A group of letters in squares

Description automatically generatedA screenshot of a grid

Description automatically generated

HÌNH 10.18

(a) Pixel p và 4 lân cận của nó,

{b) Pixel p và các

lân cận theo đường

chéo của nó,

(c) Pixel p và 8 lân

cận của nó,

(d) Các pixel p và q là 4 lân cận và 8 lân cận.

(e) Các pixel p và q là 8 liền kề nhưng không liền kề 4.

{f) Các pixel được tô

bóng đều có 4 kết nối và 8 kết nối.

(g) Các pixel được tô bóng là các pixel

được kết nối 8 nhưng không

Một thành phần được kết nối chỉ được xác định theo đường dẫn và định nghĩa của đường dẫn lần lượt phụ thuộc vào loại kề được sử dụng. Điều này ngụ ý rằng bản chất của một thành phần liên thông phụ thuộc vào dạng kề cận nào mà chúng ta chọn, trong đó kề cận 4 và kề 8 là phổ biến nhất. Hình 10.19 minh họa ảnh hưởng của tính kề cận trong việc xác định số lượng thành phần được kết nối trong một ảnh. Hình 10. l 9(a) hiển thị một ảnh nhị phân nhỏ có bốn thành phần được kết nối 4. Hình 10. l 9(b) cho thấy rằng việc chọn 8 lân cận sẽ giảm số lượng thành phần được kết nối xuống còn hai.Chức năng hộp công cụ bwlabel tính toán tất cả các thành phần được kết nối trong một

Hai pixel tiền cảnh p và q được gọi là liên kết 4 nếu tồn tại một đường dẫn 4 kết nối giữa chúng, bao gồm toàn bộ các pixel tiền cảnh [Hình. I0. 1 8(f)]. Chúng được kết nối 8 nếu tồn tại một đường dẫn 8 kết nối giữa chúng [Hình. I0.18(g)]. Đối với bất kỳ pixel nền trước nào, p, tập hợp tất cả các pixel nền trước được kết nối với nó được gọi là thành phần được kết nối chứa p.

Hai pixel p và q được gọi là 4 lân cận nếu q € N4 (p ). Tương tự, p và q được gọi là kề nhau 8 nếu q € N8(p). Hình 10.18(d) và (e) minh họa những khái niệm này. Đường dẫn giữa các điểm ảnh p1 và p,, là một chuỗi các điểm ảnh p1 , p2, • thíor , p11\_ 1 , p,, sao cho pk liền kề với pk+ I ' for 1 ::::; k < n. Một đường dẫn có thể có 4 nối hoặc 8 nối, tùy thuộc vào loại kề được sử dụng.hình ảnh nhị phân. Cú pháp gọi là

[L, num ] = bwlabel (f, conn )

trong đó f là hình ảnh nhị phân đầu vào và conn chỉ định kết nối mong muốn (4 hoặc 8).

Đầu ra L được gọi là ma trận nhãn và num (tùy chọn) cho biết

A group of numbers on a grid

Description automatically generated

HÌNH 10.19

Các thành phần được kết nối.

(a) Bốn thành phần 4 nối.

(b) Hai 8 nối các thành phần.

(c) Ma trận nhãn thu được bằng cách sử dụng kết nối 4.

(d) Ma trận

nhãn thu được bằng cách sử dụng kết nối 8.

tổng số thành phần được kết nối được tìm thấy. Nếu tham số conn bị bỏ qua, giá trị mặc định của nó là 8. Hình 1 0.19(c) hiển thị ma trận nhãn cho hình ảnh trong Hình 10.19(a), được tính bằng bwlabel ( f, 4). Các pixel trong mỗi thành phần được kết nối khác nhau được gán một số nguyên duy nhất, từ 1 đến tổng số thành phần được kết nối được tìm thấy. Nói cách khác, tập hợp các pixel có nhãn 1 thuộc về thành phần được kết nối đầu tiên; tập hợp các pixel có nhãn 2 thuộc về thành phần kết nối thứ hai; và như thế. Các pixel nền được gắn nhãn 0. Hình 10.19(d) hiển thị ma trận nhãn tương ứng với Hình 10. 19(a), được tính bằng bwlabel (f, 8).

Ví dụ này cho thấy cách tính toán và hiển thị khối tâm của từng thành phần được kết

nối trong Hình 10.17(a). Đầu tiên, chúng ta sử dụng bwlabel để tính các thành phần 8

liên thông:

>> f = imread ( ' obj ects . t if ' );

>> [L, n] = bwlabel( f ) ;

Hàm find (Phần 5.2.2) rất hữu ích khi làm việc với ma trận nhãn.

Ví dụ: lệnh gọi hàm f ind sau đây trả về chỉ số hàng và cột cho tất cả các pixel thuộc đối tượng thứ ba:

>> [ r , c] = find (L == 3 ) ;

Hàm trung bình với r và c làm đầu vào, sau đó tính khối tâm của vật thể này.

>> rbar = mean ( r ) ;

>> cbar = mean(c) ;

Một vòng lặp có thể được sử dụng để tính toán và hiển thị tâm khối lượng của tất cả các vật thể trong ảnh. Để làm cho khối tâm hiển thị khi được xếp chồng lên hình ảnh, chúng tôi hiển thị chúng bằng cách sử dụng điểm đánh dấu " \* " màu trắng phía trên điểm đánh dấu hình tròn tô màu đen, như sau:

>> imshow(f)

>> hold on % So later plotting commands plot on top of the image .

>> for k= 1 : n

[ r, c] = f ind ( L k) ;

rbar = mean(r ) ;

cbar = mean(c) ;

plot ( cbar , rbar, ' Marker ' , 'o' , ' Marke rEdgeColor' , 'k' , ...

' Marke rFaceColor ', 'k' , ' Marke rSize ', 10) ;

plot ( cbar, rbar, ' Marker ' , '\*' , ' Marke rEdgeColor ' , 'w' ) ;

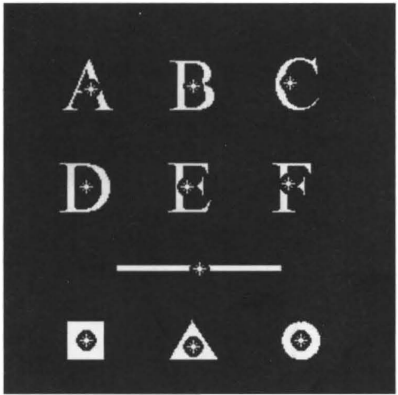
end

Hình 10.20 thể hiện kết quả.

HÌNH I 0.20

Khối tâm (dấu hoa

thị màu trắng) được

hiển thị chồng lên các thành

phần được kết nối tương

ứng của chúng.

**10.5 Tái thiết hình thái**

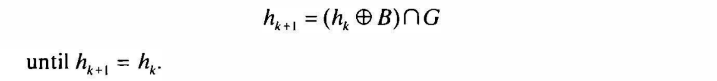
Tái thiết là một sự biến đổi hình thái bao gồm hai hình ảnh và một phần tử cấu trúc (thay vì một hình ảnh và phần tử cấu trúc). Một hình ảnh, điểm đánh dấu, là điểm bắt đầu cho quá trình chuyển đổi. Hình ảnh còn lại, mặt nạ, hạn chế sự chuyển đổi. Phần tử cấu trúc được sử dụng xác định kết nối. Trong phần này chúng tôi sử dụng kết nối 8 (mặc định), ngụ ý rằng B trong phần thảo luận sau đây là ma trận 3 X 3 của ls, với tâm được xác định tại tọa độ (2, 2). Trong phần này chúng ta xử lý ảnh nhị phân; tái thiết thang độ xám được thảo luận trong Phần 10.6.3.

Nếu G là mặt nạ và F là điểm đánh dấu, việc tái tạo G từ F, ký hiệu là RG (F), được xác định bằng quy trình lặp sau:

1. Khởi tạo h1 làm ảnh đánh dấu, F

2. Tạo phần tử cấu trúc: B = ones ( 3).

3. Lặp lại:



4. *RG(F) = hk+1*

Điểm đánh dấu F phải là tập con của G:



Hình 10.21 minh họa thủ tục lặp lại trước đó. Mặc dù công thức lặp này hữu ích về mặt khái niệm nhưng vẫn tồn tại các thuật toán tính toán nhanh hơn nhiều. Chức năng hộp công cụ imreconst ruct sử dụng thuật toán "tái thiết lai nhanh" được mô tả trong Vincent [1993). Cú pháp gọi imreconstruct là:

out = imrecon st ruct ( marker, mask )

trong đó điểm đánh dấu và mas k được xác định ở phần đầu của phần này.

**10.5.1 Mở bằng tái tạo**

Trong việc mở hình thái, sự xói mòn thường loại bỏ các vật thể nhỏ và sự giãn nở tiếp theo có xu hướng khôi phục lại hình dạng của các vật thể còn lại. Tuy nhiên, độ chính xác của việc phục hồi này phụ thuộc vào sự giống nhau giữa hình dạng và thành phần cấu trúc. Phương pháp được thảo luận trong phần này, mở bằng cách tái tạo, khôi phục hình dạng ban đầu của các đối tượng còn lại sau khi bị xói mòn. Việc mở bằng cách tái tạo ảnh G sử dụng phần tử cấu trúc B, được định nghĩa là Rc(G 8 B).

• So sánh giữa mở và mở bằng cách xây dựng lại một hình ảnh chứa văn bản được thể hiện trong Hình 10.22. Trong ví dụ này, chúng ta quan tâm đến việc trích xuất từ Hình 10.22(a) các ký tự có chứa các nét dọc dài.

A collage of different shapes

Description automatically generated



HÌNH 10.21 Tái thiết hình thái. (a) Ảnh gốc (mặt nạ). (b) Hình ảnh đánh dấu. ( c)-( e) Kết quả trung gian sau 100, 200 và 300 lần lặp tương ứng. (f) Kết quả cuối cùng. (Đường viền của các đối tượng trong ảnh mặt nạ được xếp chồng lên (b)-(e) làm tham chiếu trực quan.)

Bởi vì cả mở và mở bằng cách xây dựng lại đều có điểm chung là xói mòn, nên chúng tôi thực hiện bước đó trước tiên, sử dụng phần tử cấu trúc dọc, mỏng có chiều dài tỷ lệ với chiều cao của ký tự:

>> f = imread ( ' book\_t ext\_bw . tif ' ) ;

>> fe = imerode (f, ones (51 , 1));

Hình 10.22(b) thể hiện kết quả. Lỗ mở, được minh họa trong Hình 10.22( c ), là được tính bằng imopen:

>> fo = imopen (f,ones (51 , 1));

Lưu ý rằng các nét dọc đã được khôi phục chứ không phải các ký tự còn lại chứa các nét đó. Cuối cùng, chúng tôi có được sự tái thiết:

>> fobr = imreconst ruct (fe , f ) ;

HÌNH 10.22

A close-up of a black and white screen

Description automatically generatedHÌNH 10.22 Tái thiết hình thái: (a) Ảnh gốc.

(b) Ảnh bị xói mòn theo đường thẳng đứng;

(c) được mở bằng một đường thẳng đứng; và

(d) mở bằng cách xây dựng lại theo đường thẳng đứng.

( e) Các lỗ đã được lấp đầy.

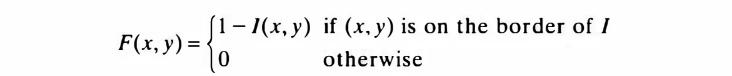
(f) Các ký tự chạm vào đường viền (xem đường viền bên phải).

(g) Đã xóa các ký tự viền.

Kết quả ở Hình 1 0.22(d) cho thấy các ký tự chứa nét dọc dài đã được khôi phục chính xác; tất cả các ký tự khác đã bị xóa. Các phần còn lại của Hình 10.22 được giải thích trong hai phần sau.

**10.5.2 Lấp đầy lỗ**

Tái tạo hình thái có nhiều ứng dụng thực tế, mỗi ứng dụng được đặc trưng bởi việc lựa chọn hình ảnh đánh dấu và mặt nạ. Ví dụ, hãy để tôi biểu thị một hình ảnh nhị phân và giả sử rằng chúng ta chọn hình ảnh đánh dấu, F, bằng 0 ở mọi nơi ngoại trừ đường viền hình ảnh, nơi nó được đặt thành 1 - /:



Sau đó,



là một ảnh nhị phân bằng I với tất cả các lỗ được lấp đầy, như minh họa trong Hình 10.22(e).

Hàm hộp công cụ imf ill thực hiện tính toán này một cách tự động khi sử dụng các lỗ đối số

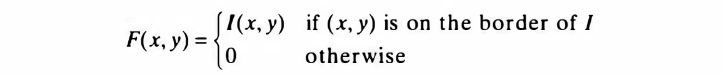
tùy chọn ' :

g = imf ill (f, ' holes ')

Chức năng này được thảo luận chi tiết hơn trong Phần 12.1.2.

10.5.3 Xóa các đối tượng viền

Một ứng dụng hữu ích khác của việc tái tạo là loại bỏ các đối tượng chạm vào viền của hình ảnh. Một lần nữa, nhiệm vụ chính là chọn điểm đánh dấu thích hợp để đạt được hiệu quả mong muốn. Giả sử chúng ta định nghĩa hình ảnh đánh dấu, F, là



nơi *I* là hình ảnh gốc. Sau đó, sử dụng *I* làm hình ảnh mặt nạ, việc tái tạo



tạo ra một ảnh, H, chỉ chứa các vật thể tiếp xúc với đường viền, như minh họa trong Hình 10.22(f). Sự khác biệt, 1 - H, được hiển thị trong Hình 10.22(g), chỉ chứa các đối tượng từ ảnh gốc không chạm vào đường viền. Chức năng hộp công cụ imclearborder tự động thực hiện toàn bộ quy trình này. Cú pháp của nó là

g = imclearborde r(f , conn )

trong đó f là hình ảnh đầu vào và g là kết quả. Giá trị của conn có thể là 4 hoặc 8 (mặc định). Chức năng này ngăn chặn các cấu trúc nhẹ hơn môi trường xung quanh và được kết nối với đường viền hình ảnh.

**10.6 Hình thái thang xám**

Tất cả các phép toán hình thái nhị phân được thảo luận trong chương này, ngoại trừ phép biến đổi hit-or-miss, đều có phần mở rộng tự nhiên cho các ảnh thang độ xám. Trong phần này, giống như trong trường hợp nhị phân, chúng ta bắt đầu với sự giãn nở và xói mòn, đối với ảnh thang độ xám được xác định theo cực tiểu và cực đại của các vùng lân cận pixel.

**10.6.1 Sự giãn nở và xói mòn**

Độ giãn thang độ xám của ảnh thang độ xám /theo phần tử cấu trúc b, ký hiệu là f B

b, được định nghĩa là



trong đó Db là tập xác định của b, và f(x, y) được giả sử bằng -∞ bên ngoài tập xác định f. Phương trình này thực hiện một quá trình tương tự như tích chập không gian, được giải thích ở Mục 3.4.1. Về mặt khái niệm, chúng ta có thể nghĩ đến việc xoay phần tử cấu trúc 1 80° quanh gốc của nó và dịch nó sang tất cả các vị trí trong thời đại hình ảnh, giống như một hạt nhân chập được xoay và sau đó dịch về hình ảnh.

Tại mỗi vị trí được dịch, các giá trị phần tử cấu trúc xoay được thêm vào các giá trị pixel của hình ảnh và giá trị tối đa được tính toán.

Một điểm khác biệt quan trọng giữa tích chập và giãn nở thang độ xám là ở chỗ, Db là một ma trận nhị phân xác định vị trí nào trong vùng lân cận được bao gồm trong phép toán tối đa. Nói cách khác, đối với một cặp tọa độ (x0, y0) tùy ý trong miền Db, thuật ngữ f(x - x0,y - y0) + b(x0,y0) chỉ được đưa vào tính toán tối đa nếu Db là 1 tại các tọa độ đó. Điều này được lặp lại cho tất cả các tọa độ (x', y') € Db mỗi khi tọa độ (x, y) thay đổi. Vẽ b(x', y') dưới dạng hàm của tọa độ x' và y' sẽ trông giống như một "bề mặt" kỹ thuật số với chiều cao ở bất kỳ cặp tọa độ nào được cho bởi giá trị của b tại các tọa độ đó.

Phép giãn nở thang độ xám thường được thực hiện bằng cách sử dụng các phần tử cấu trúc phẳng trong đó giá trị (chiều cao) của b bằng 0 tại tất cả các tọa độ mà D,, được xác định trên đó. Đó là,



Trong trường hợp này, phép toán max được xác định hoàn toàn bằng mẫu 0s và ls trong ma trận nhị phân Db và phương trình giãn nở thang màu xám được đơn giản hóa thành



Do đó, phép giãn nở thang màu xám phẳng là một toán tử cực đại cục bộ, trong đó mức tối đa được lấy trên một tập hợp các pixel lân cận được xác định bởi hình dạng không gian của các phần tử có giá trị I trong Db .

Các phần tử cấu trúc không phẳng được tạo bằng hàm st rel bằng cách truyền cho nó hai ma trận: ( 1) ma trận 0s và ls chỉ định phần tử cấu trúc do-main và (2) ma trận thứ hai chỉ định các giá trị chiều cao, Ví dụ:

>> b = strel ([1 1 1), [1 2 1))

b =

Nonflat STAEL obj ect containing 3 neighbors .

Neighborhood :

1 1 1

Height :

1 2 1

tạo phần tử cấu trúc 1 x 3 có giá trị chiều cao là b(0, -1) = 1, b(0, 0) = 2 và b(0, 1 ) = 1.

Aerial view of a city

Description automatically generatedCác phần tử cấu trúc phẳng cho ảnh thang độ xám được tạo bằng cách sử dụng strel theo cách tương tự như đối với ảnh nhị phân. Ví dụ: các lệnh sau đây hiển thị cách giãn hình ảnh f trong Hình 1 0,23(a) bằng cách sử dụng phần tử cấu trúc phẳng 3 x 3:

HÌNH 10.23

Sự giãn nở và xói mòn.

(a) Ảnh gốc.

(b) Hình ảnh giãn nở.

(c) Hình ảnh bị xói mòn.

(d) Độ dốc hình thái.

( Hình ảnh gốc được cung cấp bởi NASA.)

>> se = strel ( ' square ', 3 ) ;

>> gd = imdilate ( f, se ) ;

Hình 10.23(b) thể hiện kết quả. Đúng như dự đoán, hình ảnh hơi mờ. Phần còn lại của hình này được giải thích trong phần thảo luận sau.

Sự xói mòn thang độ xám của phần tử cấu trúc image.fby thang độ xám b, ký hiệu là f 8 b, được định nghĩa là



trong đó Dh là miền xác định của b và .f được coi là +oo nằm ngoài miền xác định của f. Như trước đây, chúng tôi nghĩ về mặt hình học về việc chuyển phần tử cấu trúc sang tất cả các vị trí trong ảnh. Tại mỗi vị trí được dịch, các giá trị phần tử cấu trúc được trừ khỏi các giá trị pixel của hình ảnh và mức tối thiểu được tính toán.

Giống như sự giãn nở, xói mòn thang xám thường được thực hiện bằng cách sử dụng cấu trúc phẳng các phần tử. Phương trình xói mòn thang xám phẳng sau đó được đơn giản hóa thành



Do đó, xói mòn thang xám phẳng là một toán tử cục bộ tối thiểu, trong đó mini-mum được tiếp quản trên một tập hợp các điểm ảnh lân cận được xác định bởi hình dạng không gian của các phần tử có giá trị 1 của Db. Hình 10.23(c) cho thấy kết quả của việc sử dụng hàm imerode với cùng phần tử cấu trúc đã được sử dụng trong Hình 10.23(b):

>> ge = imerode (f, se );

Sự giãn nở và xói mòn có thể được kết hợp để đạt được nhiều hiệu ứng khác nhau. Ví dụ, việc trừ một hình ảnh bị xói mòn khỏi phiên bản giãn nở của nó sẽ tạo ra một "độ dốc hình thái", là thước đo sự thay đổi mức xám cục bộ trong hình ảnh. Ví dụ như để

>> morph\_g rad = gd - ge ;

tạo ra hình ảnh trong Hình 10.23(d), đây là độ dốc hình thái của hình ảnh trong Hình 10.23(a). Hình ảnh này có các đặc điểm tăng cường cạnh tương tự như các đặc điểm có thể thu được bằng cách sử dụng các thao tác chuyển màu được thảo luận trong Phần 7.6.1 và sau đó trong Phần 11.1.3.

**10.6.2 Mở và đóng**

Các biểu thức mở và đóng ảnh thang độ xám có dạng giống như đối tác nhị phân của chúng. Việc mở ảnh thang độ xám f bằng phần tử cấu trúc b, ký hiệu f ◦ b, được định nghĩa là



trong đó được hiểu rằng xói mòn và giãn nở là các phép toán thang độ xám được xác định trong Phần 10.6.1. Tương tự, phần đóng của b, ký hiệu là f • b, được định nghĩa là sự giãn nở theo sau là sự xói mòn:



Cả hai phép toán đều có cách giải thích hình học đơn giản. Giả sử hàm ảnh f(x, y) được xem như một bề mặt 3-D; nghĩa là, các giá trị cường độ của nó được hiểu là các giá trị độ cao trên mặt phẳng xy. Khi đó, việc mở bởi b có thể được hiểu về mặt hình học là đẩy phần tử cấu trúc b lên phía dưới của bề mặt và dịch nó qua toàn bộ miền của f. Lỗ mở được xây dựng bằng cách tìm các điểm cao nhất mà bất kỳ bộ phận nào của phần tử kết cấu đạt tới khi nó trượt khỏi bề mặt bên dưới.

Hình 10.24 minh họa khái niệm trong một chiều. Hãy coi đường cong trong Hình 10.24(a) là các giá trị dọc theo một hàng của hình ảnh. Hình 10.24(b) cho thấy một phần tử cấu trúc phẳng ở một số vị trí, được đẩy lên so với mặt dưới của đường cong. Việc mở hoàn toàn được hiển thị dưới dạng đường cong nặng