



## Tạp chí Khoa học Kỹ thuật Mỏ - Địa chất

Trang điện tử: <http://tapchi.humg.edu.vn/>



# Sử dụng cấu trúc cạnh kép (DCEL) để lưu trữ và xử lý một số thao tác biên tập mô hình mạng lưới tam giác không quy chuẩn (TIN)

Ngô Thị Liên <sup>1,\*</sup>, Trần Thùy Dương <sup>2</sup>, Lê Quang Hùng <sup>3</sup>

<sup>1</sup> Trung tâm Tin học Trắc địa và Bản đồ, Viện Khoa học Đo đạc và Bản đồ, Việt Nam

<sup>2</sup> Khoa Trắc địa - Bản đồ và QLĐĐ, Trường Đại học Mỏ - Địa chất, Việt Nam

<sup>3</sup> Công ty Cổ phần Công nghệ Tài nguyên - Môi trường và Vật liệu, Việt Nam

### THÔNG TIN BÀI BÁO

### TÓM TẮT

#### Quá trình:

Nhận bài 04/10/2016

Chấp nhận 15/11/2016

Đăng online 30/12/2016

#### Từ khóa:

Cấu trúc cạnh kép

Tam giác hóa

Thuật toán tăng dần

Hoán đổi tam giác

Trong việc xây dựng và xử lý các thao tác biên tập mô hình mạng lưới tam giác không quy chuẩn (TIN) có thể sử dụng các cấu trúc dữ liệu biểu diễn tam giác khác nhau, trong số đó có cấu trúc cạnh kép. Hiện nay mô hình mạng lưới tam giác thường được xử lý với số lượng tam giác rất lớn, vì vậy việc nghiên cứu và tìm ra cấu trúc dữ liệu phù hợp cho mô hình tam giác là cần thiết. Với mục đích đánh giá cấu trúc cạnh kép để ứng dụng trong mô hình mạng lưới tam giác, bài báo đã phân tích cấu trúc dữ liệu cạnh kép trong việc lưu trữ và xử lý một số thao tác biên tập mô hình TIN. Bài báo sử dụng phương pháp phân tích, so sánh cấu trúc cạnh kép với các cấu trúc dữ liệu khác và thực nghiệm lập trình sử dụng cấu trúc cạnh kép trong lưu trữ và xử lý mô hình TIN bằng ngôn ngữ Visual Basic 6.0. Bài báo đã đánh giá được những ưu, nhược điểm và đưa ra những điều chỉnh để cấu trúc cạnh kép phù hợp hơn trong việc lưu trữ và xử lý một số thao tác biên tập mô hình TIN. Hơn nữa, việc sử dụng cấu trúc cạnh kép sẽ thuận tiện cho việc kết hợp xử lý một số bài toán liên quan tới địa hình và địa chính sau này.

© 2016 Trường Đại học Mỏ - Địa chất. Tất cả các quyền được bảo đảm.

## 1. Đặt vấn đề

Khi xây dựng mô hình số độ cao theo mạng lưới tam giác không quy chuẩn (TIN) cần phải lựa chọn cấu trúc dữ liệu phù hợp để đảm bảo mô hình này có tính linh hoạt cao, có thể ứng dụng xử lý cho nhiều bài toán cụ thể. Cách tổ chức các cấu trúc dữ liệu có ảnh hưởng trực tiếp đến việc tam giác hóa

và các thao tác biên tập mô hình TIN. Thuật toán tăng dần với cấu trúc dữ liệu mạng lưới tam giác theo điểm cùng thuộc tính tam giác liền kề đã được nghiên cứu trong (Trần Thùy Dương và Nguyễn Văn Hiệp, 2007). Trên thực tế nhiều nghiên cứu đã chỉ ra rằng, cấu trúc cạnh kép thuận tiện cho việc xử lý các thao tác biên tập mô hình TIN cũng như mô hình topo tổng quát (Скворцов, 2002; Trần Thùy Dương và Phạm Thế Huỳnh, 2014). Tính chặt chẽ của cấu trúc cạnh kép phù hợp cho việc thực hiện một số thao tác cơ bản trong việc tạo mô hình topo như: cho biết một

\*Tác giả liên hệ.

E-mail: [ngolien.humg@gmail.com](mailto:ngolien.humg@gmail.com)

vùng, tìm vùng kế cận; cho biết một vùng, tìm các cạnh biên của vùng; cho biết một đỉnh, tìm tất cả các cạnh liên quan (Mark de Berg nnk, 2000). Hiện nay, cấu trúc cạnh kép đã trở thành cấu trúc chuẩn trong xây dựng cơ sở dữ liệu địa chính ở Việt Nam và một số nước trên thế giới.

Trong mô hình TIN cấu trúc cạnh kép có đôi chút khác biệt so với cấu trúc cạnh kép tổng quát của mô hình topo do tính đặc thù của mô hình TIN là chỉ bao gồm một mạng lưới các tam giác. Do vậy khi xây dựng cấu trúc cạnh kép cho mô hình TIN có thể rút gọn một số thuộc tính để tiết kiệm bộ nhớ máy tính. Cấu trúc cạnh kép rất thuận tiện để cho việc xử lý các thao tác biên tập tuy nhiên nhược điểm của cấu trúc này là các tam giác không được biểu diễn trực tiếp cũng như sự lãng phí lớn bộ nhớ. Nó chiếm lớn hơn 64.N byte. Bao gồm 8 byte biểu diễn tọa độ và 4 byte biểu diễn các chỉ số, chưa tính tới các bộ nhớ bị mất do biểu diễn các dữ liệu bổ sung của tam giác (Скворцов, 2002).

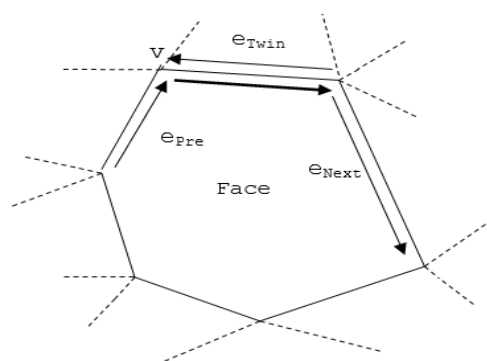
Hiện nay, để nâng cao khả năng khai thác dữ liệu trong giải quyết một số bài toán liên quan tới quản lý đô thị như xác định vùng ngập úng và phạm vi ảnh hưởng của nó tới hiện trạng sử dụng đất hay bài toán xây dựng các công trình ngầm, công trình nổi xác định phạm vi ảnh hưởng để di dời công trình liên quan,... thì việc kết hợp dữ liệu "địa hình" và "địa chính" là rất cần thiết. Do vậy việc sử dụng cấu trúc cạnh kép trong mô hình TIN là một giải pháp hợp lý khi kết hợp với mô hình topo để giải quyết các bài toán trên.

## 2. Cấu trúc cạnh kép tổng quát trong mô hình topo

Cấu trúc cạnh kép tổng quát trong việc tạo mô hình topo được tổ chức theo ba thành phần là cấu trúc điểm, cấu trúc cạnh và cấu trúc vùng.

### 2.1. Cấu trúc điểm

Cấu trúc điểm lưu trữ các thuộc tính thành phần tọa độ x, y. Để thuận tiện cho các thao tác biên tập có thể gắn thêm một thuộc tính cạnh có điểm gốc với cùng tọa độ. Như vậy dung lượng cần thiết để lưu trữ một điểm là  $2 \times 8 + 4 = 20$  byte. Với cách lưu trữ như vậy ta có thể dễ dàng xác định được các cạnh liên quan tới một đỉnh mà không mất quá nhiều thời gian tìm kiếm. Tuy nhiên, với một số lượng điểm quá lớn thì việc lưu trữ thêm một thuộc tính cũng sẽ tốn khá nhiều bộ nhớ. Nếu



Hình 1. Cấu trúc cạnh kép tổng quát trong mô hình topo

sử dụng ngôn ngữ Visual Basic thì cấu trúc điểm được mô tả như sau:

```
Private Type Point
    iE as Long      : Cạnh
    x as Double    : Tọa độ x
    y as Double    : Tọa độ y
End Type
```

### 2.2. Cấu trúc cạnh

Cấu trúc cạnh bao gồm các thuộc tính liên quan đến cạnh như: điểm gốc của cạnh; cạnh đảo  $e_{Twin}$  (cạnh ngược chiều); cạnh sau  $e_{Next}$  (có điểm gốc là điểm đích của cạnh e và có cùng vùng bên phải); cạnh trước  $e_{Pre}$  (có điểm đích là điểm gốc của cạnh e và có cùng vùng bên phải); vùng bên phải Fr. Như vậy dung lượng cần thiết để lưu trữ một cạnh là  $5 \times 4 = 20$  byte. Cách lưu trữ dữ liệu này khá chặt chẽ giúp cho việc xác định các mối quan hệ hàng xóm trong bài toán topo trở nên dễ dàng, tuy nhiên cũng tốn khá nhiều bộ nhớ. Cấu trúc cạnh được mô tả bằng ngôn ngữ Visual Basic như sau:

```
Private type Edge
    iV as Long          : Đỉnh
    iTwin as Long      : Cạnh đảo
    iNext as Long      : Cạnh sau
    iPre as Long       : Cạnh trước
    iFr as Long        : Mặt
End Type
```

### 2.3. Cấu trúc vùng

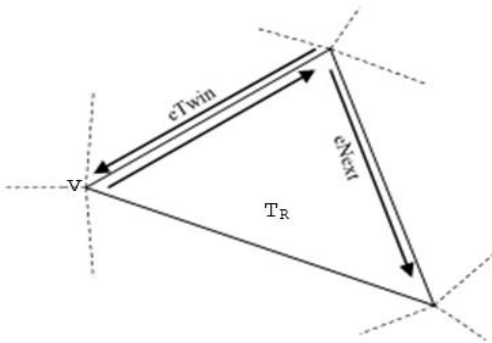
Cấu trúc vùng bao gồm hai thuộc tính: thuộc tính thứ nhất là cạnh thuộc vùng đó theo chiều thuận chiều kim đồng hồ (trong trường hợp vùng đang xét tới là vùng biên thì thuộc tính này rỗng); thuộc tính thứ hai là một danh sách các cạnh đảo

chiều mà mỗi cạnh ứng với một vùng nằm bên trong vùng đó, do đó số cạnh sẽ bằng số vùng nằm bên trong (Mark de Berg, and nnk, 2000). Như vậy dung lượng tối thiểu cần thiết để lưu trữ cấu trúc này là  $2 \times 3 = 6\text{byte}$ . Việc lưu trữ cấu trúc vùng như vậy rất thuận tiện khi giải quyết các trường hợp vùng nằm trong vùng của bài toán topo. Cấu trúc vùng được mô tả bằng ngôn ngữ Visual Basic như sau:

```
Private type Face
iEOut as Long: Cạnh đảo thuộc vùng ngoài
iEIn() as Long: Cạnh thuộc vùng
End Type
```

**3. Cấu trúc cạnh kép trong mô hình TIN**

Cấu trúc cạnh kép của mô hình TIN được biểu diễn qua ba thành phần là cấu trúc điểm, cấu trúc cạnh và cấu trúc tam giác. Theo (Скворцов, 2002) cấu trúc này được mô tả như Hình 2.



Hình 2. Cấu trúc cạnh kép trong mô hình TIN

**3.1. Cấu trúc điểm**

Cấu trúc điểm trong mô hình tam giác bao gồm các thuộc tính tọa độ x, y. Như vậy dung lượng tối thiểu để lưu trữ một điểm là  $2 \times 8 = 16\text{byte}$ . Cấu trúc điểm ở đây so với cấu trúc điểm trong cấu trúc cạnh kép tổng quát đã bị lược bỏ thuộc tính cạnh e. Trong việc xử lý dữ liệu có số lượng điểm lớn lên tới hàng chục triệu điểm (với khả năng thu thập dữ liệu bằng các công nghệ LIDAR, IFSAR, GNSS hiện nay) thì cấu trúc này sẽ giúp tiết kiệm một dung lượng lớn bộ nhớ máy tính. Tuy nhiên, việc rút gọn như vậy có thể gây khó khăn cho các thao tác biên tập mô hình TIN sau này. Nếu sử dụng ngôn ngữ Visual Basic, cấu trúc điểm được mô tả như sau:

```
Private Type Point_2D
```

```
x As Double : Tọa độ x
y As Double : Tọa độ y
End Type
```

**3.2. Cấu trúc cạnh**

Cấu trúc cạnh bao gồm các thuộc tính liên quan đến cạnh là đỉnh v, cạnh đảo eTwin, cạnh sau eNext và thuộc tính liên quan đến tam giác là TR. Trong đó các cạnh trong một tam giác sẽ được sắp xếp theo thứ tự thuận chiều kim đồng hồ. Như vậy để lưu trữ một cạnh cần  $4 \times 3 = 12\text{byte}$ . So với cấu trúc cạnh kép tổng quát thì cấu trúc cạnh kép ở đây đã bị lược bỏ thuộc tính cạnh trước, việc lược bỏ này là phù hợp với tính chất đặc biệt của mạng lưới tam giác (các vùng là các tam giác). Cấu trúc cạnh được biểu diễn thông qua ngôn ngữ Visual Basic như sau:

```
Private Type EdgeDC
iV1 As Long : Đỉnh
iTwin As Long : Cạnh đảo
iENext As Long : Cạnh sau
iTR As Long : Tam giác
End Type
```

**3.3. Cấu trúc tam giác**

Các tam giác trong mô hình TIN đóng vai trò như các vùng trong mô hình topo. Cấu trúc dữ liệu này biểu diễn tam giác ở dạng không tương minh do vậy cấu trúc tam giác ở đây chỉ là một danh sách các số hiệu tam giác mà không có bất cứ thuộc tính nào. Cấu trúc tam giác là dữ liệu được xác định từ cấu trúc điểm và cấu trúc cạnh, được lưu lại để thuận tiện cho việc thực hiện các bài toán ứng dụng sau này.

**4. Cập nhật dữ liệu cho các thao tác**

Để khẳng định sự phù hợp của cấu trúc cạnh kép trong mô hình TIN, nhóm tác giả đã tiến hành việc cài đặt cho thuật toán tam giác hóa theo phương pháp tăng dần và một số thao tác biên tập.

**4.1. Tạo tam giác khi chèn điểm**

Bảng 1. Dữ liệu trước khi thêm điểm

e	v	eTwin	eNext	TR
e1	v2	e'1	e2	T1
e2	v3	e'2	e3	T1
e3	v1	e'3	e1	T1

Khi một điểm được chèn nằm trong tam giác thì tam giác đó sẽ được chia thành ba tam giác nhỏ hơn, các tam giác này được kiểm tra với các tam giác kề cận để kiểm tra điều kiện hoán đổi tam giác. Sử dụng thao tác chèn điểm với lần lượt từng điểm khác, chúng ta sẽ xây dựng được một lưới tam giác Delaunay biểu diễn bề mặt địa hình. Cụ thể hơn, ta xem xét các sự kiện xảy ra khi thêm một điểm (Hình 3).

4.1.1. Điểm nằm trong tam giác

Bảng 2. Dữ liệu sau khi chèn thêm một điểm

e	v	e <sub>Twin</sub>	e <sub>Next</sub>	T <sub>R</sub>
e <sub>1</sub>	v <sub>2</sub>	e' <sub>1</sub>	e <sub>4</sub>	T <sub>1</sub>
e <sub>2</sub>	v <sub>3</sub>	e' <sub>2</sub>	e <sub>6</sub>	T <sub>2</sub>
e <sub>3</sub>	v <sub>1</sub>	e' <sub>3</sub>	e <sub>8</sub>	T <sub>3</sub>
e <sub>4</sub>	v <sub>3</sub>	e <sub>7</sub>	e <sub>5</sub>	T <sub>1</sub>
e <sub>5</sub>	v <sub>4</sub>	e <sub>8</sub>	e <sub>1</sub>	T <sub>1</sub>
e <sub>6</sub>	v <sub>1</sub>	e <sub>9</sub>	e <sub>7</sub>	T <sub>2</sub>
e <sub>7</sub>	v <sub>4</sub>	e <sub>4</sub>	e <sub>2</sub>	T <sub>2</sub>
e <sub>8</sub>	v <sub>2</sub>	e <sub>5</sub>	e <sub>9</sub>	T <sub>3</sub>
e <sub>9</sub>	v <sub>4</sub>	e <sub>6</sub>	e <sub>3</sub>	T <sub>3</sub>

Trước tiên phải xác định điểm này nằm trong tam giác nào trong mạng lưới tam giác hiện có, giả sử tam giác tìm được có số hiệu là T<sub>1</sub>, các cạnh là e<sub>1</sub>, e<sub>2</sub>, e<sub>3</sub>. Tam giác T<sub>1</sub> được mô tả thông qua cấu trúc cạnh kép như Bảng 1.

Sau đó, nối đỉnh v<sub>4</sub> với các đỉnh v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub> của tam giác T<sub>1</sub> và chia tam giác đó thành ba tam giác T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>. Các tam giác mới được tạo ra thông qua việc thêm các cạnh khi nối đỉnh v<sub>4</sub> với ba đỉnh của tam giác T<sub>1</sub>, số hiệu của tam giác T<sub>1</sub> được giữ

nguyên và chỉ thay đổi các thuộc tính của các cạnh của tam giác T<sub>1</sub> (Bảng 2).

(Trong quá trình này, thuộc tính e<sub>Twin</sub> của các cạnh e<sub>1</sub>, e<sub>2</sub>, e<sub>3</sub> không thay đổi nên ta không xét tới)

4.1.2. Điểm nằm trên cạnh

Nếu đỉnh v<sub>4</sub> nằm trên cạnh e của tam giác T<sub>1</sub> thì dựa vào thuộc tính e<sub>Twin</sub> của cạnh e ta có thể tìm được tam giác kế cận, nối điểm v<sub>4</sub> với hai đỉnh đối diện của hai tam giác liền kề (Hình 4).

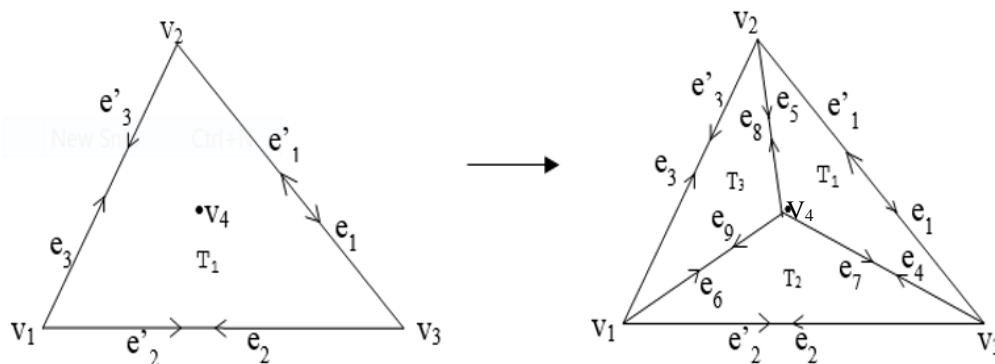
So với cấu trúc dữ liệu được mô tả trong (Trần Thùy Dương và Nguyễn Văn Hiệp, 2007) thì thao tác này không cần cập nhật ba tam giác liền kề.

4.2. Hoán đổi tam giác

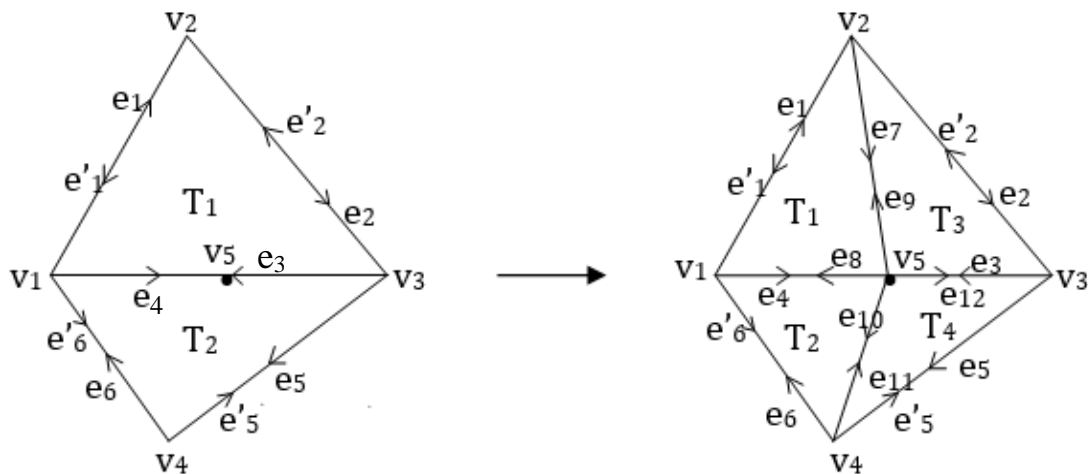
Thao tác hoán đổi tam giác được sử dụng trong cả quá trình tam giác hóa cũng như việc biên tập, chỉnh sửa mô hình.

Bảng 3. Dữ liệu trước khi thêm điểm

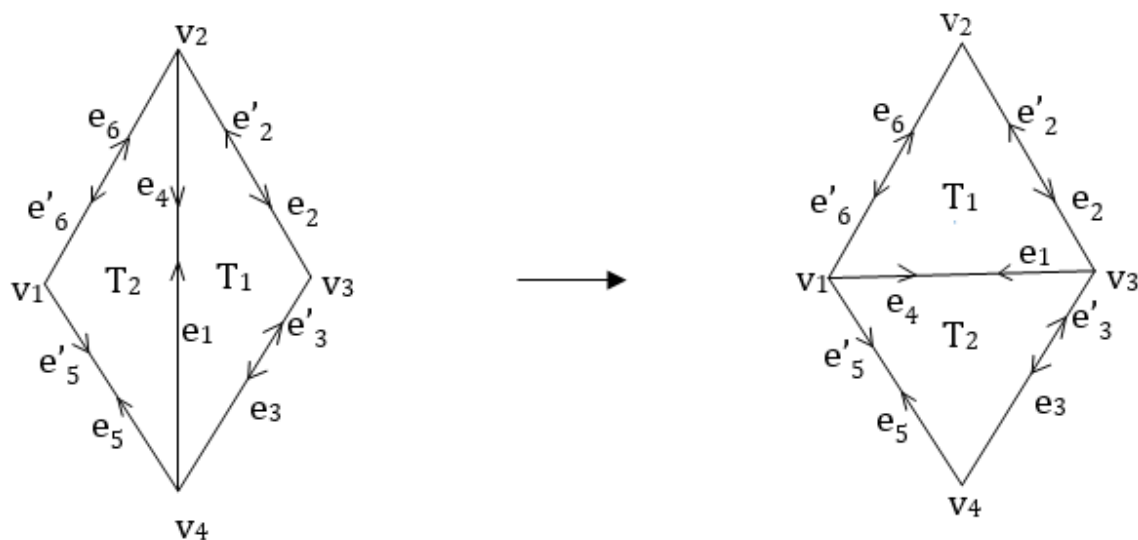
e	v	e <sub>Twin</sub>	e <sub>Next</sub>	T <sub>R</sub>
e <sub>1</sub>	v <sub>1</sub>	e' <sub>1</sub>	e <sub>2</sub>	T <sub>1</sub>
e <sub>2</sub>	v <sub>2</sub>	e' <sub>2</sub>	e <sub>3</sub>	T <sub>1</sub>
e <sub>3</sub>	v <sub>3</sub>	e <sub>4</sub>	e <sub>1</sub>	T <sub>1</sub>
e <sub>4</sub>	v <sub>4</sub>	e <sub>3</sub>	e <sub>5</sub>	T <sub>2</sub>
e <sub>5</sub>	v <sub>3</sub>	e' <sub>5</sub>	e <sub>6</sub>	T <sub>2</sub>
e <sub>6</sub>	v <sub>4</sub>	e' <sub>6</sub>	e <sub>4</sub>	T <sub>2</sub>



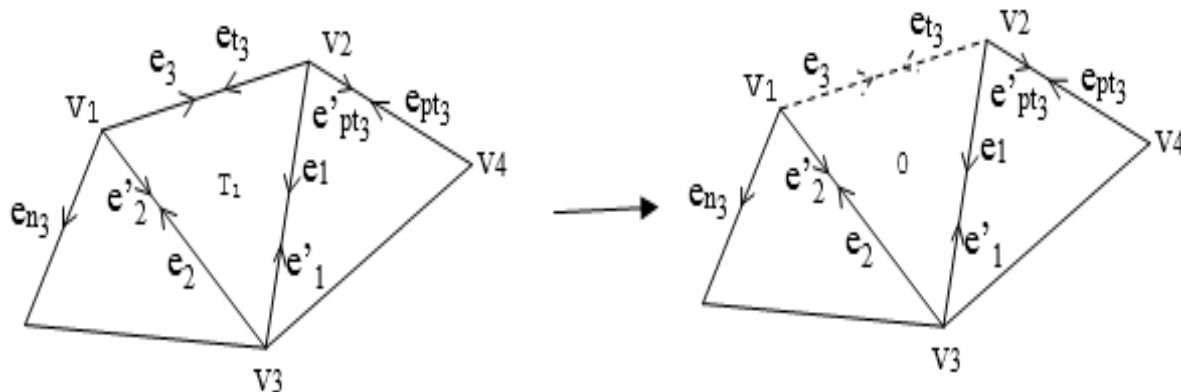
Hình 3. Điểm v<sub>4</sub> nằm trong tam giác T<sub>1</sub>



Hình 4. Đỉnh  $v_5$  nằm trên cạnh  $e_3$



Hình 5. Hoán đổi tam giác theo điều kiện Delaunay



Hình 6. Trường hợp xóa tam giác có một cạnh biên

Khi tam giác hóa có thể xảy ra hai trường hợp: nếu đường tròn ngoại tiếp tam giác con chứa đỉnh còn lại của tam giác liền kề (không thỏa mãn điều kiện Delaunay) hoặc khi biên tập chỉnh sửa mô hình TIN hai tam giác cần hoán đổi là  $T_1$  và  $T_2$  (cạnh cần chọn là cạnh  $e_1$ ) thì sẽ phải thực hiện thao tác hoán đổi tam giác như Hình 5.

Bảng 4. Dữ liệu sau khi thêm điểm

e	v	eTwin	eNext	TR
e <sub>1</sub>	v <sub>1</sub>	e' <sub>1</sub>	e <sub>7</sub>	T <sub>1</sub>
e <sub>2</sub>	v <sub>2</sub>	e' <sub>2</sub>	e <sub>3</sub>	T <sub>3</sub>
e <sub>3</sub>	v <sub>3</sub>	e' <sub>3</sub>	e <sub>9</sub>	T <sub>3</sub>
e <sub>4</sub>	v <sub>1</sub>	e <sub>8</sub>	e <sub>10</sub>	T <sub>2</sub>
e <sub>5</sub>	v <sub>3</sub>	e' <sub>5</sub>	e <sub>11</sub>	T <sub>4</sub>
e <sub>6</sub>	v <sub>4</sub>	e' <sub>6</sub>	e <sub>4</sub>	T <sub>2</sub>
e <sub>7</sub>	v <sub>2</sub>	e <sub>9</sub>	e <sub>8</sub>	T <sub>1</sub>
e <sub>8</sub>	v <sub>5</sub>	e <sub>4</sub>	e <sub>1</sub>	T <sub>1</sub>
e <sub>9</sub>	v <sub>5</sub>	e <sub>7</sub>	e <sub>2</sub>	T <sub>3</sub>
e <sub>10</sub>	v <sub>5</sub>	e <sub>11</sub>	e' <sub>5</sub>	T <sub>2</sub>
e <sub>11</sub>	v <sub>4</sub>	e <sub>10</sub>	e <sub>12</sub>	T <sub>4</sub>
e <sub>12</sub>	v <sub>5</sub>	e <sub>3</sub>	e <sub>5</sub>	T <sub>4</sub>

### 4.3. Xóa tam giác

Trong cấu trúc này, tồn tại các cạnh có thuộc tính eTwin là các cạnh có thuộc tính TR bằng 0

được gọi là cạnh biên. Thao tác xóa tam giác ở đây được chia thành các trường hợp sau:

- Xóa tam giác thông thường (tức là các tam giác không có chứa cạnh biên);
- Xóa tam giác có một cạnh là cạnh biên;
- Xóa tam giác có hai cạnh là cạnh biên;
- Xóa tam giác có cả ba cạnh là cạnh biên.

#### 4.3.1. Xóa tam giác thông thường.

Trong trường hợp này, các thuộc tính bị thay đổi chỉ bao gồm các thuộc tính TR của các cạnh tạo thành tam giác cần xóa. Đầu tiên ta phải xác định tam giác cần xóa, sau đó cập nhật thuộc tính TR cho các cạnh tạo thành tam giác này. Trong ngôn ngữ Visual Basic, thao tác này được mô tả như sau:

```

DS_EdgesDC(iEx).iTR = 0
DS_EdgesDC(iExn).iTR = 0
DS_EdgesDC(iExnn).iTR = 0
    
```

Trong đó danh sách DS\_EdgesDC ( ) là danh sách lưu trữ các nửa cạnh và ex, ex<sub>n</sub>, ex<sub>nn</sub> là các cạnh của tam giác cần xóa. Trong đó ex<sub>n</sub> là cạnh sau của cạnh ex và ex<sub>nn</sub> là cạnh sau của cạnh ex<sub>n</sub>.

#### 4.3.2. Xóa tam giác có chứa một cạnh biên.

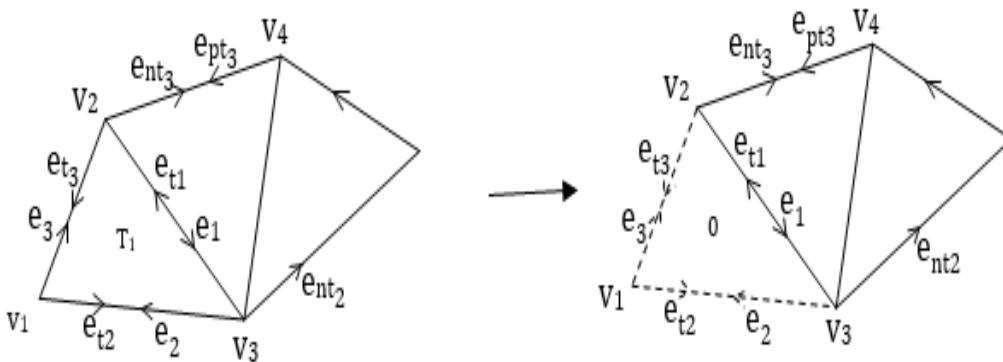
Trong trường hợp này, khi xóa tam giác ta cần cập nhật các thuộc tính của hai cạnh còn lại và các thuộc tính của các cạnh có liên quan tới cạnh bị xóa. Như Hình 6, Bảng 6.

Bảng 5. Dữ liệu trước và sau khi hoán đổi

Dữ liệu trước hoán đổi					Dữ liệu sau hoán đổi				
e	v	eTwin	eNext	TR	e	v	eTwin	eNext	TR
e <sub>1</sub>	v <sub>2</sub>	e' <sub>1</sub>	e <sub>2</sub>	T <sub>1</sub>	e <sub>1</sub>	v <sub>3</sub>	e' <sub>1</sub>	e <sub>3</sub>	T <sub>1</sub>
e <sub>2</sub>	v <sub>4</sub>	e' <sub>2</sub>	e <sub>3</sub>	T <sub>1</sub>	e <sub>2</sub>	v <sub>4</sub>	e' <sub>2</sub>	e <sub>4</sub>	T <sub>1</sub>
e <sub>3</sub>	v <sub>1</sub>	e' <sub>3</sub>	e <sub>1</sub>	T <sub>1</sub>	e <sub>3</sub>	v <sub>1</sub>	e' <sub>3</sub>	e' <sub>3</sub>	T <sub>2</sub>
e <sub>4</sub>	v <sub>4</sub>	e <sub>1</sub>	e' <sub>3</sub>	T <sub>2</sub>	e <sub>4</sub>	v <sub>1</sub>	e <sub>1</sub>	e' <sub>2</sub>	T <sub>2</sub>
e <sub>5</sub>	v <sub>3</sub>	e' <sub>5</sub>	e <sub>4</sub>	T <sub>2</sub>	e <sub>5</sub>	v <sub>3</sub>	e' <sub>5</sub>	e <sub>2</sub>	T <sub>2</sub>
e <sub>6</sub>	v <sub>2</sub>	e' <sub>6</sub>	e' <sub>2</sub>	T <sub>2</sub>	e <sub>6</sub>	v <sub>2</sub>	e' <sub>6</sub>	e <sub>1</sub>	T <sub>1</sub>

Bảng 6. Dữ liệu trước và sau khi xóa tam giác trường hợp 2.

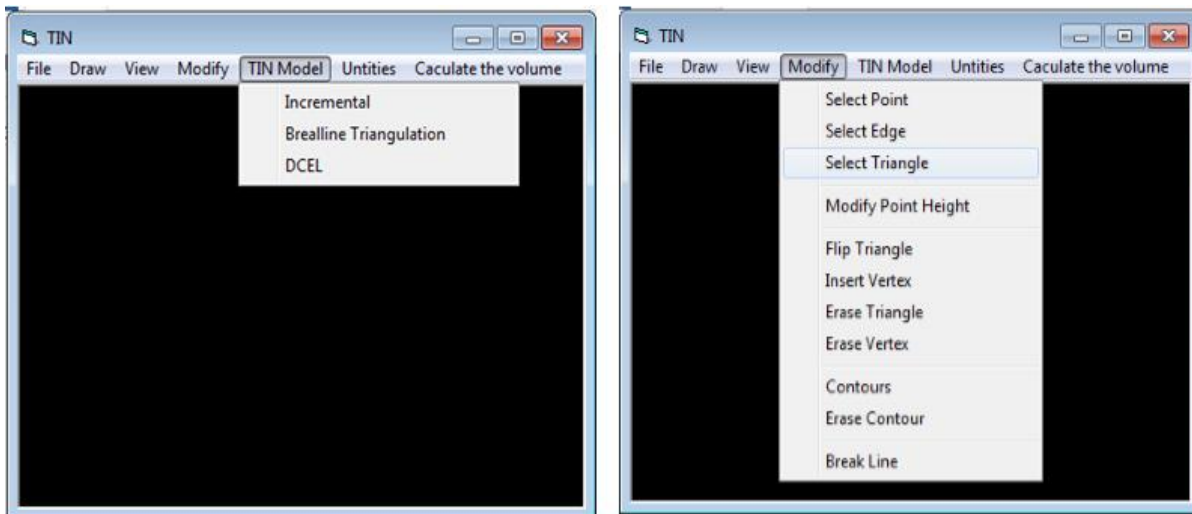
Dữ liệu ban đầu					Dữ liệu sau khi xóa tam giác				
e	v	eTwin	eNext	TR	e	v	eTwin	eNext	TR
e <sub>1</sub>	v <sub>2</sub>	e' <sub>1</sub>	e <sub>2</sub>	T <sub>1</sub>	e <sub>1</sub>	v <sub>2</sub>	e' <sub>1</sub>	e <sub>2</sub>	0
e <sub>2</sub>	v <sub>3</sub>	e' <sub>2</sub>	e <sub>3</sub>	T <sub>1</sub>	e <sub>2</sub>	v <sub>3</sub>	e' <sub>2</sub>	e <sub>n3</sub>	0
e <sub>3</sub>	v <sub>1</sub>	e <sub>t3</sub>	e <sub>1</sub>	T <sub>1</sub>	e <sub>3</sub>	0	e <sub>t3</sub>	0	0
e <sub>t3</sub>	v <sub>2</sub>	e <sub>3</sub>	e <sub>n3</sub>	0	e <sub>t3</sub>	0	e <sub>3</sub>	0	0
e <sub>pt3</sub>	v <sub>4</sub>	e' <sub>pt3</sub>	e <sub>t3</sub>	0	e <sub>pt3</sub>	v <sub>4</sub>	e' <sub>pt3</sub>	e <sub>1</sub>	0



Hình 7. Trường hợp xóa tam giác có hai cạnh biên.

Bảng 7. Dữ liệu sau khi xóa tam giác T<sub>1</sub> trường hợp 3

Dữ liệu ban đầu					Dữ liệu ban đầu				
e	v	e <sub>Twin</sub>	e <sub>Next</sub>	T <sub>R</sub>	e	v	e <sub>Twin</sub>	e <sub>Next</sub>	T <sub>R</sub>
e <sub>1</sub>	v <sub>2</sub>	e <sub>t1</sub>	e <sub>2</sub>	T <sub>1</sub>	e <sub>1</sub>	v <sub>2</sub>	e <sub>t1</sub>	e <sub>nt2</sub>	0
e <sub>2</sub>	v <sub>3</sub>	e <sub>t2</sub>	e <sub>3</sub>	T <sub>1</sub>	e <sub>2</sub>	0	e <sub>t2</sub>	0	0
e <sub>3</sub>	v <sub>1</sub>	e <sub>t3</sub>	e <sub>1</sub>	T <sub>1</sub>	e <sub>3</sub>	0	e <sub>t3</sub>	0	0
e <sub>t3</sub>	v <sub>2</sub>	e <sub>3</sub>	e <sub>t2</sub>	0	e <sub>t3</sub>	0	e <sub>3</sub>	0	0
e <sub>pt3</sub>	v <sub>4</sub>	e <sub>nt3</sub>	e <sub>t3</sub>	0	e <sub>pt3</sub>	v <sub>4</sub>	e <sub>nt3</sub>	e <sub>1</sub>	0
e <sub>t2</sub>	v <sub>1</sub>	e <sub>2</sub>	e <sub>nt2</sub>	0	e <sub>t2</sub>	0	e <sub>2</sub>	0	0



Hình 8. Chương trình tam giác hóa theo thuật toán tăng dần và một số thao tác biên tập.

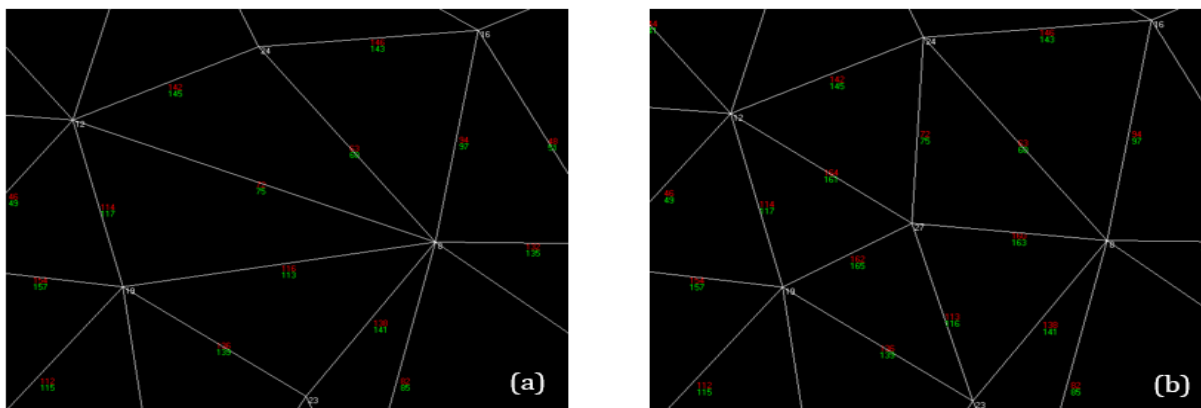
4.3.3. Xóa tam giác có hai cạnh là cạnh biên

Trường hợp này, các thuộc tính thay đổi là toàn bộ các thuộc tính của các cạnh của tam giác cần xóa và toàn bộ các cạnh đảo của mỗi cạnh. Như vậy để thực hiện thao tác này ta chỉ cần tiến hành cập nhật các thuộc tính đó bằng 0.

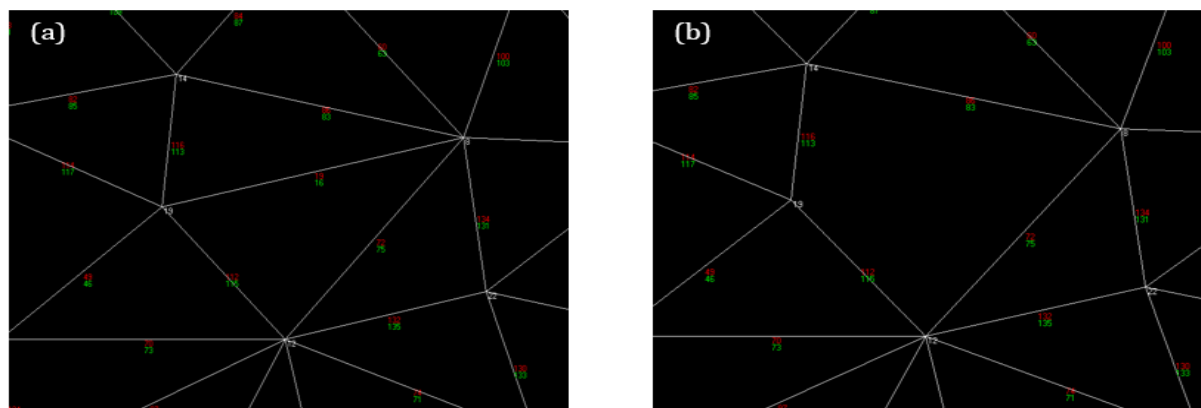
Do các tam giác được biểu diễn gián tiếp thông qua các cạnh nên việc xóa tam giác trở nên khó khăn và xảy ra nhiều trường hợp đặc biệt. Như Hình 7, Bảng 7.

5. Thực nghiệm và kết luận

5.1. Thực nghiệm



Hình 9. (a) Trước khi chèn điểm 27; (b) Sau khi chèn điểm 27



Hình 10. (a) Trước khi xóa tam giác, (b) Sau khi xóa tam giác

Để khảo sát cấu trúc cạnh kép trong việc tam giác hóa và thực hiện một số thao tác biên tập, nhóm tác giả đã tiến hành xây dựng chương trình thực nghiệm bằng ngôn ngữ Visual Basic như Hình 8, 9, 10. Một số thao tác biên tập được thể hiện bằng chương trình nhóm tác giả đã xây dựng bằng ngôn ngữ Visual Basic 6.0.

**5.2. Kết luận**

Việc sử dụng cấu trúc cạnh kép cho việc tam giác hóa và xử lý các thao tác biên tập mô hình TIN cho thấy cấu trúc này có một số ưu, nhược điểm như sau :

Ưu điểm: Phù hợp và thuận lợi đối với các thao tác liên quan trực tiếp đến cạnh như thao tác chèn điểm, thao tác hoán đổi tam giác.

Nhược điểm: Do các tam giác được biểu diễn gián tiếp thông qua các cạnh nên đối với các thao tác liên quan trực tiếp đến các tam giác thì cấu trúc này còn hạn chế.

Các kết quả đánh giá đã được kiểm chứng bằng chương trình viết trên môi trường Visual

Basic 6.0.

**Tài liệu tham khảo**

Agarwal, P. K. and Van Kreveld. M., 1994. Implicit point location in arrangements of line segments, with an application to motion planning. *Internat. J. Comput. Geom. Appl* 4, 369-383.

Agarwal, P., and Erickson, J., 1998. Geometric range searching and its relatives. (In B. Chazelle, J. Goodman, and R. Pollack, editors), *Advances in Discrete and Computational Geometry*, 1-56. American Mathematical Society.

Akman, V., 1984. Unobstructed Shortest Paths in Polyhedral Environments. *Lecture Notes in Computer Science* 251. Springer-Verlag.

Aronov, B., de Berg, M., and Gray, C., 2006. Ray shooting and intersection searching amidst fat convex polyhedra in 3-space. *In Proc. 22nd*



- Annu. ACM Sympos. Comput. Geom.*, 88-94.
- Atkinson, K.E., 1989. *An Introduction to Numerical Analysis*, 2nd Edition. John Wiley and Sons, New York.
- Aurenhammer, F., and Schwarzkopf, O., 1992. A simple on-line randomized incremental algorithm for computing higher order Voronoi diagrams. *Internat. J. Comput. Geom. Appl.* 2, 363-381.
- Barequet G., Dickerson M., Eppstein D., 1996. On triangulating three-dimensional polygons, *In Proceedings of the twelfth annual symposium on Computational geometry*, New York, NY, USA, SCG '96, ACM, 38-47.
- Скворцов, А. В., 2002. Триангуляция Делоне и её применение, *издательство Томского университета*.
- Mark, D. B., Marc, V. K., Mark, O., Otfried, C., Ne, S., 2000. *Computational Geometry, Algorithms and Applications*, Springer-Verlag, Berlin.
- P. Agarwal. Range searching. (In J. E. Goodman and J. O'Rourke, editors) 2004. *Handbook of Discrete and Computational Geometry*, 2nd edn., chapter 36. Chapman & Hall/CRC.
- Robert Sedgewick, 1995. *Cẩm nang thuật toán, Tập 1: Các thuật toán thông dụng*, Nhà xuất bản Khoa học kỹ thuật, Hà Nội.
- Robert Sedgewick, 1995. *Cẩm nang thuật toán, Tập 2: Các thuật toán chuyên dụng*, Nhà xuất bản Khoa học kỹ thuật, Hà Nội.
- Trần Thùy Dương, Nguyễn Văn Hiệp, 2007. Thuật toán tăng dần với cấu trúc dữ liệu mạng lưới tam giác theo điểm cùng thuộc tính tam giác liền kề. *Tạp chí khoa học kỹ thuật Mỏ - Địa Chất* 20, 17-21.
- Trần Thùy Dương, Phạm Thế Huynh, 2014. Một cách tiếp cận mới trong việc giải quyết bài toán chồng phủ vùng sử dụng cấu trúc dữ liệu danh sách cạnh liền kết kép. *Tạp chí khoa học kỹ thuật Mỏ - Địa Chất* 46, 73-76.

## ABSTRACT

### Use the doubly connected edge list (DCEL) structure for storing and processing some editing triangular irregular network (TIN) operations

Lien Thi Ngo <sup>1,\*</sup>, Duong Thuy Tran <sup>2</sup>, Hung Quang Le <sup>3</sup>

<sup>1</sup> *Informatic center of geodesy and cartography, Vietnam Institute of geodesy and cartography*

<sup>2</sup> *Faculty of Geomatics and Land Administration, University of Mining and Geology, Vietnam*

<sup>3</sup> *Resource Environment and Materials Technology Jointstock Company, Vietnam*

When building and handling some editing of triangular irregular network (TIN) operations, there are some different data structures that can be used to present the triangular irregular network, the doubly connected edge list (DCEL) structure is one of them. Currently, the triangular network is often handled with a lot of the triangles so researching the data structure which is suitable for the triangulation is necessary. With evaluating the DCEL structure purpose to use for triangulation, the paper analyzed and compared it with other structures for storing and processing some editing the triangular irregular network operations. The paper used analytical method and comparative method the DCEL structure with other structures and experiment program using the DCEL structure for storing and processing TIN model by Visual Basic 6.0 language. The paper evaluated advantages and disadvantages and gave some adjusting for the DCEL structure to more suitable for storing and processing some editing TIN model operations. Moreover, using the DCEL will facilitate for the combination handle some problems related to topographic and cadastral later.