TỔNG QUAN: Các tiến bộ gần đây trong công nghệ quét ảnh đã làm cho việc quét các đường nét trên hệ thống giá rẻ trở nên khả thi và đưa ra dưới dạng hình ảnh raster. Các hình ảnh này có thể được chỉnh sửa, tự động mảnh hơn thành các đường liên thông và chuyển đổi thành các biểu diễn vector bằng cách sử dụng các kỹ thuật xử lý ảnh thông thường. Một nghiên cứu về các kỹ thuật chỉnh sửa raster, bao gồm mảnh hơn, điền vào và phát hiện điểm, đã được tiến hành bằng cách sử dụng phần mềm chuyên dụng. Các kỹ thuật này dựa trên việc mã hóa trạng thái của một vùng xung quanh ba-cấu-mố và một điểm thành một byte duy nhất. Byte này có giá trị từ 0 đến 255 và lưu trữ tất cả các điều kiện xung quanh có thể xảy ra cho một pixel. Việc triển khai các phương pháp khác nhau để mảnh hơn, điền vào và phát hiện điểm đã được đơn giản hóa đáng kể bằng cách sử dụng các đặc điểm xung quanh này. Ngoài ra, cũng có thể phát triển và kiểm tra bảng quyết định cho các nhiệm vụ chỉnh sửa raster và đảm bảo tính đối xứng và nhất quán trong việc áp dụng chúng. Một phương pháp mẫu để chuyển đổi các đường liên thông raster đã được chỉnh sửa thành vector cũng đã được phát triển. Khi các biểu diễn vector của các đường đã được hình thành, chúng được định dạng dưới dạng Đồ thị Đường số, và được rà soát kỹ hơn bằng cách xóa các đỉnh không cần thiết và làm mịn bằng một kỹ thuật phù hợp với đường cong.

Introduce

Các nỗ lực chuyển đổi từ raster sang vector đã đặt ra những thách thức cho các nhà nghiên cứu trong nhiều năm qua. Peuquet (1981) cung cấp một cái nhìn tổng quan về các phương pháp đã được phát triển cho quá trình chuyển đổi raster sang vector. Dựa trên một số kỹ thuật có sẵn, bài báo này mô tả việc sử dụng và tinh chỉnh một số phương pháp để xử lý linework dưới dạng raster và vector. Các khía cạnh tự động của quá trình chuyển đổi raster sang vector được mô tả, nhưng các bước chỉnh sửa tương tác vượt quá phạm vi của nghiên cứu này.

Bài báo cũng cố gắng thu gọn khoảng cách giữa các nghiên cứu lý thuyết về xử lý raster sang vector và các hệ thống sản xuất khóa sẵn đang được sử dụng nhưng thường không được hiểu rõ. Trong quá trình này, các kỹ thuật đã được phát triển để tạo ra các khe hở, chốt, mũi nhọn, tuyết, trượt điểm, và các hiện tượng không mong muốn khác của quá trình xử lý tự động. Qua trải nghiệm này, đã có được sự hiểu rõ hơn về các kỹ thuật tự động hoạt động tốt nhất cho một ứng dụng cụ thể và một số hiểu biết về sự kết hợp tối ưu của các kỹ thuật chỉnh sửa tự động và thủ công.

Background

Gần đây, Jenson (1985) đã phát triển các kỹ thuật để trích xuất các lưu vực thủy văn và định rõ các mạng dẫn nước từ các mô hình định dạng raster của mô hình độ cao số học (OEM). Trong việc áp dụng các kỹ thuật này vào công việc dự án, đã xác định rằng việc chuyển đổi các đường dẫn nước sang định dạng vector sẽ tốt nhất để cho phép việc chồng chéo hiệu quả với dữ liệu được định vị khác, đo độ dài và diện tích, và sắp xếp các dòng sông. Có sẵn các kỹ thuật để chuyển đổi các lưu vực thủy văn thành dạng đa giác (Nichols, 1981), nhưng không có phương pháp để chuyển đổi các đường dẫn nước raster sang dạng vector.

Cần một phương pháp để chuyển đổi linework được định dạng raster, được trích xuất từ OEM, thành chuỗi vector hoặc cung có thể được đưa vào hệ thống thông tin địa lý (GIS) dựa trên vector. Đã nhận thấy rằng vấn đề này tương tự như vấn đề chuyển đổi linework từ bản đồ quét sang vector. Hy vọng rằng việc quét dữ liệu bản đồ trong tương lai có thể được tạo điều kiện tốt hơn thông qua việc phát triển một giải pháp chung cho vấn đề này và hiểu rõ hơn về các bước xử lý và chuyển đổi liên quan.

ENCODING OF TOPOLOGICAL SURROUNDS

Các linework được định dạng raster có thể được thể hiện dưới dạng hình ảnh nhị phân, trong đó các pixel được biểu diễn bằng số 1 nếu có một đường nét và số 0 cho không gian giữa các đường nét. Có nhiều kỹ thuật có sẵn để xử lý và tinh chỉnh các đường nét trong dạng này (Rosenfeld và Kak, 1982; Pavlidis, 1982). Hầu hết các kỹ thuật sử dụng một tập hợp các quy tắc để thao tác trên mỗi pixel dựa trên trạng thái của nó và của tám hàng xóm liền kề. Do các hoạt động xung quanh khu vực thường mất thời gian và tốn năng lượng tính toán, đặc biệt chú ý đã được dành cho vấn đề này.

Phương pháp mã hóa được mô tả ở đây được lấy cảm hứng từ Golay (1969), người đã mô tả 64 trạng thái xung quanh cho một cấu trúc dữ liệu hình lục giác. Bởi vì chúng ta thường làm việc với dữ liệu hình ảnh trong định dạng raster, kỹ thuật của chúng tôi mã hóa cho mỗi pixel một byte duy nhất gồm tám bit mô tả hoàn toàn trạng thái của tất cả các hàng xóm kề cận. Để thực hiện các hoạt động xử lý raster (ví dụ: điền vào, mảnh hơn hoặc phát hiện điểm), một tập hợp các quy tắc quyết định được phát triển và đặt trong một bảng với một mục cho mỗi trạng thái xung quanh có thể có. Sau đó, hình ảnh raster được xử lý theo cách sau:

(a) mã hóa trạng thái xung quanh cho mỗi pixel và lưu trữ nó (b) đối với mỗi pixel, (1) tìm quy tắc quyết định được tham chiếu bởi trạng thái xung quanh (2) thay đổi pixel nơi được chỉ định bởi quy tắc (3) nếu đã thay đổi, cập nhật các giá trị xung quanh cho các pixel hàng xóm (c) nếu có bất kỳ thay đổi nào được thực hiện trong lần lặp này, quay lại bước (b)

Bằng cách mã hóa các giá trị xung quanh trong lần lặp đầu tiên và chỉ cập nhật chúng khi có sự thay đổi, việc mã hóa không cần thiết có thể được tránh trong các lần lặp tiếp theo. Điều này có thể cải thiện đáng kể tốc độ xử lý, vì nhiều chức năng xử lý được áp dụng lặp đi lặp lại cho đến khi không có thay đổi nào xảy ra trong một lần lặp hoàn chỉnh.

Mã hóa có thể được thực hiện thủ công bằng cách cộng lại các giá trị được hiển thị trên sơ đồ sau cho các pixel hàng xóm có một đường nét hiện diện: [đính kèm sơ đồ không được cung cấp]

Mã hóa kỹ thuật số có thể được thực hiện bằng cách thiết lập các bit nhị phân tương ứng của một byte 8 bit. Trong cả hai trường hợp, giá trị kết quả chiếm trong phạm vi giá trị từ a đến 255, tức là có 256 trạng thái có thể có. Một ví dụ về một subroutine Fortran để thực hiện việc mã hóa này được hiển thị trong Hình 1.

Toàn bộ tập hợp 256 trạng thái xung quanh có thể có được hiển thị trong Hình 2. Trong hình này, các trạng thái xung quanh đã được nhóm lại để dễ dàng và nhất quán trong việc phát triển quyết định. 256 trạng thái xung quanh đã được kết hợp thành các nhóm mẫu bằng cách sắp xếp theo một tập các thuộc tính. Trong mỗi nhóm đếm (được đánh số từ 1 đến 8 và được tổ chức theo chiều dọc) là các nhóm con có thể chứa tối đa tám mẫu. Những nhóm con này được gọi là nhóm mẫu và có thể bao gồm một mẫu; các xoay 90°, 180° và 270° của nó; và hình ảnh và các xoay của nó. Do nhiều mẫu là đối xứng, chúng tạo ra các mẫu không duy nhất hoặc trùng lặp khi xoay. Kết quả là, một số nhóm con sẽ chỉ chứa bốn, hai hoặc ít nhất một mẫu duy nhất. Việc nhóm giúp tổng hợp 256 trạng thái xung quanh có thể quản lý được thành 49 nhóm mẫu. Không một trong các hoạt động được mô tả trong bài báo này yêu cầu phân chia dưới mức nhóm mẫu.

Mỗi nhóm mẫu có thể được đại diện bằng một tập các thuộc tính cho phép phát triển bảng quyết định một cách hiệu quả. Hình 3 là một bảng các nhóm mẫu và các thuộc tính liên quan. Một thuộc tính cơ bản là số hàng xóm có đường nét hiện diện. Để thực hiện các hoạt động mảnh hơn, hữu ích để biết xem mẫu được phân loại là đơn giản (Rosenfeld và Kak, 1982) hay không bội (Pavlidis, 1980). Điều kiện này đúng khi một pixel phải có chính xác một pixel hoặc một tập hợp kết nối các pixel kề. Landy và Cohen (1985) sử dụng thông tin tương tự, được gọi là số chuyển đổi từ đen sang trắng. Ngoài ra, họ xác định số đường nét kết quả khi pixel trung tâm bị xóa. Điều này được gọi là số khoảng trống, vì nó đo lường xem việc mảnh hơn có tạo ra một khoảng trống trong kết nối hay không. Những đặc điểm này là không đổi cho mỗi nhóm mẫu và được sử dụng để xác định và sắp xếp các nhóm được hiển thị trong Hình 2.

Raster operations

Đã thực hiện một số hoạt động raster trên hai tập dữ liệu ví dụ. Những ví dụ này thể hiện các hoạt động raster và vector khác nhau liên quan đến việc chuyển đổi từ raster-linework thành các đường cung vector.

Tập dữ liệu mẫu 1 là một mô hình độ cao kỹ thuật số (được hiển thị dưới dạng một hình ảnh sắc thái như trong Hình 4a). Hình 4b cho thấy linework đã được trích xuất từ DEM dưới dạng raster như được mô tả bởi Jenson (1985). Linework được mô tả dưới dạng một hình ảnh nhị phân với các đường nét được đại diện bằng số một và nền được đại diện bằng số không.

Tập dữ liệu mẫu 2 là một phần nhỏ của một bản đồ địa hình, được quét bằng thiết bị quét raster thương mại có sẵn. Hình 5a biểu thị hình ảnh gốc với 256 mức giá trị có thể, trong đó các đường nét có giá trị thấp hơn và nền hình ảnh có giá trị cao hơn. Trong Hình 5b, đã thực hiện một quá trình lọc convolution để cải thiện độ tương phản cục bộ của các đường nét so với nền. Điều này thường được gọi là tăng cường cạnh hoặc bộ lọc high-pass, và được mô tả bởi Moik (1980). Sau đó, dữ liệu được chuyển đổi thành một hình ảnh nhị phân bằng cách chọn một ngưỡng giá trị tốt nhất để phân tách các đường nét với nền (Hình 5c).

Filling operations

Mục đích của hoạt động lấp đầy là loại bỏ những khoảng trống nhỏ trong đường nét gây thêm sự phức tạp không cần thiết. Thuật ngữ "fill" cũng được sử dụng trong các ứng dụng đồ họa máy tính, nơi nó được hiểu là việc lấp đầy hoặc tô màu các vùng bên trong các đa giác. Mặc dù tương đối tương tự, nhưng quá trình này không phải là quá trình chúng ta cần trong việc chỉnh sửa tự động của các đường nét. Trong quá trình quét đường nét, có đôi khi tạo ra các đường gãy nhỏ hoặc khoảng trống trong các đường nét là những hiện tượng phụ thuộc vào quá trình quét hoặc do nhược điểm trong các đường nét ban đầu. Trong một số trường hợp, chúng ta muốn loại bỏ các khoảng trống bao gồm hợp lệ nhưng quá nhỏ để coi là đáng kể. Ngoài việc lọc ra các khoảng trống nhỏ, việc lấp đầy cũng có xu hướng tổng quát hóa và lấp đầy các đặc điểm góc nhọn trên các cạnh bên ngoài của đường nét. Điều này đôi khi có thể phù hợp, đặc biệt vì việc thưa dòng tiếp theo được thực hiện từ cạnh đường nét vào bên trong và bị ảnh hưởng nhiều bởi cạnh đường nét gồ ghề. Hình 4c cho thấy cách lấp đầy có thể được sử dụng để loại bỏ các lỗ nhỏ hoặc khoảng trống trong các đường dẫn mà đã được trích xuất từ OEM. Các quy tắc cho hoạt động lấp đầy được hiển thị trong Hình 3 và được đánh dấu Y cho các điều kiện xung quanh cho thấy rằng một ô trung tâm nên được đặt từ 0 (không có đường) thành 1 (đường).

Thinning operation

Đoạn văn trên đề cập đến các phương pháp để làm mảng raster trở nên mảnh hơn để tạo ra một đường viền liên kết duy nhất (thường được gọi là "line skeleton"). Đa phần các phương pháp này đều sử dụng quá trình lặp và một số sử dụng phương pháp xử lý hai bước. Một phương pháp thực hiện quá trình mảnh trong bước đầu tiên và loại bỏ tất cả các đường viền ngoại trừ đường viền duy nhất (Landy và Cohen, 1985). Bảng quyết định cho quá trình mảnh trong bước đầu tiên không được cho phép mảnh thành nhỏ hơn hai pixel. Điều này đảm bảo không tạo ra khoảng trống bằng cách mảnh cả hai bên của một đường viền trong cùng một lần quét. Một phương pháp tương tự, được mô tả bởi Rosenfeld và Kak (1982), cũng yêu cầu hai bước. Phương pháp của họ mảnh lần lượt các cạnh trên và cạnh trái của đường viền, sau đó là cạnh dưới và cạnh phải, để tránh tạo ra khoảng trống.

Đoạn văn cũng đề cập đến hai quy tắc mảnh được thể hiện trong Hình 3, mỗi quy tắc đều bao gồm hai bước. Quy tắc đầu tiên là phương pháp được sử dụng trong xử lý các tập dữ liệu ví dụ và mảnh mềm hơn so với phương pháp thứ hai, gọi là phương pháp Landy. Cả hai được bao gồm để chỉ ra rằng không có một bộ quy tắc duy nhất phù hợp cho tất cả các tập dữ liệu. Nói chung, quá trình mảnh có thể được thực hiện trên tất cả các mẫu mà khoảng trống sẽ không được tạo ra bằng cách xóa pixel trung tâm (tức là, duy trì kết nối 8-hướng), và mẫu đó là đơn giản (tức là, chỉ tìm thấy một chuyển đổi đen-trắng duy nhất khi kiểm tra các hàng xóm theo thứ tự vòng tròn). Trường hợp đặc biệt là "kết nối 'r' (nhóm mẫu 16), nó sẽ trở thành "kết nối 'Y" trừ khi xem xét. Ngoài ra, pixel đơn lẻ hoặc cô lập (nhóm mẫu 1) có thể bị xóa trong quá trình mảnh, miễn là chúng được coi là sai lầm hoặc không quan trọng. Các đầu mút (nhóm mẫu 2 đến 4) có thể được bảo tồn hoặc không bảo tồn vì một số đầu mút có thể là đường viền hợp lệ và một số có thể là nhánh nhỏ được tạo ra như hiện tượng của việc mảnh các đường viền dày (nhiều hơn bốn pixel). Nhánh nhỏ cũng có thể bị xóa sau khi chuyển đổi thành vectơ, khi độ dài có thể được tính toán và sử dụng làm tham số chỉnh sửa. Hình 4d và hình 5d cho thấy các đường dẫn chảy sau khi đã thực hiện quá trình điền và mảnh hai bước. Trong cả hai ví dụ, quá trình mảnh đã giảm bớt hình ảnh nhị phân ban đầu thành các đường viền liên kết rời rạc và loại bỏ nhiều pixel không cần thiết.

Node marking

Sau khi các đường đã được điền và rút gọn, việc đánh dấu các điểm nút có thể được thực hiện. Hai bảng quy tắc cho việc đánh dấu nút được hiển thị trong Hình 3. Các nút là các điểm kết thúc hoặc kết thúc đường (được hiển thị dưới dạng chữ "T" trong Hình 3) có thể được phát hiện dựa trên số lượng pixel I-count, hoặc pixel có một hàng xóm chiếm dụng. Các nút kết nối (được hiển thị dưới dạng chữ "C" trong Hình 3), hoặc các nút mà các đường nối lại, có thể được phát hiện dựa trên các mẫu có ba hoặc nhiều sự chuyển đổi, chỉ ra ba đường đi từ nút đó. Một trường hợp đặc biệt là "+" (nhóm mẫu 31), đó là một điểm kết nối, ngay cả khi giá trị khoảng cách là một (tức là không có khoảng trống sẽ được tạo ra nếu pixel trung tâm bị loại bỏ). Các nút kết nối nhiều lần (được hiển thị dưới dạng chữ "M" trong Hình 3), hoặc nhóm nút, làm cho việc xác định chính xác giao điểm của các đường trở nên khó khăn. Trừ khi giảm xuống một điểm duy nhất, các nút kết nối nhiều lần cũng có thể tạo ra một mạng lưới các cung ngắn được kết nối với nhau trong quá trình tạo vectơ. Rosenfeld và Kak (1982) đề xuất một giải pháp là thực hiện một lượt đi qua hình ảnh thứ hai, kiểm tra các điểm kết nối liền kề và tính toán một điểm duy nhất (trọng tâm) có thể được sử dụng như điểm kết nối.

Creation of

Sau khi đánh dấu các nút, quá trình chuyển đổi linework thành cung vectơ khá đơn giản. Chúng ta bắt đầu từ một điểm cuối hoặc nút kết nối và chỉ cần đi theo các đường mà được đảm bảo chỉ có một con đường để đi cho đến khi tìm thấy một nút khác. Để tránh việc đi theo một đường hai lần, các pixel được đánh dấu khi tọa độ của chúng được thêm vào đường. Phương pháp này hoạt động tốt khi toàn bộ hình ảnh có thể nằm trong bộ nhớ chính của máy tính hoặc trong bộ nhớ hiển thị hình ảnh, như trong trường hợp hệ thống được sử dụng cho nghiên cứu này.

Định dạng cho các đường, chuỗi hoặc cung kết quả có thể được biểu diễn bằng tọa độ tương đối (ví dụ: dưới, trên và trái), hoặc có thể được chuyển đổi thành đo đạc tuyệt đối (ví dụ: tọa độ Descartes). Bảng Đồ Digital Line Graph (OLG) của USGS lưu trữ tọa độ tuyệt đối, thường được biểu thị bằng đơn vị bản đồ được tính bằng inch hoặc đơn vị địa chất trong mét. Đối với dự án này, định dạng OLG đã được chọn vì có nhiều hệ thống sẵn có đọc và ghi dữ liệu dưới dạng này.

Vector operation

Các linework đã được chuyển đổi thành vectơ bao gồm các đoạn đường ngắn có độ dài là một đơn vị pixel cho các hàng xóm kề nhau theo cả bốn hướng hoặc 1,414 đơn vị cho các hàng xóm chéo. Đường thẳng có thể được rút gọn bằng cách loại bỏ các điểm trung gian trên đường. Ngoài ra, đường có thể được rút gọn hơn bằng cách sử dụng một phương pháp tổng quát hóa như phương pháp được mô tả bởi Douglas và Peucker (1973). Đối với các ví dụ được sử dụng trong nghiên cứu này, việc rút gọn điểm từ 50 đến 80 phần trăm là có thể với sự khác biệt đáng kể trong chất lượng của các đường.

Line smoothing

Việc rút gọn điểm không làm giảm sự xuất hiện của các cung góc sắc nhọn mà thể hiện những phần còn lại từ nguồn gốc raster của chúng. Thực tế, việc rút gọn điểm, khi được sử dụng độc quyền, sẽ loại bỏ các góc tù và chỉ để lại các góc nhọn trên các đường. Các góc sắc nhọn tồn tại tại các điểm giao nhau có thể được xử lý bằng cách vẽ một đường cong thông qua tập hợp các điểm. Điều này sẽ giới thiệu các điểm mới nhưng cũng tạo ra các cung có diện mạo chấp nhận được hơn. Một trong những kỹ thuật như vậy là nội suy spline bậc ba (được mô tả bởi Monmonier (1982)). Trong Hình 4f, nội suy spline đã được thực hiện trên các đường dẫn với kết quả cho thấy diện mạo mượt mà hơn. Trong Hình 5f, nội suy spline đã được thực hiện và thêm vào đó, các cung ngắn đã được xóa bỏ. Trong trường hợp này, các cung ngắn được loại bỏ chủ yếu là các nhánh nhỏ sai lầm (phát sinh trong quá trình mảnh) và cầu nối giữa các đường.

Sumary

Việc xem xét kết quả của các kỹ thuật xử lý này trên các tập dữ liệu mẫu cho thấy vẫn còn nhiều lỗi trong linework. Điều này chỉ ra sự cần thiết của một cơ sở chỉnh sửa tương tác tốt. Đáng chú ý nhất là sự gián đoạn trong các đường và kết nối sai lầm giữa các đường đồng mức. Cả hai tập dữ liệu mẫu đều yêu cầu chỉnh sửa tương tác đáng kể sau khi chuyển đổi thành dạng vectơ. Điều này đúng cho hầu hết các đường loại mà chúng ta tạo ra từ OEMs hoặc các đường quét từ bản đồ chưa được chuẩn bị đặc biệt.

Nghiên cứu này được tiến hành để chuyển đổi các đường viền được định dạng thành các cung vectơ trong một định dạng phù hợp cho xử lý hệ thống thông tin địa lý. Điều này đã được thực hiện bằng cách tập hợp một số kỹ thuật xử lý hình ảnh và chỉnh sửa vectơ thành một hệ thống nguyên mẫu linh hoạt nhưng chưa hoàn chỉnh. Việc tổ chức và tài liệu hoá các điều kiện xung quanh có thể có ứng dụng rộng hơn so với yêu cầu của dự án này, và được cho là việc sử dụng các nhóm mẫu có thể cung cấp một khung việc hiệu quả và nhất quán cho nhiều hoạt động hàng xóm được thực hiện trong lĩnh vực raster. Những kinh nghiệm thu được trong dự án này cũng đã giúp xác định một cách khách quan các bước cần thiết trong quá trình chuyển đổi từ raster sang vectơ. Việc hiểu biết về các quy trình này sẽ trở nên càng quý giá hơn khi các hệ thống quét trở nên phổ biến hơn và các hệ thống hoàn chỉnh được tùy chỉnh tốt hơn để đáp ứng các yêu cầu cụ thể.