

# MỘT VÀI PHƯƠNG PHÁP HIỆU CHỈNH DỮ LIỆU BẢN ĐỒ SỐ HOÁ

ĐẶNG VĂN ĐỨC

This paper deals with an algorithm for the reduction of the numbers of the points to represent a digitized line and a method of processing the spatial conflicts in the line generalization which are used to design an integrated software for geographical information, maps and graphics database. The methods have been used in the development work of a new software called PopMap by Vietnam IOIT for United Nations Department for Economic and Social Information and Policy Analysis (DESIPA).

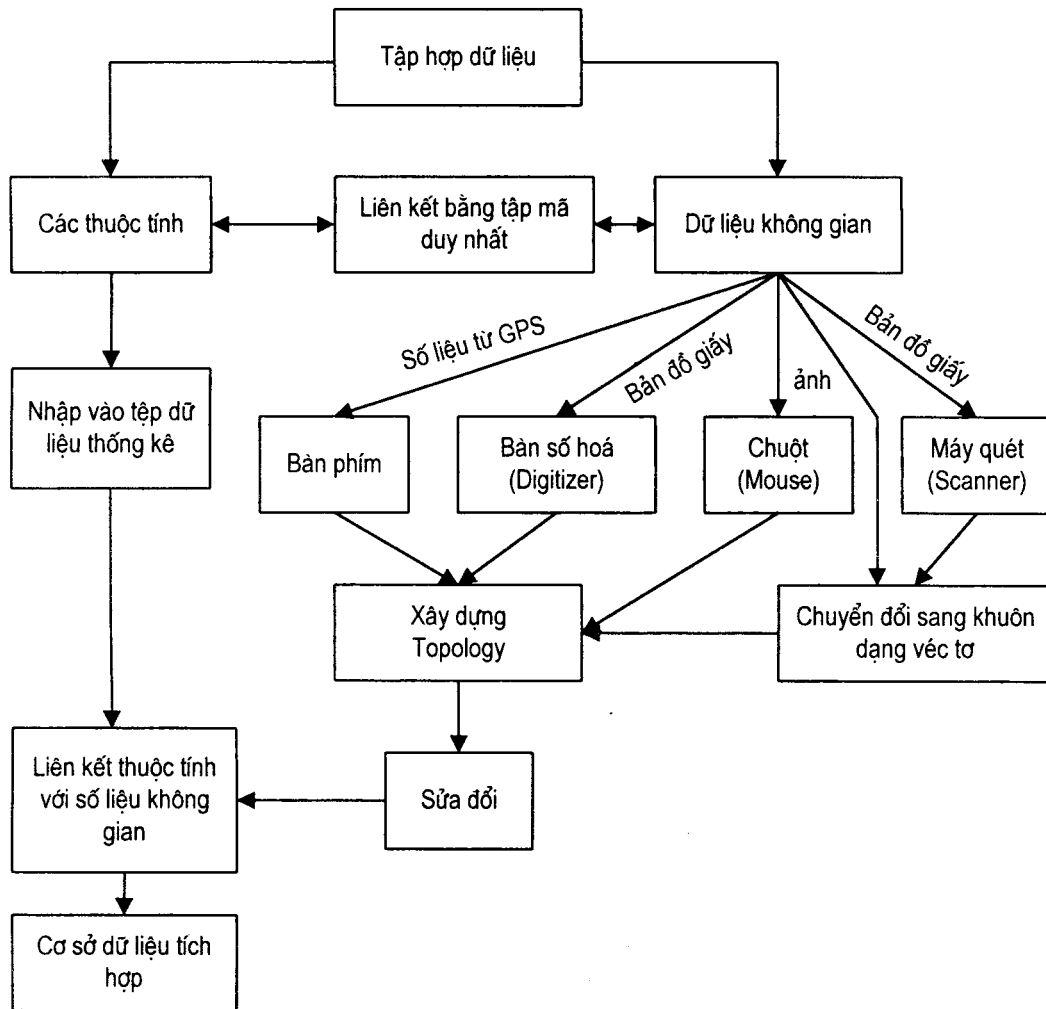
## I. VẤN ĐỀ

Phần lớn các dữ liệu của các hệ thống tin học đều liên quan đến vị trí địa lý. Các hệ thống thông tin tích hợp (HTTTTH) của các dữ liệu địa lý và các dữ liệu phi hình học ngày càng trở thành chuẩn mực của một hệ thống tin học.

Việc nghiên cứu các mô hình dữ liệu và hệ thống cho phù hợp với một lớp ứng dụng riêng đã và đang được tiến hành ở nhiều nơi trong và ngoài nước. Một sơ đồ tiến trình xây dựng cơ sở dữ liệu của HTTTTH được mô tả trên hình 1. Đầu vào của một HTTTTH gồm dữ liệu thống kê và bản đồ. Để dễ dàng thực hiện các phép tính không gian trên bản đồ theo các tỷ lệ khác nhau... thì bản đồ phải được lưu trữ dưới dạng véctor. Bản đồ giấy phải được số hoá bằng bàn số hoá (digitizer) hay máy quét (scanner). Số liệu từ đầu ra của máy quét phải được chuyển sang khuôn mẫu véctor là các đối tượng điểm. Các điểm được nối với nhau để tạo thành các lớp (layer) địa lý. Từ khối dữ liệu sơ khai có thể có rất nhiều điểm hoặc/và đường bị dư thừa. Chúng làm giảm tốc độ hiển thị bản đồ và làm tăng đáng kể dung lượng bộ nhớ để lưu trữ. Bài báo này đề cập đến một số phương pháp loại bỏ các thông tin dư thừa hay làm sạch dữ liệu bản đồ trong khuôn dạng véctor.

## II. MỘT SỐ THUẬT TOÁN LÀM GIẢM SỐ ĐIỂM BẢN ĐỒ SỐ HOÁ

Tất cả các phương pháp số hoá bản đồ đều phát sinh ra nhiều điểm, đoạn thẳng hơn số lượng cần thiết. Giả sử bản đồ được số hoá bằng bàn số hoá (digitizer). Số liệu mà máy tính nhận được từ bàn số hoá ở một thời điểm là tọa độ của một lớp (layer), thí dụ đường biên hành chính. Nhiệm vụ của chương trình biểu diễn số liệu bản đồ là phải



Hình 1. Tiến trình thiết lập cơ sở dữ liệu tích hợp

nối các điểm lại thành các chuỗi để tạo ra các lớp (đường biên) như trên bản đồ giấy. Các layer địa lý được tạo ra như một dãy liên tục của các điểm nối với nhau từng đôi một.

$$E : \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

trong đó:  $x, y$  là tọa độ hai chiều.

Thuộc tính	Toạ độ
P	$x_1, y_1$
L	$x_2, y_2$
L	$x_3, y_3$
L	$x_4, y_4$
P	$x_5, y_5$

Các toạ độ nói trên phát sinh do người sử dụng nhấn phím của bàn số hoá một cách ngẫu nhiên. Vì vậy, hai điểm liên tiếp được phát sinh ra có thể trùng nhau, có thể cùng nằm trên đường thẳng, hay có thể gần nhau đến mức không cần thiết. Thí dụ trên đây là các số liệu của bản đồ được lưu lại trong tệp. Thuật tính P có nghĩa là chấm một điểm. Thuộc tính L có nghĩa là phải nối nó với toạ độ trước đó để thành đoạn thẳng. Các trường hợp sau đây có thể xảy ra:

a)  $x_2 = x_3$  và  $y_2 = y_3$  hay điểm  $(x_2, y_2)$  và  $(x_3, y_3)$  trùng nhau, có thể loại bỏ một trong hai điểm đó.

b) Các điểm  $(x_1, y_1)$ ,  $(x_2, y_2)$  và  $(x_3, y_3)$  cùng nằm trên đường thẳng, như vậy điểm  $(x_2, y_2)$  có thể là dư thừa.

c) khoảng cách giữa hai điểm  $(x_3, y_3)$  và  $(x_4, y_4)$  quá gần nhau, đến mức hai điểm có thể thay bằng một điểm.

Sau đây là một số phương pháp loại bỏ các điểm dư thừa trong cơ sở dữ liệu bản đồ.

1. Phương pháp 1 đơn giản, dễ cài đặt, song thời gian xử lý cho toàn bộ bản đồ còn chậm và chưa loại bỏ triệt để các toạ độ dư thừa. Trong phương pháp này, tất cả các điểm phải được xét duyệt để xem chúng có bị đụng độ với các điểm còn lại không. Các bước thực hiện của phương pháp I như sau:

- Xếp xếp lại các toạ độ theo thứ tự tăng dần theo trục  $y$  và theo trục  $x$ .
- Ta kí hiệu toạ độ bắt đầu của một râu là  $p$ , các toạ độ để tạo ra đoạn thẳng có kí hiệu là  $L$ . Nếu toạ độ đang xét là  $P$ , toạ độ tiếp theo trùng với toạ độ đang xét thì loại bỏ chúng. Nếu toạ độ đang xét là  $L$ , toạ độ tiếp theo trùng với toạ độ đang xét có kí hiệu là  $L$  thì loại bỏ. Nếu chúng là  $P$  thì phải giữ lại vì chúng là toạ độ khởi đầu của một râu mới.
- Nếu điểm đang xét có kí hiệu là  $L$  nằm trên đường thẳng tạo bởi hai toạ độ trước và sau chúng cùng có kí hiệu  $L$  và không có kí hiệu  $P$  nào nữa gán cho toạ độ này thì toạ

độ đang xét bị loại bỏ. Từ tọa độ trước và tọa độ sau điểm đang xét ta tính góc  $\theta$ . ( $\theta$  là góc tạo ra bởi đường thẳng và trục hoành). Sau đó tính góc  $\theta_1$  của đường thẳng tạo ra bởi điểm đang xét ta tính góc và điểm trước nó. Nếu hai góc  $\theta, \theta_1$  bằng nhau thì ba điểm nằm trên đường thẳng.

- Nếu khoảng cách hai điểm có cùng kí hiệu là  $L$  nhỏ hơn một số  $T$  cho trước thì loại bỏ một trong hai điểm. Khoảng cách của hai tọa độ  $a(x_1, y_1)$  và  $b(x_2, y_2)$  được tính như sau:

$$|a - b| = [(x_1 - x_2)^2 + (y_1 - y_2)^2]^{1/2}.$$

Cài đặt bằng chương trình:

```
typedef struct
```

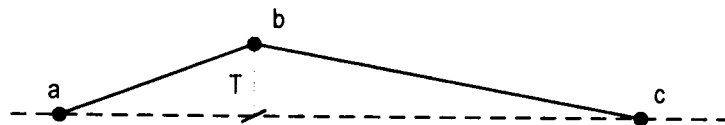
```
{
    int x
    int y
}Point2;
```

```
double DistanceBetween2Points(Point2*a,Point2*b)
```

```
{
    double dx, dy;
    dx = a->x - b->x;
    dy = a->y - b->y;
    return(sqrt((dx*dx) + (dy*dy)));
}
```

- Tiếp tục thực hiện cho đến cuối tệp.

Phương pháp này đã loại bỏ được một số lượng đáng kể các điểm dư thừa trong tập dữ liệu bản đồ. Tuy nhiên chúng chưa có khả năng loại bỏ các điểm như trên hình sau

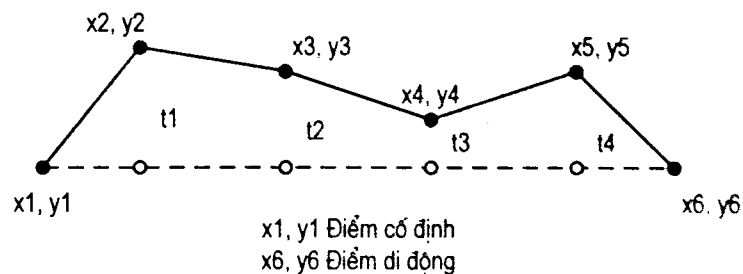


Hình 2.

Nếu khoảng cách  $T$  trên hình hai nhỏ hơn giá trị cho phép thì các đoạn thẳng  $ab, bc$  được thay bằng đoạn thẳng  $ac$  có nghĩa là điểm  $b$  bị loại bỏ.

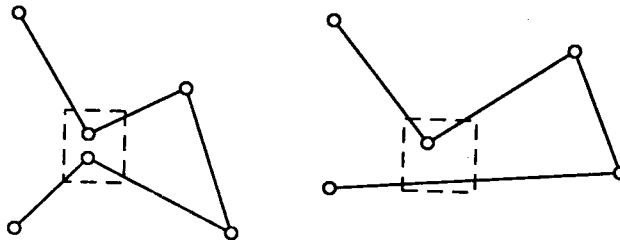
2. Phương pháp 2 thực hiện nhanh hơn phương pháp trên đây. Ta xem xét từng khâu của lớp bản đồ. Xâu (chain) là các đoạn thẳng nối hai node. Node là giao điểm của các đường thẳng hay điểm cuối, chúng không phải là điểm trung gian khi tạo chain. Phương pháp này bắt đầu bằng việc tạo ra các bảng node, chain từ tập dữ liệu sơ khai. Thuật toán loại bỏ các điểm dư thừa được mô tả như sau:

Toạ độ thứ nhất của xâu được coi là điểm chốt và toạ độ cuối của xâu là điểm di động. Hai điểm này tạo ra một đoạn thẳng. Trên hình 3 là đoạn thẳng  $(x_1, y_1), (x_2, y_2)$ . Tính các khoảng cách của các điểm nằm trong hai điểm chốt và điểm di động tới đoạn thẳng tạo bởi điểm chốt và điểm di động, đó là các khoảng cách  $t_1, t_2, t_3$  và  $t_4$ . Nếu tất cả các khoảng cách đó đều nhỏ hơn một giá trị  $T$  cho phép thì đoạn thẳng nối giữa điểm chốt và điểm di động được loại bỏ. Nếu điều kiện không thoả mãn, có nghĩa là có điểm phải được giữ lại. Điểm có khoảng cách lớn hơn giá trị quy định  $T$  trở thành điểm di động mới. Công việc lại tiếp tục với đoạn thẳng nối từ điểm cố định đến điểm di động mới. Điểm di động mới được chọn là điểm không thoả mãn điều kiện là có khoảng cách nhỏ hơn  $T$  và ở xa điểm chốt nhất. Trên hình 3 điểm  $(x_5, y_5)$  được chọn. Một khi khoảng cách cực đại của các điểm đến đoạn thẳng nhỏ hơn  $T$  thì điểm cố định mới sẽ là điểm di động hiện hành và điểm di động được gán lại. Nếu  $t_1, \dots, t_4 < T$  thì toạ độ  $(x_6, y_6)$  sẽ là điểm cố định mới. Chu kỳ xử lý được lặp lại, kết quả là tất cả các điểm đã đóng vai trò là điểm cố định được giữ lại cho lớp bản đồ.



Hình 3.

Để tăng tốc độ xử lý, một ngăn xếp được sử dụng. Tất cả các toạ độ đã được chọn làm điểm di động trong chu kỳ xử lý kể trên đều được đưa vào ngăn xếp. Khi điểm cố định chuyển đến vị trí mới, điểm di động mới sẽ là toạ độ trong đỉnh của ngăn xếp. Như



Hình 4. Dụng độ

Điểm-điểm (trái)

Điểm-đoạn thẳng (phải)

vậy không cần tính toán lại cho toàn bộ các điểm nằm trong khoảng từ điểm cố định đến điểm di động. Khi cài đặt ta phải tính khoảng cách  $d$  từ một điểm  $t(x_i, y_i)$  đến một đoạn thẳng  $Ax + By + C = 0$  nối hai điểm  $a, b$  và được tính như sau (theo tài liệu [6],[7]):

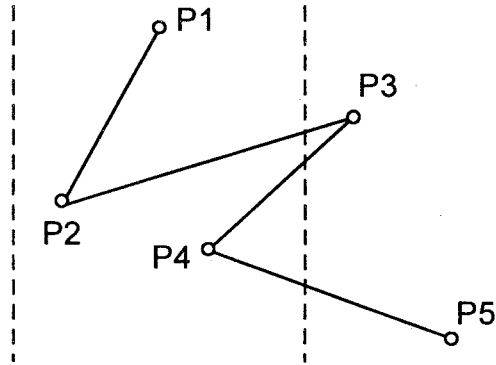
$$d = |s| / (A^2 + B^2)^{1/2}$$

trong đó

$$s = Ax_i + By_i + C.$$

### III. MỘT PHƯƠNG PHÁP TÌM KIẾM ĐIỂM VÀ ĐƯỜNG DỤNG ĐỘ TRONG DỮ LIỆU BẢN ĐỒ SỐ KHOÁ

Giả sử bản đồ được số hoá và được lưu trữ trong tệp theo khuôn dạng được mô tả trên. Khi xử lý bản đồ như co dẫn để hiển thị ra màn hình, chuyển ra máy in,... rất dễ xảy ra hiện tượng dụng độ giữa điểm-điểm và điểm-đoạn thẳng như mô tả trên hình 4. Dụng độ ở đây có nghĩa là một điểm nào đó của bản đồ bị chồng lên hay ở rất gần một điểm khác hay một đoạn thẳng. Theo thực nghiệm của Swiss Society of Cartography 1987 thì với bản đồ in đen trắng, khoảng cách đọc 30cm thì các đường hay điểm phải cách nhau ít nhất là 0.25mm. Vậy nếu khoảng cách còn 0.20mm thì chúng được coi là bị dụng độ. Vì số lượng điểm và đường của một bản đồ là rất lớn, cho nên cần phải có một phương pháp tìm kiếm nhanh để chỉ ra các điểm dụng độ trong một khoảng thời gian có thể chấp nhận được. Giả sử số liệu bản đồ vào từ bàn số hoá hay số liệu vào từ máy quét và đã chuyển sang khuôn dạng véctor được ghi liên tục cho một xâu như sau:  $P_1, P_2, P_3, P_4, P_5$  (xem hình 5). Trước khi thực hiện thuật toán, số liệu được sắp xếp lại theo chiều tăng dần của giá trị  $x$ :  $P_1, P_2, P_3, P_4, P_5$

Hình 5. Tìm đựng độ Điểm-điểm cho  $P_1$ 

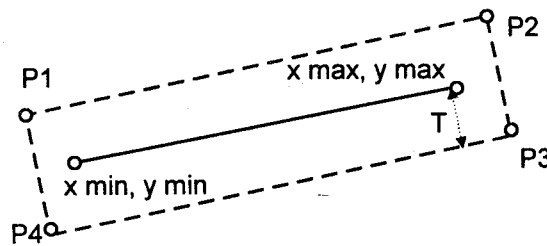
Việc tìm kiếm đựng độ điểm-điểm cho  $P_1$  được thực hiện theo các bước sau đây:

- a) Bắt đầu từ điểm  $P_1$  trong danh sách đã sắp xếp như trình bày trên đây.
- b) Tìm theo chiều bên trái (với  $x$  giảm dần) cho đến khi xác định được một giá trị  $x$  vượt quá giá trị  $T$  cho trước ( $T$  là giới hạn của đựng độ). Chỉ số của điểm ngay trước đó được lưu lại như giá trị cận dưới. Trong hình trên là chỉ số của điểm  $P_2$ .
- c) Thực hiện tương tự theo chiều về phải (với  $x$  tăng dần). Chỉ số của điểm ngay trước đó được lưu lại như giá trị cận trên, chỉ số của điểm  $P_4$ .
- d) Giá trị của chỉ số cận trên và cận dưới tạo thành miền chứa các điểm có thể đựng độ với  $P_1$  do tọa độ  $x$ . Tính khoảng cách của từng điểm này đến  $P_1$  theo công thức Euclid như đã trình bày ở phần trên. Nếu khoảng cách đó nhỏ hơn  $T$  thì điểm vừa tính là đựng độ với  $P_1$ .

Việc tìm kiếm đựng độ điểm-đoạn thẳng được thực hiện tương tự như trên. Song ở đây, phải xác định được một hình chữ nhật bao quanh đoạn thẳng đang xét (xem hình 6). Để cho an toàn, chiều cao và chiều rộng của hình chữ nhật được tính như sau:

- a) Tính giá trị  $\min x, y$  và  $\max x, y$  của đoạn thẳng đang xét.
- b) Tọa độ góc dưới, trái của hình chữ nhật là:  $\min x - T, \min y + T$ . Tọa độ trên, bên phải của hình chữ nhật là  $\max x + T, \max y - T$ .

Tất cả các điểm nằm trong hình chữ nhật đều bị đựng độ với đường đang xét. Vì số liệu đã được xếp sắp theo chiều tăng của  $x$ , cho nên số lượng điểm phải xét là giới hạn. Việc tìm ra các điểm đựng độ đã giúp hiệu chỉnh lại số liệu sao cho bản đồ hiển thị một cách rõ ràng.



Hình 6. Tìm kiếm độ Đỉnh-đường

#### IV. CÀI ĐẶT

Các phương pháp làm sạch số liệu bản đồ trên đây đã được cài đặt bằng ngôn ngữ C PopMap trong môi trường DOS và Windows cho một số phần mềm trong đó có phần mềm PopMap do Viện công nghệ thông tin thiết kế và lập trình theo đơn đặt hàng của Ban thông tin kinh tế xã hội và phân tích chiến lược của Liên hiệp quốc. Một phần mềm đã và đang được sử dụng trong khoảng 100 nước (xem [2]).

#### TÀI LIỆU THAM KHẢO

1. DESIPA, PopMap User's guide and reference manual, United Nations, New York, 1994.
2. Population Software Notes, United Nations DESIPA, New York, January 1996.
3. Keith C. Clarke, Analytical and computer cartography, Prentice Hall, New Jersey, USA, 1990.
4. M. Goodchild, K. Kemp, Technical issues in GIS, University of California, 1990.
5. A. Robinson, R. Sale, J. Morrison, P. Muehrcke, Elements of Cartography, John Wiley & Sons, USA, 1984.
6. J. Arvo, Graphics GEMS II, Academic Press, INC. USA, 1991.
7. R. Cromley, Digital Cartography, Prentice Hall, New Jersey, USA, 1992.

Viện Công nghệ thông tin

Nhận ngày 12, 2 1996