

# MỘT PHƯƠNG PHÁP XÁC ĐỊNH VỊ TRÍ HIỂN THỊ THÔNG TIN PHI HÌNH HỌC TRONG HỆ THỐNG THÔNG TIN ĐỊA LÝ

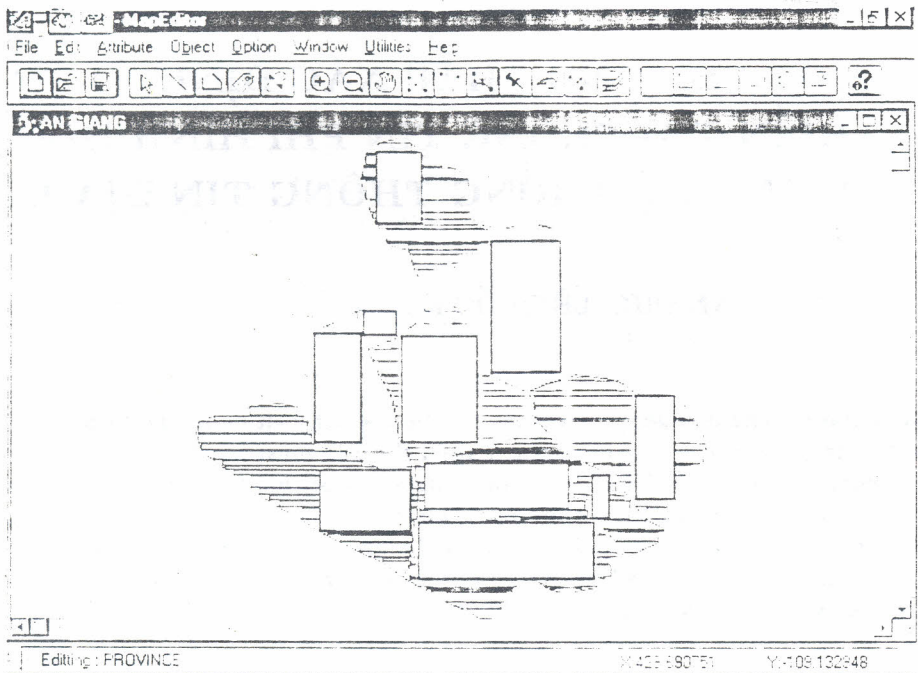
ĐẶNG VĂN ĐỨC, LÊ QUỐC HUNG<sup>1)</sup>, TRẦN CẨM NGÂN<sup>2)</sup>

**Abstract.** The non-geometrical information which can be in the form of number, character string, graph on the thematic map in the GIS must be clear, readable, and balanced in geographical areas. This report presents an algorithm for computing a number of candidate labeling rectangles within a polygon from which a suitable selection can be made. The relevant data graph shall be within the above mentioned rectangle. This algorithm has been applied to install the PopMap for Windows for the United Nations DEDIPA by the researchers of the IOIT.

## I. ĐẶT VẤN ĐỀ

Hệ thống thông tin địa lý (GIS) hướng dữ liệu kinh tế, xã hội là một loại hệ thống GIS chuyên dụng [2]. Hệ thống GIS loại này thường được xây dựng trên mô hình dữ liệu véc tơ. Các đặc trưng địa lý như sông ngòi, đường giao thông, các trường học, bệnh viện, các đường biên hành chính,... được mô tả bằng các đối tượng điểm, đường hay vùng trong cơ sở dữ liệu (CSDL). Thí dụ để mô tả một đơn vị hành chính trên màn hình máy tính hay máy in ta phải trừu tượng và đơn giản hóa thể giới thực hành đa giác hay tập hợp các tọa độ trong hệ trục tọa độ thực (tọa độ kinh tuyến - vĩ tuyến của trái đất). CSDL địa lý ngoài các dữ liệu hình học như nói trên chúng còn lưu trữ cả các dữ liệu thống kê, đó là các thuộc tính liên quan đến dữ liệu hình học tương ứng. Một trong các chức năng quan trọng của hệ thống GIS là khả năng thể hiện các kết quả thao tác trên CSDL địa lý dưới hình thức dễ hiểu, dễ đọc cho người dùng. Các dữ liệu dưới dạng số hay đồ họa của một thực thể (thí dụ của một đơn vị hành chính) phải được biểu diễn ngay trong thực thể đó. Nếu ta coi các thực thể miền được giới hạn bởi đa giác, thì vấn đề nghiên cứu đặt ra ở đây là tìm ra tập các hình chữ nhật nội tiếp trong đa giác. Từ đó sẽ lựa chọn được một hình chữ nhật phù hợp, là hình chữ nhật có diện tích lớn nhất từ chúng, để thể hiện dữ liệu như trên hình 1.

Tài liệu [1] cũng đã đề cập đến nhu cầu tìm kiếm miền hợp lý trong đối tượng vùng để hiển thị các dữ liệu phi hình học. Nhưng nghiên cứu này đã mở rộng thành thuật toán tìm kiếm vùng hiển thị trong cả đa giác lồi và đa giác không lồi trên mặt hình cầu hay hình elipsoid của trái đất.



Hình 1. Vị trí hiển thị dữ liệu thống kê trên các đơn vị hành chính

## II. PHƯƠNG PHÁP

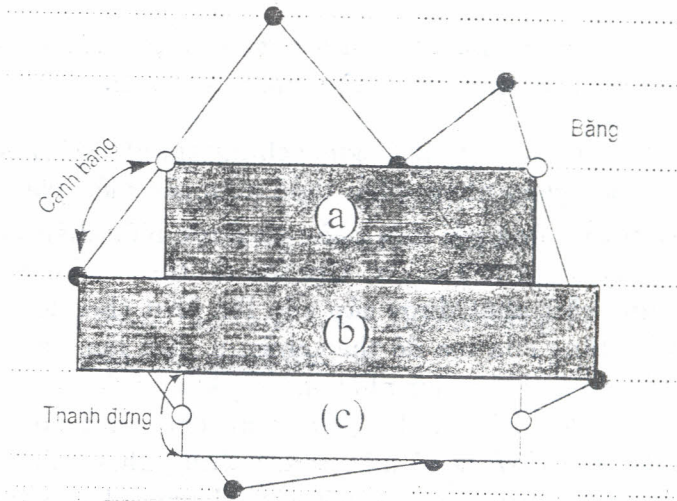
Ý tưởng cơ bản của phương pháp này bắt đầu từ chia đa giác thành các băng nhỏ, tìm diện tích của hình chữ nhật trong các băng rồi ghép nhóm chúng lại với nhau sao cho có được hình chữ nhật tổng, nội tiếp đa giác và phù hợp cho việc biểu diễn dữ liệu thống kê.

Để chia đa giác thành các băng, lần lượt qua các đỉnh của đa giác ta kẻ các đường thẳng song song với trục hoành chia đa giác thành nhiều phần. Với đa giác  $N$  đỉnh, ta có thể vẽ được nhiều nhất  $N$  đường thẳng song song qua  $N$  đỉnh; như vậy có thể chia đa giác ra cực đại là  $N + 1$  phần. Nhưng với băng ở trên cùng và băng ở thấp nhất, nghĩa là tương ứng với đỉnh có tung độ  $y$  lớn nhất và  $y$  nhỏ nhất sẽ không phải quan tâm đến vì nó không chứa miền đa giác. Cho nên trong một đa giác có thể có nhiều nhất  $N - 1$  băng.

Các đường thẳng song song qua các đỉnh đa giác chia các cạnh của chúng thành những đoạn nhỏ. Ta gọi các đường thẳng đó là đường phân băng, những đoạn nhỏ đó là những cạnh băng. Một băng sẽ có hai cạnh băng nếu đó là một đa giác lõm không chứa đa giác khác nằm trong. Còn đa giác lõm hoặc những đa giác chứa đa giác khác sẽ có rất nhiều cạnh băng.

Trong mỗi một băng ta vẽ các đường thẳng song song với trục tung qua cạnh

bằng và gọi là thanh đứng. Chúng sẽ được dùng để tính cạnh bên cho hình chữ nhật. Mặc dù thanh đứng có thể được tạo ra ở bất cứ vị trí nào trên cạnh bằng, chẳng hạn qua đỉnh nằm phía trong hoặc qua đỉnh nằm phía ngoài hay qua điểm giữa cạnh bằng. Nếu chọn thanh đứng đi qua điểm phía ngoài ta sẽ được hình chữ nhật đơn vị lớn nhất song chúng sẽ không nội tiếp trong đa giác. Nếu chọn thanh đứng đi qua điểm phía trong ta sẽ được các hình chữ nhật đơn vị nội tiếp trong đa giác song có thể bỏ lãng phí một số miền trong đa giác có thể "tận dụng" được. Với mục đích là tìm miền hiển thị dữ liệu cho bản đồ cho nên các góc của hình chữ nhật lớn nhất trong đa giác có thể chồm ra ngoài đa giác, song phần chồm phải khá nhỏ. Trong thực tế, các ký tự hiển thị thường nhỏ hơn phần nền mà nó chiếm. Vì vậy, tốt nhất (ở đây) là ta chọn thanh đứng đi qua điểm giữa cạnh bằng. Mục tiêu là phải tạo các hình chữ nhật tổng phù hợp nhất sao cho nếu mở rộng bề ngang cũng như bề dọc thì các hộp chứa dữ liệu hiển thị sẽ không vượt qua đường biên của đa giác.

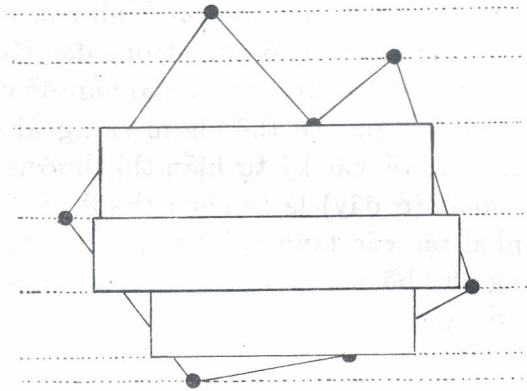


Hình 2. Các khả năng lựa chọn thanh đứng cho hình chữ nhật đơn vị

Chọn vị trí của thanh đứng sẽ tạo ra các hình chữ nhật đơn vị khác nhau. Hình (a) được tạo ra khi thanh đứng đi qua giao điểm bên trong nhất của đường thẳng chia bằng và cạnh bằng. Hình (b) được tạo ra khi thanh đứng đi qua giao điểm bên ngoài nhất của đường thẳng chia bằng. Còn hình (c) được tạo ra khi thanh đứng đi qua điểm giữa của cạnh bằng. Thực nghiệm cho thấy, khi đa giác có số lượng đỉnh rất lớn thì chọn giải pháp (a) là tốt hơn cả để tăng cường tốc độ thực hiện chương trình. Tuy nhiên nếu số lượng đỉnh ít hơn 7 thì phải chọn giải pháp (c).

Đối với một đa giác lồi mỗi băng chỉ có một hình chữ nhật đơn vị. Với đa giác  $N$  đỉnh sẽ có nhiều nhất là  $N - 1$  băng, như vậy có nhiều nhất  $N - 1$  hình chữ nhật đơn vị. Nhưng đối với đa giác không lồi  $N$  đỉnh, số hình chữ nhật đơn

vị nhiều nhất sẽ là  $(N - 1) \times (N - 1)$  do mỗi đỉnh có khả năng cắt nhiều nhất  $N - 1$  cạnh, như vậy cho mỗi băng sẽ có nhiều nhất  $N - 1$  hình chữ nhật đơn vị được tạo ra. Hình 3 minh họa các hình chữ nhật đơn vị được tạo ra trong trường hợp lựa chọn cạnh đứng hình chữ nhật đi qua điểm giữa của cạnh băng.



Hình 3. Các hình chữ nhật đơn vị của đa giác khi chọn thanh đứng đi qua điểm giữa cạnh băng

Từ những hình chữ nhật đơn vị ta sẽ ghép chúng thành nhóm để tạo ra những hình chữ nhật tổng. Việc ghép nhóm phải thỏa mãn các điều kiện sau đây:

Mỗi thanh đứng tương ứng với một phần tử được biểu diễn trong danh sách đầu vào đã sắp xếp  $s(i)$ , với  $i = 1, \dots, n$ . Gọi tọa độ theo chiều  $x$  của từng đoạn thẳng là  $x(i)$ . Các hình chữ nhật được sắp xếp lần lượt theo thứ tự từ cao đến thấp, tức là theo chiều tăng của tọa độ theo chiều  $y$ . Xét lần lượt từng hình chữ nhật đơn vị ghép nó với các hình chữ nhật đơn vị khác một cách hợp lý sao cho có được hình chữ nhật lớn nhất có thể. Gọi hình chữ nhật đơn vị là *box* (*left*, *right*, *bottom*, *top*), trong đó *left* là số hiệu thanh đứng phía trái, *right* là số hiệu thanh đứng phía phải, *bottom* và *top* là số hiệu các đường phân băng trên và dưới của hình chữ nhật đơn vị. Gọi cạnh trái phải của hình chữ nhật nằm trên băng thứ  $s$  là  $l_s$  (trái) và  $r_s$  (phải). Ta có các điều kiện sau:

$$P_{\text{left}} : x(\text{left}) = \max\{x(l_s) | \text{bottom} \leq s \leq \text{top}\}$$

$$P_{\text{right}} : x(\text{right}) = \min\{x(r_s) | \text{bottom} \leq s \leq \text{top}\}$$

$P_{\text{left}}$  và  $P_{\text{right}}$  là giới hạn trái và giới hạn phải. Điều kiện này có nghĩa là cạnh trái của hình chữ nhật lựa chọn sẽ là cạnh trái nằm phía bên phải nhất (có tọa độ  $x$  lớn nhất) trong các hình chữ nhật đơn vị; cạnh phải của hình chữ nhật lựa chọn sẽ là cạnh phải nằm ở phía bên trái nhất (có tọa độ  $x$  nhỏ nhất).

$$P_{\text{top}} : x(l_{\text{top}-1}) > x(l_{\text{top}}) \text{ hoặc } x(r_{\text{top}-1}) < x(r_{\text{top}})$$

$$P_{\text{bottom}} : x(l_{\text{bottom}+1}) > x(l_{\text{bottom}}) \text{ hoặc } x(r_{\text{bottom}+1}) < x(r_{\text{bottom}})$$

Điều kiện  $P_{\text{top}}$  có nghĩa là cạnh trái hình chữ nhật trên cùng trong nhóm lựa chọn

sẽ phải có tọa độ  $x$  nhỏ hơn tọa độ  $x$  của thanh trái nằm phía trên nó hoặc cạnh phải hình chữ nhật trên cùng trong nhóm lựa chọn sẽ phải có tọa độ  $x$  lớn hơn tọa độ  $x$  của thanh phải nằm phía trên nó.

Tương tự với điều kiện  $P_{\text{bottom}}$ , có nghĩa là cạnh trái hình chữ nhật dưới cùng trong nhóm lựa chọn sẽ phải có tọa độ  $x$  nhỏ hơn tọa độ  $x$  của thanh trái nằm phía dưới nó hoặc cạnh phải hình chữ nhật dưới cùng trong nhóm lựa chọn sẽ phải có tọa độ  $x$  lớn hơn tọa độ  $x$  của thanh phải nằm phía dưới nó.

Điều kiện miền trong:

$$P_{\text{inc}} : \text{bottom} \leq s_{\text{left}} \leq \text{top} \text{ và } \text{bottom} \leq s_{\text{right}} \leq \text{top}$$

có nghĩa là cạnh trái và cạnh phải của hình chữ nhật lựa chọn phải là một trong số các cạnh trái của hình chữ nhật đơn vị nằm trong nó.

Ghép các hình chữ nhật đơn vị dựa trên năm điều kiện bất biến trên ta sẽ có được tập các hình chữ nhật nội tiếp đa giác. Từ đó chúng ta sẽ chọn ra hình chữ nhật lớn nhất để đặt dữ liệu thống kê hay xâu ký tự.

### III. THUẬT TOÁN

Từ phương pháp trình bày trên đây, một thuật toán được xây dựng để có thể cài đặt bằng ngôn ngữ lập trình. Trong bài này, phương pháp chọn hình chữ nhật đơn vị theo (a) của hình 2 trên đây được sử dụng làm thí dụ minh họa. Các vấn đề chính cần giải quyết trong quá trình thực hiện bài toán tìm hình chữ nhật thích hợp là:

- Tính toán được các cạnh biên, thanh đứng từ đa giác cho trước để tạo được các hình chữ nhật đơn vị.
- Tìm hình chữ nhật nội tiếp lớn nhất từ những hình chữ nhật đơn vị đã tạo ra.
- Tách miền đa giác lõm theo phương nằm ngang thành các miền đa giác lồi theo phương nằm ngang.

#### 1. Tìm các hình chữ nhật đơn vị

Dữ liệu bản đồ được lưu trữ trong CSDL dưới dạng véc tơ như trình bày trong phần I. Miền đa giác được định nghĩa là miền  $E = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , trong đó:  $(x_i, y_i)$  là tọa độ các đỉnh liên tiếp và  $x_1 \equiv x_n$  và  $y_1 \equiv y_n$ . Trong CSDL đa giác được biểu diễn bởi tập hợp các tọa độ điểm như sau:

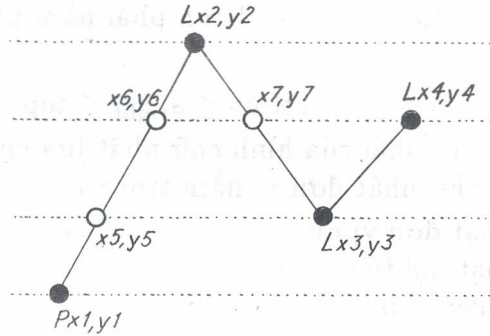
$$Px_1, y_1 \quad Lx_2, y_2 \quad Lx_3, y_3 \quad \dots \quad Lx_n, y_n \quad Lx_{n+1}, y_{n+1} \quad \dots$$

Trong đó,  $Px_i, y_i$  là tọa độ của nơi đặt bút vẽ,  $Lx_i, y_i$  là tọa độ của điểm được nối với điểm trước đó. Tọa độ của vị trí cuối danh sách được tự động nối với tọa độ của điểm đầu tiên để tạo ra đa giác.

Tọa độ của hình chữ nhật đơn vị được xác định theo các bước sau:

a) Tính giao điểm của các đường thẳng nằm ngang đi qua các đỉnh của đa giác với các cạnh của chúng để xác định tọa độ của cạnh biên

Xét lần lượt các tọa độ đỉnh của đa giác. Qua mỗi đỉnh ta vẽ đường thẳng song song với trục hoành. Tính giao điểm của đường thẳng này với cạnh của đa giác. Thí dụ trên hình 4, khi xét tọa độ của điểm  $L_{x_3, y_3}$  ta phải tính giao điểm tại  $x_5, y_5$ .



Hình 4. Giao điểm của cạnh đa giác với đường phân băng

b) Tìm các đỉnh của hình chữ nhật đơn vị từ các cạnh băng vừa tạo ra

Trên đây đã tạo ra các giao điểm của đường phân băng và các cạnh đa giác, đó là các điểm  $(x_5, y_5)$ ,  $(x_6, y_6)$  và  $(x_7, y_7)$  như trên hình 4. Để tiện cho cài đặt bằng ngôn ngữ lập trình, hãy lập cấu trúc dữ liệu như mô tả sau (bằng ngôn ngữ C++) để lưu trữ và phân biệt chúng.

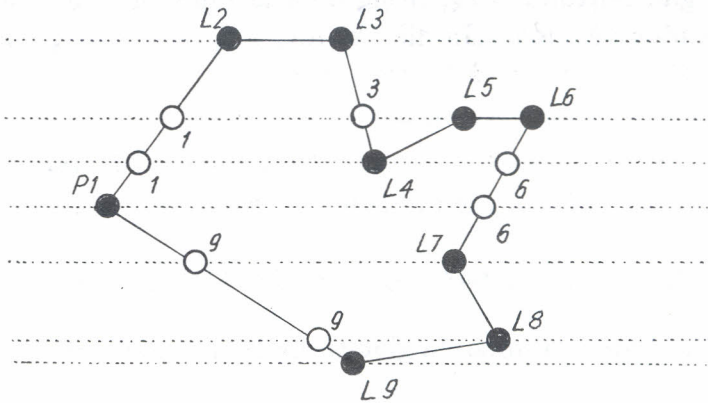
```
typedef struct {
    double x, y; // tọa độ của đỉnh đa giác hay giao điểm
    int attr; // có giá trị CONVEX nếu chúng là đỉnh của đa giác.
                // có giá trị INSECT nếu chúng là giao điểm cạnh đa giác
                // với đường phân băng
    unsigned index; // nếu attr = CONVEX thì có thứ tự đỉnh đa giác
                    // nếu attr = INSECT thì lưu thứ tự của đỉnh đa giác
                    // trước đó
}Point_st;
```

Như vậy, từ giá trị *attr* và *index* của một điểm ta có thể xác định được cạnh băng để tạo hình chữ nhật đơn vị. Các hình chữ nhật đơn vị được lập như sau:

- Nếu thuộc tính *attr* của một điểm là CONVEX thì chúng được nối với điểm có thuộc tính là CONVEX và giá trị *index* lớn hơn hay nhỏ hơn nó 1 đơn vị, và nối với điểm có thuộc tính là INSECT có giá trị *index* bằng chính nó hay nhỏ hơn nó 1 đơn vị.
- Nếu thuộc tính *attr* của một véc tơ là INSECT thì chúng được nối với điểm có thuộc tính là CONVEX và giá trị *index* bằng nó hay lớn hơn

nó 1 đơn vị, và nối với điểm có thuộc tính là INSECT có giá trị *index* bằng chính nó.

- Xét riêng cho trường hợp điểm và điểm cuối của đa giác.



Hình 5. Gán chỉ số cho các giao điểm

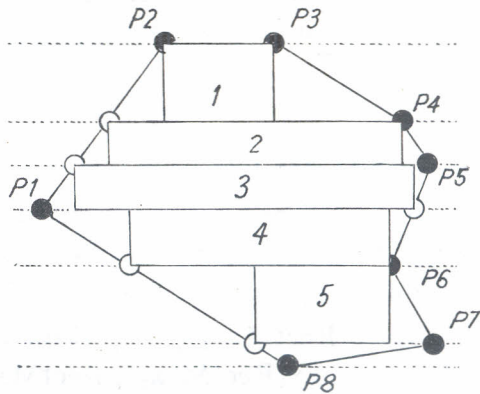
## 2. Xác định hình chữ nhật lớn nhất của một miền đa giác lồi

Một miền đa giác lồi theo phương nằm ngang là một miền đa giác mà khi ta kẻ một đường thẳng nằm ngang qua miền đó thì chúng sẽ cắt các cạnh của đa giác nhiều nhất tại 2 điểm. Các bước để xác định số hiệu thanh đứng phía trái (*left*), số hiệu thanh đứng phía phải (*right*) và số hiệu các đường phân băng trên và dưới (*bottom* và *top*) của hình chữ nhật đơn vị *box* (*left*, *right*, *bottom*, *top*) như sau:

a) Sắp xếp các hình chữ nhật đơn vị theo vị trí từ trên xuống dưới (xem hình 6).

b) Xét lần lượt các hình chữ nhật đơn vị để có được các giá trị *left*, *right*, *top*, *bottom* của hình chữ nhật nội tiếp.

- Gọi hình chữ nhật đơn vị thứ *i* là  $Rect_i$ .
- Gọi  $l_i, t_i, r_i, b_i$  là tọa độ trái, đỉnh, phải, đáy của  $Rect_i$ .
- Gọi  $RectMax_i$  là hình chữ nhật nội tiếp tìm được từ hình chữ nhật đơn vị *i*.
- Gọi  $left_i, top_i, right_i, bottom_i$  là tọa độ trái, đỉnh, phải, đáy của  $RectMax_i$ .



Hình 6. Các hình chữ nhật đơn vị

Các tọa độ đỉnh  $left_i, top_i, bottom_i$  của  $RectMax_i$  được tính như sau:

i/ Gán  $left_i = l_i$ .

ii/ Gán  $j = 1$  và  $top_i = t_j$  nếu  $i = 1$ ; ngược lại, gán  $top_i = b_j$ , trong đó *j*

là hình chữ nhật đơn vị đầu tiên từ vị trí  $i$  duyệt lên trên và thỏa mãn điều kiện  $l_j > l_i$ , nếu không tìm được  $j$  thỏa mãn điều kiện đó thì  $\text{top}_i = t_j$  với  $j = i$ .

iii/ Gán  $k = i$  và  $\text{bottom}_i = b_k$  nếu  $i = N - 1$  (hình chữ nhật đơn vị cuối cùng); ngược lại, gán  $\text{bottom}_i = t_k$ , trong đó  $k$  là hình chữ nhật đơn vị đầu tiên từ vị trí  $i$  duyệt xuống và thỏa mãn điều kiện  $l_k > l_i$ ; nếu không tìm được  $k$  thỏa mãn điều kiện đó thì gán  $k = i$  và  $\text{bottom}_i = b_k$ .

iv/ Gán  $\text{right}_i = \min(r_m)$  với  $m = j, \dots, k$ .

Điều kiện  $P_{\text{inc}}$  tự động được thỏa mãn trong quá trình trên.

Hình chữ nhật lớn nhất từ các hình chữ nhật nội tiếp vừa tìm được trên đây là:

$$\text{RectMax} = \max(\text{RectMax}_i) \text{ với } i = 1, \dots, N - 1$$

Ví dụ: Kết quả tìm được với miền đa giác trong hình 6 được thể hiện trong bảng sau:

Hình chữ nhật đơn vị ( $\text{Rect}_i$ )	Hình chữ nhật lớn nhất tạo được từ $\text{Rect}_i$ ( $\text{RectMax}_i$ )			
	$\text{left}_i$	$\text{top}_i$	$\text{right}_i$	$\text{bottom}_i$
$\text{Rect}_1$	$l_1$	$t_1$	$r_1$	$b_1$
$\text{Rect}_2$	$l_2$	$t_2$	$r_2$	$b_3$
$\text{Rect}_3$	$l_3$	$t_3$	$r_3$	$b_3$
$\text{Rect}_4$	$l_4$	$t_2$	$r_4$	$b_4$
$\text{Rect}_5$	$l_5$	$t_2$	$r_5$	$b_5$

\* Trường hợp đặc biệt:

a) Giả sử trong bước ii), khi duyệt lên ta không tìm được  $\text{Rect}_j$  nào thỏa mãn điều kiện:  $l_j = l_i$  mà ta tìm được một số  $\text{Rect}_j$  thỏa điều kiện:  $l_j > l_i$  thì ta chọn tọa độ  $\text{top}_i$  lần lượt là các  $b_j$  và ta có được một tập hợp:

$\{\text{RectMax}_{i1}, \text{RectMax}_{i2}, \dots\}$ . Hình chữ nhật lớn nhất tìm được từ  $\text{Rect}_i$  là:  $\text{RectMax}_i = \max(\text{RectMax}_{i1}, \text{RectMax}_{i2}, \dots)$ .

b) Giả sử trong bước iii/, khi duyệt xuống ta không tìm được  $\text{Rect}_k$  nào thỏa điều kiện:  $l_k > l_i$  mà ta tìm được một số  $\text{Rect}_k$  thỏa điều kiện  $l_k = l_i$  thì ta chọn tọa độ  $\text{bottom}_i$  lần lượt là các  $t_k$  và có được một tập hợp:  $\{\text{RectMax}_{i1}, \text{RectMax}_{i2}, \dots\}$ . Hình chữ nhật lớn nhất tìm được từ  $\text{Rect}_i$  là:  $\text{RectMax}_i = \max(\text{RectMax}_{i1}, \text{Rectmax}_{i2}, \dots)$ .

Vì các điều kiện bất biến chỉ đúng cho một miền đa giác lồi. Đối với những miền đa giác lõm (không lồi), thuật toán được cải tiến theo các bước sau:



i/ Tách miền đa giác không lồi thành những miền đa giác lồi đơn vị theo phương nằm ngang.

ii/ Tìm hình chữ nhật lớn nhất trong từng miền đa giác lồi đơn vị.

iii/ Hình chữ nhật lớn nhất trong các hình chữ nhật lớn nhất của các miền đa giác lồi đơn vị là hình chữ nhật cần tìm.

### 3. Phương pháp tách một miền đa giác bất kỳ thành các miền đa giác lồi theo phương nằm ngang

#### Bài toán:

Cho:  $E = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  là một miền đa giác bất kỳ.

Mục đích: Chia miền đa giác  $E$  thành các miền đa giác con  $E_i$  sao cho  $E_i$  là các miền không lồi theo phương nằm ngang lớn ở mức có thể có trong miền đa giác  $E$ .

#### Thuật toán:

- Gọi hai hình chữ nhật là “dính” nhau nếu chúng liên thông và top của hình này trùng với bottom của hình kia.
- Coi miền  $E_i = \{\text{Rect}_x, \text{Rect}_y, \text{Rect}_z, \dots\}$  là tập hợp các hình chữ nhật đơn vị thứ  $x, y, z, \dots$ , chúng thỏa mãn điều kiện sau:  $\text{Rect}_x$  dính  $\text{Rect}_y$ ,  $\text{Rect}_y$  dính  $\text{Rect}_z, \dots$

Các bước của thuật toán sẽ là:

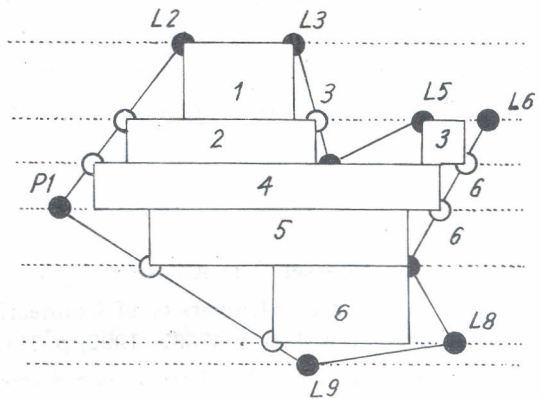
i/ Chia đa giác  $E$  theo từng băng nằm ngang.

ii/ Tìm các hình chữ nhật đơn vị trong miền đa giác  $E$ .

iii/ Khởi đầu số miền  $E_i$  bằng số hình chữ nhật đơn vị trong băng thứ nhất, và mỗi miền  $E_i$  có phần tử đầu tiên là các hình chữ nhật đơn vị tương ứng.

iv/ Duyệt với băng tiếp theo. Với mỗi hình chữ nhật đơn vị ở băng mới ta kiểm tra:

- nếu nó dính với hình chữ nhật đơn vị cuối cùng của một miền  $E_i$  thì ghép nó vào miền đó và nó trở thành phần tử cuối cùng của một miền  $E_i$ ,
- nếu nó không dính với một phần tử cuối cùng nào của miền đã tạo thì tạo một miền mới (số miền tăng thêm 1) với phần tử đầu tiên là hình



Hình 7. Các hình chữ nhật đơn vị tạo được với một miền đa giác lồi theo phương nằm ngang

chữ nhật đơn vị này.

v/ Lặp lại việc duyệt bằng theo bước iv/ cho đến bằng cuối cùng.

Ví dụ: Đối với miền đa giác không lồi trong hình 7, kết quả tách các miền đa giác lồi đơn vị như sau:

$$E_1 = \{\text{Rect}_1, \text{Rect}_3, \text{Rect}_4, \text{Rect}_5, \text{Rect}_6\}$$

$$E_2 = \{\text{Rect}_2, \text{Rect}_3, \text{Rect}_4, \text{Rect}_5, \text{Rect}_6\}$$

#### IV. CÀI ĐẶT VÀ KẾT LUẬN

Để sử dụng có hiệu quả thuật toán thì phải xác định số các hình chữ nhật được tạo ra. Mỗi hình chữ nhật chỉ được tạo ra trong vòng lặp tạo nhóm, vòng lặp này được đặt trong vòng lặp tính các thanh và các cạnh biên. Số các hình chữ nhật phụ thuộc vào số thanh đứng và số cạnh biên. Còn các thanh đứng và cạnh biên phụ thuộc vào số đỉnh của đa giác. Ta đánh giá thuật toán cho hai trường hợp đa giác lồi và đa giác không lồi.

Đối với đa giác lồi  $N$  đỉnh, số các bằng lớn nhất có thể tạo ra là  $N - 1$  vậy số các cạnh biên lớn nhất tính được là  $2 \times (N - 1)$ . Số các hình chữ nhật đơn vị lớn nhất là  $N - 1$ . Hơn nữa, mỗi hình chữ nhật đều có khả năng tạo nhóm với những hình chữ nhật đơn vị khác nếu lấy cạnh trái của nó làm chuẩn. Quá trình tạo nhóm sẽ phải duyệt lại qua các  $(N - 2)$  hình chữ nhật đơn vị đối với mỗi hình chữ nhật đang xét. Như vậy, đối với đa giác lồi độ phức tạp của thuật toán là  $O(N^2)$  bởi vì số hình chữ nhật đơn vị lớn nhất là  $N - 1$  với mỗi hình chữ nhật đơn vị ta phải duyệt qua nhiều nhất  $N - 2$  hình chữ nhật đơn vị còn lại để tạo nhóm.

Đối với đa giác không lồi cũng tương tự như trên, nhưng số hình chữ nhật đơn vị là  $(N - 1) \times (N - 1)$ . Số bằng lớn nhất ta có thể tạo ra là  $N - 1$  bằng tương ứng với mỗi bằng có thể có lớn nhất  $N - 1$  cạnh bằng đối với đa giác gấp khúc. Như vậy độ phức tạp của thuật toán khi đa giác không lồi là  $O(N^3)$ .

#### TÀI LIỆU THAM KHẢO

1. Jan W. van Roessel, The American Cartographer, Vol. 16, No. 3 (1989) 201-109.
2. Robert G. Cromley, University of Connecticut, Digital Cartography, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1992, p. 311.
3. United Nations DESIPA, Population Software Notes, NewsLetter, New York, USA, 1996.

1) Viện Công nghệ thông tin  
Trung tâm KHTN và CNQG.

2) Trường Đại học Bách khoa Hà Nội.

Nhận bài ngày 2-8-1997