据当前边的属性来确定边界边，如果一条边仅有左边三角形或者仅有右边三角形，则说明这条边为边界边。除了边界边的简化对模型的边界有影响外，与边界边共顶点的非边界边的简化也会造成模型边界轮廓变形，图5说明了这种情况，其中粗线条表示模型的边界边，虚线表示非边界边。



a 折叠前　　b 折叠后

图5　非边界边折叠造成模型边界变形

从图5a中可知，e1不是边界边，但它与边界边相邻，当将e1折叠到 $\boldsymbol{v}$ 点时，其模型在此处的边界也更改为e2和e3。如图5b所示，在对这些边做处理时，应当尽量将折叠边的新位置设置在边界上，如图5a所示，应将边e1折叠到b点，而不能将其折叠到 $\boldsymbol{v}$ 点，这样就可以保证模型的边界特性。

2　转移边折叠算法

常规的边折叠算法无法解决 $\boldsymbol{V}$ 落在边的影响范围之外、突兀边及重合边和重合三角形的问题，为此，本文提出了一种转移边折叠的算法，转移边折叠的方法是将TIN内所有的边分为常规边和特殊边。对于常规边采用常规的边折叠算法，而特殊边则采用转移边折叠算法，转移边折叠算法是将应折边e转移为其相邻的边，并对相邻的边进行折叠的一种算法，与传统的边折叠算法相比，转移边折叠算法多了2个步骤，一是寻找特殊边，二是转移折叠目标。

2.1　特殊边的检测

特殊边包括在边界三角形上的边和折叠后出现错误的内部边检测，内部边检测又可分为突兀边检测和重复边检测，内部边的检测过程又称为预折叠过程，即从折叠后的结果判断边是否为特殊边，边界边和内部边的检测方法不同，具体方法如下：

1）边界边检测。边界边与其他边的区别在于该边只有一侧有三角形，而其他边两侧都有三角形，可通过该特性区分边界边和其他边，边界边所在三角形为边界三角形。根据特殊边的定义范围，边界三角形的其他两个边也为特殊边。

2）内部边的检测。对于TIN三角形网格而言，折叠前与折叠后的三角形法向量应相差很小，如果折叠后的三角形法向量翻转（大于90°），如图3中的Δbcd，则说明本次折叠会产生突兀边或重合三角形，若折叠后法向量无法计算，则说明折叠后部分三角形重合成了一条线，将会产生重合的边，从而判断边e为特殊边。其具体步骤如下：

1）利用常规的边折叠算法对边e折叠，并将e折叠到 $\boldsymbol{V}$ 点。

2）对e影响范围内的三角形重新计算法向量，并于折叠前的法向量对比。

3）根据对比结果判断该边是否为特殊边。

从上面分析表明，计算三角形法相量成为监测突兀边的最主要的环节，在TIN模型的三角形网络中，其三角形中边的位置是按照逆时针方向排列的，设△ABC为折叠边e影响范围内的三角形，△ABC按逆时针方向构建法向量，若将e折叠到点 $\boldsymbol{V}$，为了不破坏△ABC的拓扑结构，可将 $\boldsymbol{V}$ 点的坐标赋值给C点，如图6所示。

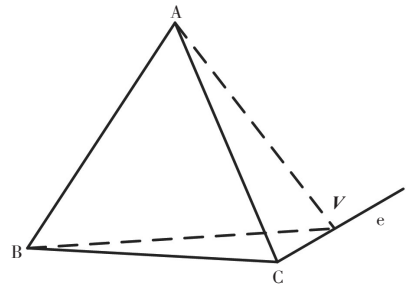


图6　三角形法向量的计算

则△ABC和△AB $\boldsymbol{V}$ 的法向量可由公式（6）计算得到：

$$\begin{aligned}\boldsymbol{N}_1&=\mathrm{Nor}(\overrightarrow{\mathrm{BA}})\times\mathrm{Nor}(\overrightarrow{\mathrm{BC}})\\ \boldsymbol{N}_2&=\mathrm{Nor}(\overrightarrow{\mathrm{BA}})\times\mathrm{Nor}(\overrightarrow{\mathrm{B}\boldsymbol{V}})\end{aligned}\tag{6}$$

式中，$\boldsymbol{N}_1$ 为△ABC的法向量；$\boldsymbol{N}_2$ 为△AB $\boldsymbol{V}$ 的法向量；Nor为向量归一化函数。而：

$$\cos(\theta)=\boldsymbol{N}_1\cdot\boldsymbol{N}_2\tag{7}$$

式中，θ 为折叠前后两个三角形的夹角；则有：当 $\cos\theta>0$ 时，e为常规边；当 $\cos\theta\leqslant 0$ 时，e为特殊边。

2.2　转移折叠目标

当要简化特殊边的时候，需要转移折叠目标，折

叠目标的转移需要遵循3个原则：

1）转移的目标不能是特殊边。

2）转移目标应该为边e影响范围内的边集合U中一条边。

3）在边集合U中找到一个折叠代价最小的边作为转移目标。

2.3 算法流程

转移边折叠算法流程如图7所示，转移边折叠算法先利用公式（3）计算出每个边的折叠代价，并根据简化比例和折叠边代价筛选出待折叠边集合U。筛选的方法是首先根据折叠边代价为边排序（从小到大），得到排序集合S，再根据简化比例从S中获取U。对于U中的每条边e算出是否为特殊边，并将特殊边属性写入边的数据结构。若e不是特殊边，则直接利用边折叠算法将e折叠；若e是特殊边，则在e的影响范围内寻找转移目标ef，并对ef折叠，直到算法结束。

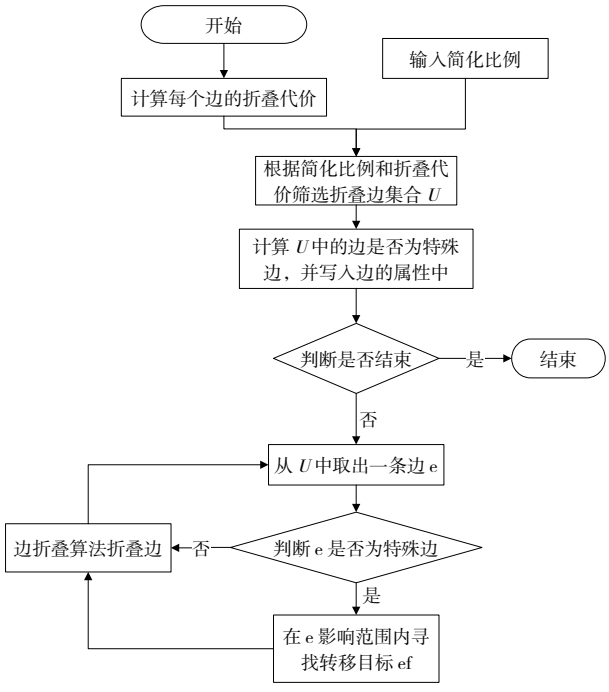


图7 转移边折叠算法步骤

3 实验分析

本节将从转移边折叠算法的简化质量方面进行实验分析，并引入三角形品质的概念量化算法的简化质量。一般来讲，三角形品质高则简化质量高，反之，简化质量低。三角形品质可通过公式（8）来计算。

$$Q=\frac{S}{L_{\max}} \tag{8}$$

式中，Q为三角形的品质；S为三角形的面积；$L_{\max}$为三角形的最长边。

三角形的品质用于描述三角形的尖度信息，三角形的尖度越大，三角形的品质越小。如果TIN模型经过简化后得到的三角形尖度大，则会影响TIN模型的光照计算，并使纹理失真，因此在简化过程中应尽量保证三角形的品质。由于转移边折叠算法加入了特殊边的判别，特殊边可防止三角形局部简化过大，有利于局部简化均匀，可大大提高模型的简化质量。

利用转移边折叠、Memoryless、UDSM和边折叠算法将三角形数量为116 983的TIN模型简化到50%时，各个三角形品质区间三角数量占比如表1所示。

表1 三角形品质区间占比表

三角形品质区间	转移边折叠/%	Memoryless算法/%	UDSM算法/%	边折叠/%
$Q>0.4$	12	5	4	0.5
$0.3<Q\leqslant 0.4$	20	15	12	15
$0.2<Q\leqslant 0.3$	62	51	46	21
$0.1<Q\leqslant 0.2$	5	12	18	40.5
$Q\leqslant 0.1$	1	17	20	23

从表1中可以看出，利用转移边折叠算法简化后的TIN模型较优，Memoryless算法和UDSM算法在三角形品质方面相当。

4 结 语

在边折叠算法中，由于折叠点的不可控性，导致在折叠后会出现突兀边、重复边和重复三角形，进而造成折叠后三角形的品质下降，纹理变形及光照计算错误等问题。为此，本文提出了一种解决这些问题的转移边折叠算法，与边折叠算法相比，用该方法简化后的TIN模型三角形品质更优，也可有效地防止纹理变形和光照计算错误等问题。后续还将继续优化该方法，通过理论和定量的方式研究该方法是如何降低纹理变形和避免光照计算错误的。

参考文献

[1] 潘正风,程效军. 数字地形测量学[M]. 武汉:武汉大学出版社,2015

[2] 栾婉娜,刘成. 基于逆Loop细分的半正则网格简化算法[J]. 图学学报,2020,41(6):980-986

[3] 曹卫群,鲍虎军,彭群生. 基于高斯球的近似共面合并层次细节模型[J]. 自动化学报,1999,25(1):1-8

[4] 刘钊,张涛. 利用顶点删除法简化道路三维地形模型[J]. 测绘通报,2014(8):62-64

[5] 陈雨,张建伟. 顾及局部特征的机场三维模型边折叠算法[J]. 现代计算机,2021,14(10):38-41

（下转第22页）