ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA: TOÁN - TIN HỌC
CHUYÊN NGÀNH : PHƯƠNG PHÁP TOÁN TRONG TIN HỌC

LUẬN VĂN TỐT NGHIỆP

ĐỀ TÀI:

BÀI TOÁN XỬ LÝ VÀ PHÂN TÍCH ĐỂ ĐẾM CÁC ĐỐI TƯỢNG ẢNH HAI CHIỀU

Giáo Viên Hướng Dẫn : **ThS. PHẠM THẾ BẢO**
Giáo Viên Phản Biện : **TS. NGUYỄN ĐÌNH THÚC**
Sinh Viên Thực Hiện : **NGUYỄN THỊ THANH NHÀN**
LƯU HỮU THUẬN

NIÊN KHOÁ 1998 - 2002

NHẬN XÉT CỦA GIẢNG VIÊN

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

LỜI CẢM ƠN

Đầu tiên, chúng em xin chân thành cảm ơn các thầy cô đã hết lòng chỉ bảo và dạy dỗ chúng em trong suốt bốn năm học vừa qua.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Phạm Thế Bảo và thầy Hà Văn Thảo đã hướng dẫn chúng em hoàn thành đề tài này.

Chúng em xin cảm ơn khoa Toán – Tin học, Bộ môn Ứng dụng Tin học cùng các thầy cô đã tạo mọi điều kiện thuận lợi và giúp đỡ rất nhiều trong quá trình thực hiện đề tài này.

Đồng thời, chúng em cũng xin cảm ơn khoa Sinh đã hỗ trợ dữ liệu cho đề tài.

Cuối cùng xin gởi lời cảm ơn chân thành nhất đến cha mẹ, người thân đã động viên chúng em trên con đường học vấn.

LỜI MỞ ĐẦU

Trong thời đại bùng nổ về công nghệ thông tin như hiện nay, máy vi tính ngày càng được sử dụng rộng rãi trên tất cả các lĩnh vực từ nghiên cứu khoa học kỹ thuật đến các ứng dụng trong cuộc sống hàng ngày. Máy vi tính có thể là người cộng sự hỗ trợ đắc lực nhất của con người. Bạn có tin rằng máy tính có thể "nhìn" được hay không ? Xuất phát từ những nhu cầu thực tế, Thị giác máy tính đã ra đời và phát triển nhanh chóng trong sự quan tâm của mọi người. Sự xuất hiện của Thị giác máy tính đã làm tăng khả năng ứng dụng của máy tính trong nhiều lĩnh vực như: y tế, giáo dục, kinh tế, giao thông, quân sự,… Đây là một ngành rất rộng lớn, nó liên quan đến việc xử lý hình học để tạo mô hình thế giới thực từ các ảnh 2D và thao tác xử lý, phân tích ảnh nhằm phân lớp nhận biết và đếm đối tượng.

Trong Sinh học, việc nhận biết, phân lớp và đếm đối tượng đặc biệt là hồng cầu và bạch cầu nhằm góp phần vào việc sơ lược chẩn đoán bệnh là một vấn đề đã được đặt ra từ rất lâu. Trước đây, công việc này chủ yếu được thực hiện bằng mắt thường, vì vậy mất nhiều thời gian và độ chính xác không cao. Do đó, mục tiêu hướng tới của đề tài là ứng dụng Thị giác máy tính để thay thế con người thực hiện thao tác đó nhằm làm tăng độ chính xác và rút ngắn thời gian thực hiện.

Trong luận văn này, chúng em tìm hiểu những kiến thức liên quan đến thao tác xử lý phân tích ảnh trong Thị giác máy tính đồng thời ứng dụng vào việc nhận biết, phân lớp và đếm hồng cầu, bạch cầu trên ảnh bitmap.

Luận văn được chia thành 3 phần. Đầu tiên, chúng em tìm hiểu cấu trúc ảnh bitmap cùng các khái niệm thao tác cơ bản của Thị giác máy tính liên quan đến việc xử lý phân tích ảnh. Kế đến, chúng em đề cập đến các phương pháp cơ bản nhận biết, phân lớp và đếm đối tượng. Và cuối cùng, chúng em đưa ra hướng giải quyết cụ thể cho việc đếm số lượng hồng cầu, bạch cầu trên ảnh bitmap.

Phạm Thế Bảo

Digitally signed by Phạm Thế Bảo
DN: cn=Phạm Thế Bảo, c=VN, o=Khoa Tóan - Tin học trường Đại học khoa học Tp.HCM, Vietnam, ou=NCLAB, email=ptbao@mathdep.hcmuns.edu.vn
Reason: I am the author of this document
Date: 2006.08.20 09:51:08 +07'00'

MỤC LỤC

CHƯƠNG 1 : TỔNG QUAN ẢNH

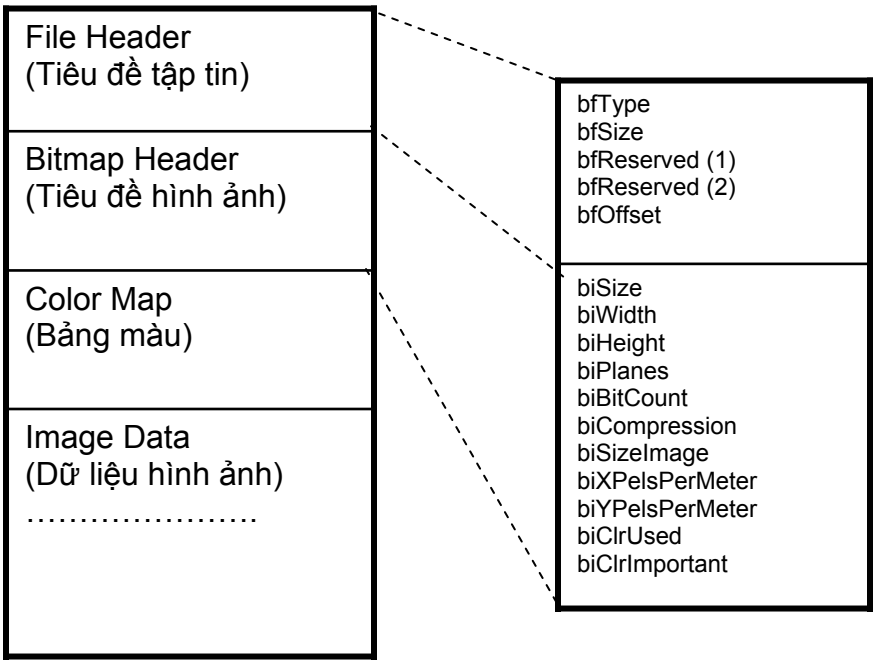
1 Ảnh Bitmap

1.1 Cấu trúc ảnh Bitmap

Tương tự với các loại ảnh khác, tập tin Bitmap (BMP) bao gồm:

- Tiêu đề (Header), phần này bao gồm:
  - Tiêu đề tập tin (File header)
  - Tiêu đề hình ảnh (Bitmap header)
- Bảng màu (Color map)
- Dữ liệu hình ảnh (Image data)

Hình sau đây minh họa cấu trúc ảnh Bitmap:



***Hình 1.1**: Cấu trúc ảnh bitmap*

1.1.1 Tiêu đề

Phần tiêu đề bao gồm tiêu đề tập tin và tiêu đề hình ảnh.

+ Tiêu đề tập tin gồm các thông tin liên quan đến bản thân tập tin:
  - bfType: là vùng dài 2 byte, luôn chứa 2 kí tự 'BM' để thể hiện tập tin kiểu Bitmap.
  - bfSize: là vùng dài 4 byte, cho biết kích thước tổng cộng của tập tin Bitmap.
  - bfReserved1 và bfReserved2: là 2 vùng, mỗi vùng dài 2 byte, 2 vùng này chừa trống để dự phòng.
  - bfOffset: là vùng dài 4 byte, chỉ ra vị trí bắt đầu của vùng dữ liệu.

+ Tiêu đề hình ảnh gồm các chi tiết liên quan đến hình ảnh chứa trong tập tin:
  - biSize: là vùng dài 4 byte, cho biết kích thước vùng tiêu đề hình ảnh.
  - biWidth: là vùng dài 4 byte, cho biết chiều rộng của hình.
  - biHieght: là vùng dài 4 byte, cho biết chiều dài của hình.
  - biPlanes: là vùng dài 2 byte, luôn chứa giá trị 1.
  - biBitCount: là vùng dài 2 byte, cho biết số bit để diễn đạt trị số pixel, các giá trị có thể là 1, 4, 8 hoặc 24.
  - biCompression: là vùng dài 4 byte, cho biết dữ liệu ảnh có được nén hay không.
  - biSizeImage: là vùng dài 4 byte, cho biết kích thước bản thân ảnh đó.
  - biXPelsPerMeter và biYPelsPerMeter: là 2 vùng dài 4 byte, cho biết độ phân giải theo chiều ngang và dọc.
  - biClrUsed: là vùng dài 4 byte, cho biết số màu trong bảng màu.
  - biClrImpotant: là vùng dài 4 byte, cho biết có bao nhiêu màu trong hình là màu quan trọng.

1.1.2 Bảng màu

Bảng màu chỉ ra các giá trị cường độ màu được sử dụng trong ảnh .
Bảng màu có hay không, dài hay ngắn là tùy thuộc vào loại ảnh Bitmap.

1.1.3 Dữ liệu hình ảnh

Pixel được lưu trữ theo dòng, từ trái sang phải trong mỗi dòng. Những dòng được lưu trữ từ dưới lên trên của bức ảnh.

1.2 Tính toán và lưu trữ trên ảnh

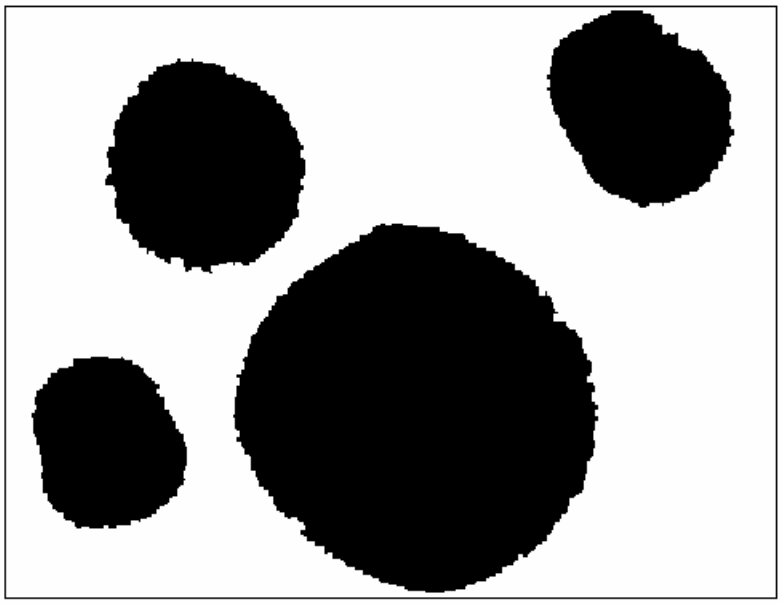
Một cảnh vật trong không gian ba chiều được thể hiện dưới dạng ảnh hai chiều trong máy tính để có thể thao tác. Ảnh trên máy tính được lưu trữ dưới dạng những con số để có thể tính toán khi thao tác với nó. Để làm được điều này, người ta dùng ảnh raster. Ảnh raster giống như một ma trận số hai chiều, mỗi phần tử trong ma trận tương ứng với một pixel trên bức ảnh.

Đề tài này, chúng em sử dụng ảnh Grey-level . Vì vậy dữ liệu hình ảnh sẽ được lưu trữ trong một ma trận hai chiều mà mỗi phần tử có giá trị trong khoảng từ 0 đến 255 ứng với giá trị màu của nó.

2 Ảnh Bi-level

2.1 Giới thiệu ảnh Bi-level

- Ảnh Bi-level là ảnh chỉ có 2 cường độ màu, thường là đen và trắng.
- Ảnh Bi-level có được từ ảnh màu bằng cách nén các cường độ màu đến khi chỉ còn hai cường độ.
- Ảnh Bi-level được sử dụng để dễ dàng phân biệt các đối tượng với nền ảnh. Sau đó, đối tượng có thể được nhận biết từ hình dạng, kích thước, phương hướng của nó.



Hình 2.1: *Ảnh Bi-level gồm các tế bào hồng cầu và bạch cầu*

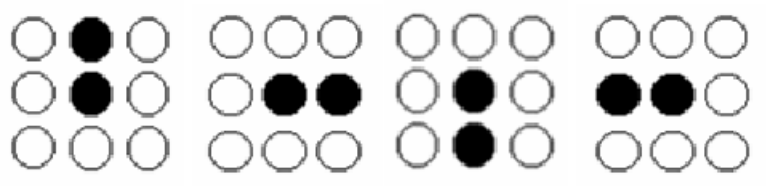
2.2 Các khái niệm cơ bản

2.2.1 Lân cận của 1 pixel

- Một pixel P ở dòng i cột j trên bức ảnh raster (Kí hiệu: P[i,j]) nằm kề các pixel khác theo sơ đồ sau:

8 (i-1,j-1)	1 (i-1,j)	2 (i-1,j+1)
7 (i,j-1)	0 (i,j)	3 (i,j+1)
6 (i+1,j-1)	5 (i+1,j)	4 (i+1,j+1)

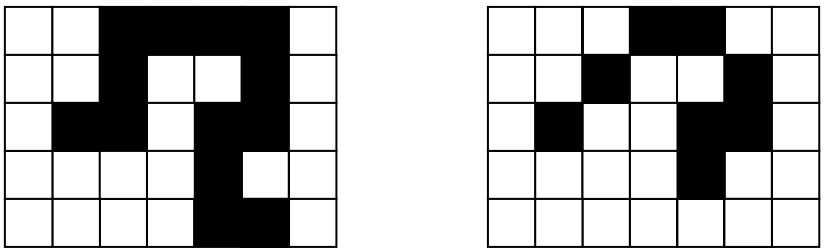
- Pixel P mang số 0, các pixel lân cận được đánh số từ 1 đến 8. Lân cận dọc của P mang số 1, 5; Lân cận ngang của P mang số 3, 7; Lân cận chéo của P mang số 2,4,6,8.
- Pixel Q là lân cận 4 của P nếu nó là lân cận dọc hoặc lân cận ngang của P.
- Pixel Q là lân cận 8 của P nếu nó là lân cận của P.



Hình 2.2.1: *Lân cận 4*

2.2.2 Đường đi

Một đường đi từ pixel $P[i_0,j_0]$ đến pixel $Q[i_n,j_n]$ là một dãy các pixel $P_0[i_0,j_0]$, $P_1[i_1,j_1]$, $P_2[i_2,j_2]$, … $P[i_n,j_n]$ thỏa điều kiện pixel P_k $[i_k,j_k]$ là lân cận của pixel P_{k+1} $[i_{k+1},j_{k+1}]$ với $\forall k : 0 \le k \le n-1$



Đường đi 4 Đường đi 8

Hình 2.2.2: *Đường đi*

2.2.3 Vùng đối tượng-Foreground

Tập hợp tất cả các pixel đen trong ảnh được gọi là vùng đối tượng hay Foreground. Kí hiệu : S.

2.2.4 Sự liên kết

Cho pixel P thuộc vùng đối tượng S, P liên kết với pixel Q thuộc S nếu có một đường đi từ P đến Q gồm những pixel thuộc S.

2.2.5 Vùng liên thông

Tập hợp các pixel trong đó mỗi pixel liên kết với tất cả các pixel còn lại được gọi là vùng liên thông.

Trong bài viết này dùng thuật ngữ đối tượng hoặc vùng để thay cho vùng liên thông.

2.2.6 Nền ảnh(Background) và lỗ trống(Hole)

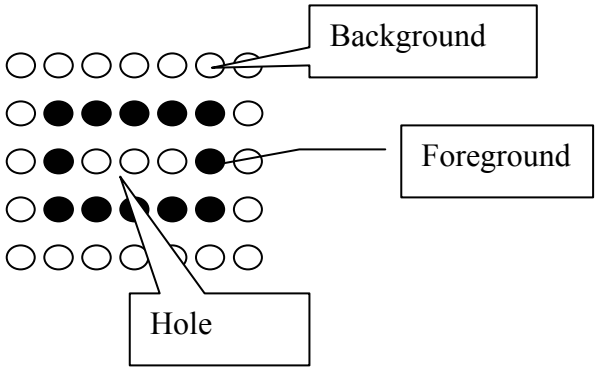
Gọi U: toàn bộ không gian bức ảnh.

Đặt $\overline{S}$ = U-S;

Tập hợp tất cả các vùng liên thông của $\overline{S}$ mà có pixel nằm trên đường viền của ảnh được gọi là nền ảnh (Background).

Các vùng còn lại của $\overline{S}$ được gọi là lỗ trống (Hole).

Nếu không cần sự phân biệt rõ ràng, thuật ngữ "nền ảnh" được dùng để bao gồm cả nền ảnh và lỗ trống được định nghĩa ở trên.

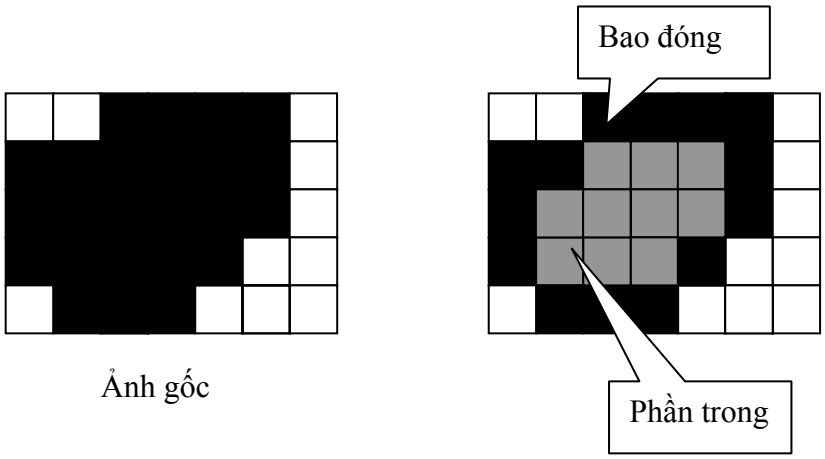


Hình 2.2.6: *Nền ảnh, vùng đối tượng và lỗ trống*

2.2.7 Bao đóng và phần trong

Bao đóng của một vùng là tập hợp các pixel thuộc vùng đó mà có lân cận thuộc nền hoặc lỗ trống (thường sử dụng lân cận 4).

Phần trong của đối tượng là tập hợp các pixel thuộc vùng đó mà không nằm trên bao đóng của nó.



Hình 2.2.7: *Bao đóng và phần trong*

2.3 Các số đo cơ bản của vùng

2.3.1 Diện tích

- Diện tích của một vùng được tính bằng tổng số pixel tạo nên vùng đó.
- Phương pháp tính diện tích của một vùng:

+ Xác định vùng cần được tính diện tích, đánh dấu tất cả các pixel thuộc vùng đó với 1 giá trị cường độ duy nhất.

+ Các pixel có giá trị cường độ đó được đếm và kết quả đếm cuối cùng chính là diện tích của vùng.

2.3.2 Chu vi

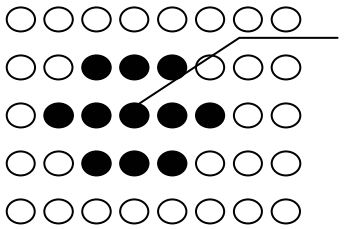
- Chu vi của một vùng chính là số pixel tạo nên bao đóng của vùng đó.
- Xác định trọng của một một pixel P:
  + Nếu hai lân cận của P đều là lân cận 4 thì P có trọng là 1.
  + Nếu hai lân cận của P đều là lân cận chéo thì P có trọng là 1.414.
  + Nếu P có một lân cận 4 và một lân cận chéo thì P có trọng là 1.207.
- Phương pháp tính chu vi của một vùng:
  + Các pixel trên bao đóng được đánh dấu.
  + Các pixel này và các lân cận của nó được kiểm tra để xác định trọng của pixel.
  + Trọng của tất cả các pixel được cộng lại; kết quả cuối cùng chính là chu vi của đối tượng.

2.3.3 Chiều dài

- Chiều dài là một số đo liên quan đến đoạn thẳng có bề rộng 1 pixel.
- Phương pháp tính chiều dài của đoạn thẳng tương tự như cách tính chu vi, tuy nhiên pixel đầu và cuối được tính trọng như sau:
  + Trọng là 0.5 nếu lân cận của nó là lân cận 4.
  + Trọng là 0.707 nếu lân cận của nó là lân cận chéo.

2.3.4 Tâm của vùng

- Tâm của vùng là một điểm mà tại đó đối tượng có thể cân bằng.Tâm của vùng còn được gọi là trọng tâm của đối tượng.



Hình 2.3.4: *Tâm vùng*

- Giả sử xét ảnh F , pixel trong ảnh có cường độ bằng 1 nếu nó thuộc về đối tượng hoặc bằng 0 nếu ngược lại. Gọi $C[C_r,C_c]$ là trọng tâm của đối tượng, khi đó vị trí của C được xác định như sau:

$$C_r = \frac{\sum_{row=1}^{NR}\sum_{col=1}^{NC} F(row,col)\times row}{area(F)}$$

$$C_c = \frac{\sum_{row=1}^{NR}\sum_{col=1}^{NC} F(row,col)\times col}{area(F)}$$

2.3.5 Số đo độ tròn(Circularity Measure)

Số đo độ tròn được xác định theo công thức sau:

$$C = 4\pi \times \frac{A}{P^2}$$

với P: chu vi của đối tượng.

A: diện tích của đối tượng.

Khi đối tượng đang xét có dạng tròn thì số đo này có giá trị bằng 1, và giá trị sẽ giảm nếu đối tượng có hình dạng không đều.

2.3.6 Công thức xác định số đo dạng hình chữ nhật

Số đo dạng hình chữ nhật được tính theo công thức:

$$R = \frac{A_r}{A_{min}}$$

với A_r : diện tích đối tượng đang xét.

A_{min} : diện tích hình chữ nhật nhỏ nhất chứa đối tượng.

Nếu đối tượng là hình chữ nhật thì tỷ số này sẽ có giá trị là 1, và sẽ giảm nếu đối tượng có hình dạng phức tạp hơn.

2.4 Một số thao tác đơn giản trên ảnh Bi-level

2.4.1 Xác định bao đóng

- Đối với hầu hết các đối tượng, sử dụng bao đóng đủ để nhận biết đối tượng, đồng thời cũng rất thuận tiện do bao đóng chứa ít pixel hơn.
- Các bước xác định bao đóng của một vùng:

+ Các pixel trên bao đóng của vùng đó được đánh dấu với cùng một giá trị mới(là những pixel có ít nhất một lân cận thuộc nền ảnh hoặc lỗ trống).

+ Sau đó, tất cả các pixel của vùng không có giá trị đó được xóa (đặt cùng giá trị với cường độ nền). Phần còn lại chính là bao đóng của vùng.

0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 0 0	0 0 0 0 2 2 2 2 0 0	0 0 0 0 2 2 2 2 0 0
0 0 0 1 1 1 1 1 0 0	0 0 0 2 1 1 1 2 0 0	0 0 0 2 0 0 0 2 0 0
0 0 0 1 1 1 1 1 0 0	0 0 0 2 2 2 2 2 0 0	0 0 0 2 2 2 2 2 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
Ảnh dữ liệu	Đánh dấu pixel thuộc bao đóng	Xóa các pixel không thộc bao đóng

Hình 2.4.1: *Các bước xác định bao đóng đối tượng*

2.4.2 Xác định trục chính của đối tượng(Principal axis)

- Định nghĩa trục chính của đối tượng: là đường thẳng đi qua tâm của đối tượng và có tổng khoảng cách đến tất cả các pixel thuộc đối tượng là ngắn nhất.
- Các bước xác định trục chính của đối tượng:

+ Định vị tâm của đối tượng.

+ Đánh dấu tất cả các pixel thuộc bao đóng và nằm phía bên trên(hoặc ngang) của tâm. Trục chính sẽ đi qua tâm và một trong số các pixel đó, do đó chúng ta có một tập các đường thẳng có thể là trục chính của đối tượng đang xét.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 # 0 0 0 0 0 0	0 0 0 0 0 0 0 4 0 0 0 0 0 0
0 0 0 0 0 0 # # # 0 0 0 0 0	0 0 0 0 0 0 3 # 5 0 0 0 0 0
0 0 0 0 0 # # # # # 0 0 0 0	0 0 0 0 0 2 # # # 6 0 0 0 0
0 0 0 0 # # x # # 0 0 0 0 0	0 0 0 0 1 # x # 7 0 0 0 0 0
0 0 0 # # # # # 0 0 0 0 0 0	0 0 0 # # # # # 0 0 0 0 0 0
0 0 0 0 # # # 0 0 0 0 0 0 0	0 0 0 0 # # # 0 0 0 0 0 0 0
0 0 0 0 0 # 0 0 0 0 0 0 0 0	0 0 0 0 0 # 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0
Đối tượng có dạng hình chữ nhật	Các pixel được đánh dấu

Hình 2.4.2: *Đánh dấu pixel*

+ Chọn đường thẳng có tổng khoảng cách đến tất cả các pixel là ngắn nhất, đường thẳng đó được xem như là trục chính của đối tượng.

2.4.3 Xác định diện tích hình chữ nhật nhỏ nhất chứa đối tượng

2.4.3.1 Dựa vào phương pháp quay đối tượng

Sau khi đã xác định được trục chính của đối tượng, ta thực hiện phép quay đối tượng sao cho trục chính song song với trục của ảnh.Khi đó, diện tích hình chữ nhật nhỏ nhất chứa đối tượng có thể được xác định như sau:

$$A_{min} = (x_{max} - x_{min})(y_{max} - y_{min})$$

với: x_{min},y_{min} là hoành độ và tung độ bé nhất của các pixel thuộc đối tượng.
x_{max},y_{max} là hoành độ và tung độ lớn nhất của các pixel thuộc đối tượng.

2.4.3.2 Xây dựng hình chữ nhật nhỏ nhất bao đối tượng

- Trục chính của đối tượng đi qua hai điểm P_1, P_2 thuộc bao đóng đối tượng.
- Trục phụ của đối tượng được xây dựng là đường thẳng đi qua tâm của đối tượng và vuông góc với trục chính. Từ đó, vị trí của hai điểm P_3, P_4 là giao điểm của trục phụ với bao đóng của đối tượng có thể được xác định.
- Từ các điểm P_1, P_2, P_3, P_4 ta có thể xây dựng được hình chữ nhật nhỏ nhất chứa đối tượng, và diện tích A_{min} có thể dễ dàng được tính.

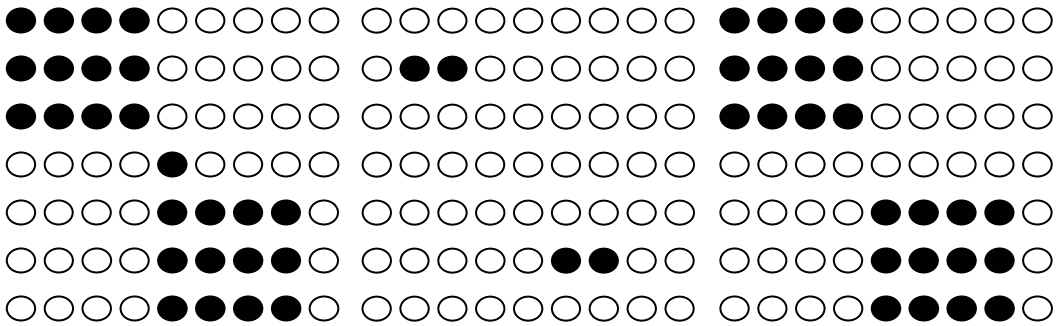
2.4.4 Mở rộng (Dilation) và thu hẹp (Erosion) vùng đối tượng

- Thu hẹp vùng đối tượng (Erosion) sẽ bỏ lớp pixel ngoài của vùng đó.
- Các bước để thu hẹp vùng đối tượng:

+ Các pixel trên bao đóng của vùng đó được đánh dấu với cùng một giá trị mới (là những pixel có ít nhất một lân cận thuộc nền ảnh hoặc lỗ trống).

+ Sau đó, tất cả các pixel của vùng có giá trị đó được xóa (đặt cùng giá trị với cường độ nền).

- Khoảng cách của pixel xa nền nhất chính là số lần thu hẹp cần thiết để xoá toàn bộ bức ảnh.
- Thu hẹp vùng đối tượng có thể được sử dụng để phân biệt các đối tượng hoặc tách các vùng chồng lên nhau một vài pixel trước khi đến số đối tượng trong ảnh.
- Thu hẹp vùng đối tượng thường được sử dụng cùng với thao tác mở rộng vùng (Dilation). Mở rộng vùng sẽ thêm một lớp mới các pixel xung quanh của vùng.

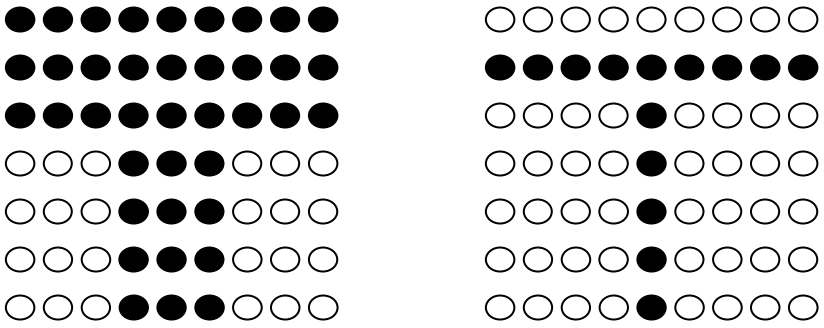


Ảnh gốc Thu hẹp vùng đối tượng Mở rộng vùng đối tượng

Hình 2.4.4: *Thu hẹp và mở rộng vùng đối tượng*

2.4.5 Lọc xương đối tượng (Skeletonization)

- Thao tác lọc xương đối tượng (Skeletonization) thường được sử dụng với những vùng bao gồm chủ yếu những đoạn thẳng. Thao tác này sẽ xóa đi những pixel phụ và tạo ra ảnh mới đơn giản hơn, có dạng tương tự với đối tượng ban đầu(gọi là xương đối tượng).



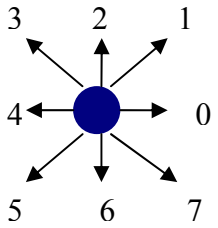
Hình 2.4.5: *Phương pháp lọc xương đối tượng*

- Xương đối tượng phải thỏa các điều kiện sau:
  + Bao gồm những vùng mỏng có bề rộng 1 pixel.
  + Nằm giữa đối tượng ban đầu.
  + Những pixel thuộc xương đối tượng phải liên kết với nhau để tạo ra số vùng tương tự như trong đối tượng ban đầu.
- Bốn nguyên tắc quyết định một pixel có bị xóa hay không (pixel phụ):
  +Nếu nó có từ 2 đến 6 pixel lân cận thuộc đối tượng.
  + Không là pixel liên kết vùng.
  + Ít nhất một lân cận ở vị trí 1, 3, 5 (bao đóng bên phải) và ít nhất một lân cận ở vị trí 3, 5, 7 (bao đóng bên dưới) là pixel nền.
  + Hoặc ít nhất một lân cận ở vị trí 7, 1, 3 (bao đóng bên trên) và ít nhất một lân cận ở vị trí 1, 5, 7 (bao đóng bên trái) là pixel nền.

2.4.6 Mã hóa theo dạng xích(Chain Code)

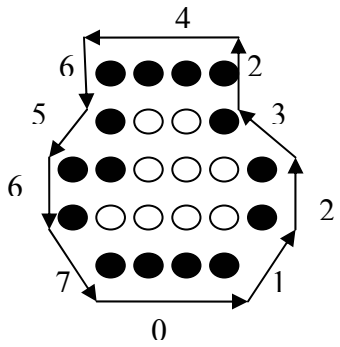
2.4.6.1 Giới thiệu Chain Code

- Chain Code dùng lưu trữ thông tin bao đóng đối tượng thay cho ảnh raster để tiết kiệm được không gian lưu trữ.
- Ý tưởng cơ bản của Chain Code là chỉ lưu trữ hướng pixel kế tiếp tương ứng mỗi pixel được liên kết trong bao đóng theo một hướng nhất định (cùng chiều hoặc ngược chiều kim đồng hồ).
- Hướng của pixel lân cận được đánh số như sau:



Hình 2.4.6.1.a: *Hướng pixel*

- Ví dụ bao đóng của một đối tượng và cách mã hoá theo dạng xích



Hình 2.4.6.1.b**:** *Bao đóng đối tượng được mã hoá theo dạng xích*

- Các bước cơ bản để tạo Chain Code:
  + Tìm pixel bắt đầu (có thể là bất kỳ pixel nào thuộc bao đóng).
  + Bước tiếp theo là định vị pixel kế tiếp, chính là pixel lân cận được chọn theo hướng của Chain Code.
  + Lặp lại bước trên cho đến khi pixel hiện hành cho đến khi pixel hiện hành là pixel bắt đầu.
- Chain Code rất linh hoạt:
  + Hướng của Chain Code dễ dàng thay đổi.
  + Có thể sử dụng Chain Code để tính một vài số đo (không phụ thuộc vào vị trí) của đối tượng mà không cần chuyển về ảnh raster như chu vi, diện tích.
- Chain Code chuẩn của một vùng là Chain Code có dãy các hướng hình thành số nguyên nhỏ nhất.

2.4.6.2 Một vài số đo được tính từ Chain Code

2.4.6.2.1 Tính chu vi

- Khoảng cách từ pixel đang xét đến lân cận theo hướng 0, 2, 4, 6 là 1 và khoảng cách đến lân cận theo hướng 1, 3, 5, 7 là 1.414.
- Nếu số phần tử chẵn trong Chain Code là N_{even}, số phần tử lẻ trong Chain Code là N_{odd} thì công thức tính chu vi là:

$$P = N_{even} + 1.414*N_{odd}$$

2.4.6.2.2 Tính diện tích

- Ý tưởng cơ bản để tính diện tích trên Chain Code dựa trên diện tích đóng góp của từng pixel thuộc bao đóng so với trục nằm ngang tùy theo hướng tương ứng của nó trên Chain Code.
- Xét pixel có tọa độ (r,c), diện tích đóng góp của nó là r nếu hướng tương ứng trong Chain Code là 0, 4; là r-0.5 nếu hướng tương ứng trong Chain Code là 1, 3, 5, 7; và không góp phần vào tổng diện tích nếu hướng tương ứng trong Chain Code là 2, 6.
- Đối với Chain Code có chiều ngược chiều kim đồng hồ, diện tích đóng góp của pixel sẽ có giá trị dương nếu hướng pixel di chuyển qua trái, và ngược lại.
- Diện tích của vùng chính là tổng diện tích đóng góp của tất cả các pixel (với vị trí pixel bắt đầu tốt nhất là (n,n)).

2.4.7 Mã hóa theo đường chạy(Run-Length Coding)

- Run-Length Coding là cách khác lưu trữ thông tin của đối tượng để có thể tiết kiệm vùng lưu trữ.
- Một đường chạy (run) bao gồm những pixel kề nhau theo một hướng được chỉ ra (thường theo chiều ngang), tất cả các pixel này có cùng giá trị cường độ.
- Một run có dạng (n)(v) với n là số pixel kề nhau có cùng giá trị cường độ v.
- Để mã hóa ảnh, bắt đầu tại pixel (0,0) và gán v với giá trị tại đó. Sau đó quét dòng đó đếm những pixel có cùng giá trị v đến khi gặp giá trị cường độ khác; nếu đến cuối dòng, tiếp tục pixel đầu tiên của dòng kế tiếp. Khi một giá trị pixel mới được tìm thấy, số đếm n và giá trị v được lưu lại, v sẽ có giá trị pixel mới này và số đếm là 1. Tiếp tục thao tác này đến khi tất cả các pixel được duyệt.

- Muốn khôi phục lại ảnh từ dạng mã theo đường chạy, kích cỡ của ảnh phải được biết trước.
- Ví dụ:

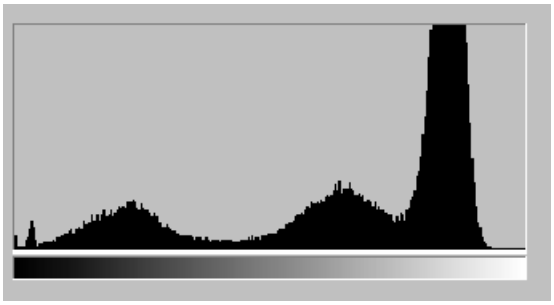
0 0 0 0 0 0 0 0 0 0 0 0 0 0	14(0) , 4(1) , 4(0) , 1(1) ,
0 0 1 1 1 1 0 0 0 0 1 0	4(1) , 3(0) , 1(1) , 4(0) ,
0 0 1 1 1 1 0 0 0 1 0 0	4(1) , 2(0) , 1(1) , 10(0) ,
0 0 1 1 1 1 0 0 1 0 0 0	1(1) , 16(0).
0 0 0 0 0 0 0 1 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0	
Ảnh dữ liệu	Mã hoá theo đường chạy

3 Ảnh Grey-Level

3.1 Biểu đồ thống kê (Grey-Level histogram)

3.1.1 Giới thiệu

- Một trong những công cụ cho việc thao tác trên ảnh Grey-level là biểu đồ thống kê mức độ xám (grey- level histogram) hay gọi tắt là biểu đồ thống kê.
- Biểu đồ thống kê là một đồ thị liệt kê tất cả các mức độ xám được sử dụng trong bức ảnh trên trục hoành và chỉ ra số pixel có mức độ tương ứng trên trục tung.
- Đối với ảnh có 8 bit trên một pixel, trục hoành có giá trị chạy từ 0 tới 255, còn trục tung tuỳ thuộc vào số pixel có trong ảnh.



Hình 3.1.1: *Biểu đồ thống kê*

3.1.2 Các dạng của biểu đồ thống kê

3.1.2.1 Biểu đồ thống kê đơn giản

- Biểu đồ thống kê đơn giản là một mảng nguyên với mỗi phần tử lưu số pixel có mức độ xám tương ứng.
- Các bước để tạo biểu đồ thống kê đơn giản:
  + Mỗi phần tử trong mảng được khởi gán là 0.
  + Sau đó mỗi pixel trong ảnh được kiểm tra, phần tử trong mảng tương ứng với mức độ xám của pixel đang xét được tăng lên.

3.1.2.2 Biểu đồ thống kê thu gọn

- Biểu đồ thống kê thu gọn cũng là một mảng nguyên với mỗi phần tử lưu số pixel nhưng có số phần tử ít hơn biểu đồ thống kê đơn giản.
- Ý tưởng cơ bản là sẽ chọn một số cố định (gọi là bề rộng thùng chứa- bin width)những mức độ xám được đếm cùng với nhau.

3.1.3 Một vài giá trị được tính từ biểu đồ thống kê

3.1.3.1 Giá trị trung bình(Mean)

- Giá trị trung bình của các mức độ xám trong một bức ảnh có thể tính bằng cách lấy tổng các giá trị trong mảng biểu đồ thống kê chia cho số pixel trong ảnh.

3.1.3.2 Giá trị Median

- Giá trị Median là mức độ xám mà có tổng số pixel có mức độ nhỏ hơn nó và tổng số pixel có mức độ lớn hơn nó xấp xỉ bằng nhau.
- Giá trị Median của một bức ảnh có thể tính bằng cách: bắt đầu tại phần tử đầu tiên trong mảng biểu đồ thống kê cộng dồn những phần tử kế tiếp cho đến khi tổng đạt tới phân nửa số pixel của ảnh.

3.1.3.3 Độ lệch chuẩn

- Độ lệch chuẩn chỉ ra độ trãi rộng của các giá trị cường độ màu.
- Độ lệch chuẩn được tính theo công thức:

$$\sigma = \sqrt{\frac{(x_i - \bar{x})^2}{N-1}}$$

với $\bar{x}$: giá trị trung bình.

3.2 Kỹ thuật Threshold

3.2.1 Khái niệm

- Threshold là thao tác chuyển ảnh Gray-level G[i,j] thành ảnh Bi-level B[i,j].
- Nếu B[i,j] có được từ G[i,j] bằng cách sử dụng ngưỡng đơn T, khi đó:

$$B[i,j] = G_T[i,j] = \begin{cases} 1 & G[i,j] \le T \\ 0 & G[i,j] > T \end{cases}$$

- Nếu mức độ xám của đối tượng thuộc miền $[T_1,T_2]$, khi đó:

$$B[i,j] = G_T[i,j] = \begin{cases} 1 & T_1 \le G[i,j] \le T_2 \\ 0 & G[i,j] < T_1, G[i,j] > T_2 \end{cases}$$

- Trong trường hợp tổng quát, ta có:

$$B[i,j] = G_T[i,j] = \begin{cases} 1 & G[i,j] \in Z \\ 0 & G[i,j] \notin Z \end{cases}$$

3.2.2 Tìm ngưỡng đơn

3.2.2.1 Dựa vào giá trị trung bình (Mean) hoặc giá trị Median

- Ngưỡng đơn T đơn giản nhất là giá trị trung bình (Mean) hoặc giá trị Median. Khi đó giả sử rằng phân nửa số pixel của toàn bộ bức ảnh thuộc về đối tượng, số pixel còn lại thuộc về nền. Tuy nhiên, điều này ít khi đúng.

3.2.2.2 Dựa vào dạng của biểu đồ

- Nếu biểu đồ thống kê của bức ảnh có 2 đỉnh lớn nhất xuất hiện rõ ràng, khi đó ngưỡng đơn được chọn là mức độ xám đại diện cho điểm thấp thất giữa hai đỉnh.

-Tuy nhiên, dạng biểu đồ thống kê có 2 đỉnh lớn rõ ràng ít khi xuất hiện. Đa số chỉ tập trung vào một đỉnh, đồng thời dạng biểu đồ thường không trơn có nhiều răng cưa nên khó xác định được đỉnh thứ hai. Có hai cách giải quyết trong trường hợp này: tính lại biểu đồ thống kê sử dụng bề rộng thùng chứa lớn hơn hoặc thay thế mỗi phần tử trong lược đồ bởi trung bình các lân cận của nó.

- Một cách đơn giản để tìm đỉnh của đồ thị là tìm một dãy pixel liên tục theo mẫu giống như đỉnh.Ví dụ, mẫu đỉnh 5 pixel thì pixel thứ 3 sẽ có mức độ xám lớn nhất, mức độ xám của pixel thứ nhất nhỏ hơn mức độ xám của pixel thứ hai và mức độ xám của pixel thứ ba lớn hơn mức độ xám của pixel thứ hai.

3.2.2.3 Phương pháp chọn lặp nhiều lần (Iterative Selection)

- Ý tưởng: tìm mức độ xám trung bình của của đối tượng(T_o) và mức độ xám trung bình của nền. Ngưỡng đơn T được tính như sau:

$$T = \frac{T_o + T_b}{2}$$

- Các bước thực hiện:

+ Khởi gán T_o,T_b. Tính T theo T_o,T_b.

+ Sau đó xác định lại T_o,T_b sử dụng ngưỡng đơn T. T_o được tính là giá trị trung bình của tất cả những pixel có mức độ xám nhỏ hơn T, T_b được tính là giá trị trung bình của tất cả những pixel có mức độ xám lớn hơn T.

+ Bước 2 được lặp lại cho đến khi giá trị T được sinh ra trong 2 lần lặp liên tiếp không đổi.

3.2.2.4 Phương pháp sử dụng số đo độ thích hợp (Correlation)

- Số đo độ thích hợp của ảnh Bi-level so với ảnh Grey-level được xác định như sau:

$$r = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i - \bar{x})^2 \sum_{i=0}^{N-1}(y_i - \bar{y})^2}}$$

với N: số pixel trong ảnh
x: ảnh xám gốc
y: ảnh đã được Threshold

- Khi đó, giá trị r sẽ thuộc đoạn [-1,1].
- Tất cả những ngưỡng có thể được sử dụng để tính độ thích hợp, ngưỡng T nào cho độ thích hợp cao nhất sẽ được chọn.

3.2.3 Chọn nhiều ngưỡng (threshold) cho ảnh

Trong trường hợp tổng quát, không thể chọn một ngưỡng đơn T cho toàn bộ bức ảnh. Bức ảnh sẽ được chia thành nhiều ảnh con và tìm ngưỡng đơn cho từng ảnh con đó.

3.2.3.1 Chia ảnh thành các vùng chữ nhật

3.2.3.1.1 Chọn giá trị Threshold cho từng vùng

- Bức ảnh sẽ được chia thành các vùng hình chữ nhật.
- Sau đó tìm ngưỡng đơn cho từng vùng, và Threshold từng vùng theo các ngưỡng tương ứng vừa tìm được.
- Tuy nhiên, trong tất cả các trường hợp, phương pháp trên sẽ tạo ra bao đóng giữa các vùng kề nhau do sử dụng hai giá trị Threshold khá khác nhau cho những pixel kề nhau.

3.2.3.1.1 Chọn giá trị Threshold cho từng pixel

- Bức ảnh sẽ được chia thành các vùng hình chữ nhật.
- Sau đó tìm ngưỡng đơn cho từng vùng.
- Chọn giá trị ngưỡng của vùng là giá trị ngưỡng cho pixel trung tâm của vùng tương ứng. Mỗi pixel còn lại có giá trị ngưỡng khác nhau tương ứng tùy thuộc vào vị trí của nó trong ảnh, được xác định như một hàm của khoảng cách dựa vào những pixel mà giá trị Threshold đã được biết.
- Gọi T[i,j] là ảnh chứa các giá trị Threshold của từng pixel tương ứng. Khi đó, ảnh dữ liệu có thể được Threshold như sau:

$$B[i,j] = \begin{cases} 0 & , G[i,j] \le T[i,j] \\ 1 & , G[i,j] > T[i,j] \end{cases}$$

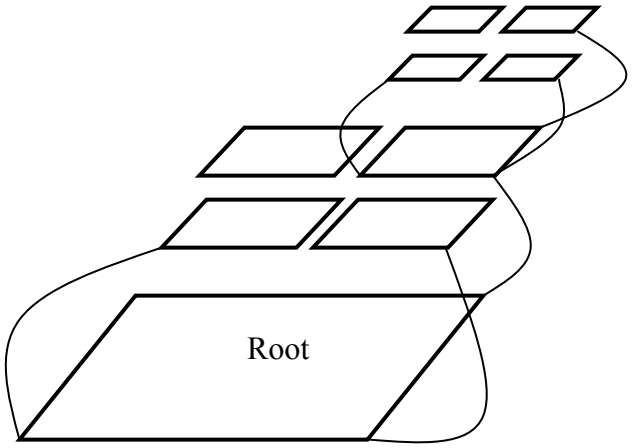
- Phương pháp này khá nhạy cảm với kích cỡ của ảnh con.

3.2.3.2 Phương pháp khoanh vùng các mức độ xám (Region Growing Method)

- Những pixel lân cận trong ảnh có cùng thuộc tính (có độ sai lệch mức độ xám ΔT cho trước) sẽ được nhóm thành một vùng.
- Tăng giá trị ΔT cho đến khi chỉ còn hai vùng được sinh ra.
- Chọn ngưỡng đơn cho từng vùng T_1,T_2 (giả sử $T_1<T_2$).
- Những pixel có mức độ xám nhỏ hơn T_1 sẽ có giá trị cường độ là 0 (pixel đối tượng), những pixel có mức độ xám lớn hơn T_2 sẽ có giá trị cường độ là 255 (pixel nền). Những pixel có mức độ xám T: $T_1 \le T \le T_2$ có giá trị cường độ được gán lại như sau: bất kỳ pixel nào có mức độ xám T_1 sẽ được gán giá trị 0 nếu nó có lân cận có giá trị 0, sau đó tương tự cho những pixel có mức độ xám lớn hơn T_1 (sẽ được gán giá trị 0 nếu nó có lân cận có giá trị 0); cuối cùng những pixel chưa xét sẽ được gán giá trị 255.

3.2.3.3 Phương pháp chia ảnh-gộp vùng (Split and Merge Method)

- Bắt đầu chia bức ảnh thành 4 vùng có diện tích bằng nhau, bất kỳ vùng nào không đồng nhất sẽ được chia tiếp tục với cách tương tự.
- Việc phân chia có thể thực hiện đệ qui, và một cây tứ phân (quad tree)với mỗi nút đại diện cho một vùng được chia có thể được thiết lập.



Hình 3.2.3.3: *Cấu trúc cây tứ phân của ảnh*

- Một vài tính chất của cây tứ phân:
  + Mỗi nút có 4 nút con.
  + Nút gốc đại diện cho toàn bộ bức ảnh, mỗi nút con của nút gốc đại diện cho ¼ bức ảnh, …
  + Do một vùng chỉ được chia nếu nó không đồng nhất nên cây sẽ không cân bằng.
  + Nút lá đại diện cho vùng đã đồng nhất, có thể chỉ là pixel đơn.
- Sau khi việc phân chia hoàn thành, giá trị trung bình của những vùng kề nhau sẽ được so sánh để quyết định chúng có được gộp lại với nhau không.

3.3 Điều chỉnh ảnh Grey-level

Dãy các mức độ xám trong một ảnh Grey-level có thể có thể được nén lại, mở rộng, dời đi hoặc gán lại,… nhằm làm thay đổi độ tương phản của ảnh. Các thao tác đó gọi là điều chỉnh ảnh.

Ví dụ: Threshold là một dạng của việc điều chỉnh mức độ xám, thao tác này sẽ giảm số mức độ xám xuống còn 2.

3.3.1 Phép biến đổi mức độ xám tuyến tính(Linear Grey-level Transformation)

- Một trong những thao tác tăng độ tương phản hữu hiệu nhất là sử dụng phép biến đổi mức độ xám tuyến tính.

- Phép biến đổi mức độ xám tuyến tính là một ánh xạ từ một tập các mức độ xám thành một tập khác tương ứng dựa trên một hàm tuyến tính.

- Xét một ảnh Grey-level có mức độ xám nhỏ nhất là $G_{\min}$ và lớn nhất là $G_{\max}$, giả sử cần cần điều chỉnh để ảnh có mức độ xám nhỏ nhất là $R_{\min}$ và lớn nhất là $R_{\max}$, khi đó phép biến đổi tổng quát là:

$$y = \frac{R_{\max} - R_{\min}}{G_{\max} - G_{\min}}(x - G_{\min}) + R_{\min}$$

với x: giá trị cường độ của pixel P đang xét

y: giá trị cường độ của pixel P sau khi biến đổi.

$m = \frac{R_{\max} - R_{\min}}{G_{\max} - G_{\min}}$: hệ số góc của phép biến đổi.

+ Nếu m<1 : dãy mức độ xám bị nén lại.

+ Nếu m=1 : dãy mức độ xám không đổi.

+ Nếu m>1 : dãy mức độ xám được mở rộng.

Tuy nhiên khi dãy mức độ xám được mở rộng, cần phải đảm bảo không được vượt quá giá trị có thể (Chẳng hạn, đối với ảnh Grey-level mức độ xám nhỏ nhất có thể là 0, và lớn nhất có thể là 255). Mặt khác, khi dãy mức độ xám bị nén lại, có thể không mở rộng lại được dãy ban đầu.

3.3.2 Phép biến đổi tuyến tính phân đoạn(Piecewise Linear Transformation)

- Phép biến đổi tuyến tính phân đoạn bao gồm nhiều hàm tuyến tính, mỗi hàm được áp dụng cho một dãy mức độ xám khác nhau.

- Phép biến đổi tuyến tính phân đoạn có dạng tổng quát như sau:

$$y = \begin{cases} a_1 x + b_1 & c_0 \le x \le c_1 \\ \quad ..... & \\ a_n x + b_b & c_{n-1} \le x \le c_n \end{cases}$$

Cần chú ý nên giữ liên tục nơi các đoạn thẳng trong phép biến đổi tuyến tính phân đoạn gặp nhau, và các đoạn thẳng này không nằm chồng lên nhau.

3.3.3 Thao tác cân bằng biểu đồ thống kê

- Trong một vài trường hợp, ảnh Grey-level chứa đầy những mức độ màu có thể sử dụng, nhưng hầu hết tất cả các pixel chỉ sử dụng một vài giá trị cường độ. Trong những trường hợp đó, thao tác cân bằng biểu đồ thống kê nên được thực hiện.

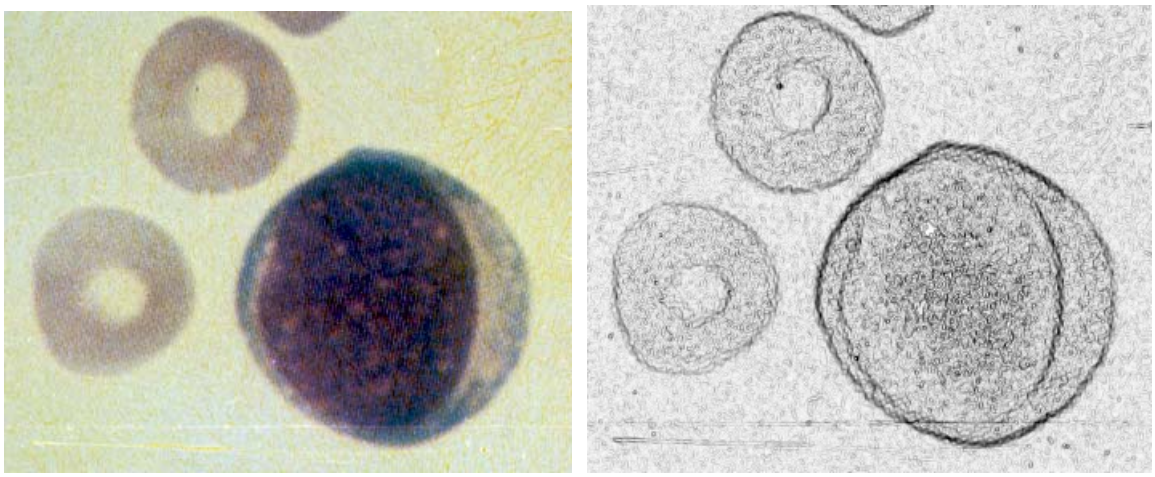
- Nếu có N mức độ xám có thể, mỗi thùng chứa trong biểu đồ thống kê sẽ có bề rộng là 1/N số pixel trong ảnh(Ký hiệu: b). Ví dụ: ảnh gồm 16 dòng, 16 cột và 8 mức độ có thể (0-7), thì mỗi thùng chứa nên có khoảng b= (16*16)/8 = 32 pixel trong nó.

- Thực hiện thao tác cân bằng biểu đồ thống kê theo luật sau: tổng số pixel có giá trị cường độ nhỏ hơn hoặc bằng k sẽ không nhỏ hơn giá trị chuẩn : k.b .Tổng số pixel có giá trị nhỏ hơn hoặc bằng k được gọi là tổng tích lũy (Cumulative Sum) tại giá trị k.

3.4 Cạnh và đường thẳng

3.4.1 Cạnh

Một cách để có thể nhận dạng đối tượng trên ảnh là chúng ta sử dụng những mức độ xám khác nhau giữa các đối tượng trong ảnh và giữa những đối tượng với nền của ảnh. Nếu các đối tượng của những lớp khác nhau không có cùng mức độ xám hay các đối tượng không bị nền của ảnh chồng lên thì ta có thể xác định được bao đóng của những đối tượng. Trong trường hợp này ta có một phương pháp gọi là dò cạnh (edge detection) để có thể làm tăng sự tương phản của những cạnh trong đối tượng để những đối tượng có thể dễ dàng phát hiện ra hơn.



(a) (b)

Hình 3.4.1 : *Minh họa phương pháp dò cạnh .*
(a) Ảnh gốc . (b) Ảnh sau khi dò cạnh.

3.4.2 Nguyên tắc cơ bản:

Nguyên tắc cơ bản phương pháp dò cạnh là những điểm trên cạnh được đánh dấu dựa trên sự thay đổi đột ngột của mức độ xám trên ảnh. Điều này nghĩa là nơi có sự thay đổi khá lớn về mức độ xám trên ảnh chính là cạnh cần dò. Chú ý rằng sự thay đổi độ sáng khi quét ngang ảnh khác với quét dọc ảnh vì thế hướng quét thì quan trọng trong phương pháp này.

3.4.2.1 Dựa vào sự thay giá trị cường độ xám theo chiều ngang hoặc dọc

3.4.2.1.1 Theo chiều ngang

Dựa trên nguyên tắc cơ bản trên, chúng ta có thể dò cạnh bằng cách xem xét sự thay đổi giá trị cường độ xám giữa những cột pixel lân cận nhau trên ảnh (Δ^c) bằng công thức sau :

$$\Delta^c_{ij} = F(i, j) - F(i, j-1) \qquad (3.4.1)$$

với i=1,2,3,…,n và j=1,2,3,…,m

Trong đó F(i, j) là giá trị cường độ xám của pixel tại vị trí (i, j) của ảnh x.

3.4.2.1.2 Theo chiều dọc

Tương tự ta có thể dò cạnh theo sự thay đổi của các dòng (Δ^r)

$$\Delta^r_{ij} = F(i, j) - F(i-1, j) \qquad (3.4.2)$$
với i=1,2,3,…,n và j=1,2,3,…,m

3.4.2.1.3 Ví dụ

Xét một phần nhỏ của một bức ảnh như sau :

0 0 0 128 128 128
0 0 0 128 128 128
0 0 0 128 128 128
0 0 0 128 128 128

Lúc này giá trị của Δ^c là

0 0 128 0 0
0 0 128 0 0
0 0 128 0 0
0 0 128 0 0

Còn giá trị của Δ^r là

0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

3.4.2.2 Dò cạnh theo một hướng bất kỳ

Tương tự ta cũng có thể dò cạnh theo một hướng nào đó.

Từ ý tưởng cơ bản trên, người ta sử dụng phương pháp mặt nạ (mark). Mặt nạ (M) là một ảnh nhỏ, thường là 3x3 pixel

$$M = \begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

Đặt G là ảnh nhỏ cũng có 3x3 pixel có giá trị tương ứng với giá trị của 3x3 pixel trong ảnh đang dò cạnh. G(i, j) có tâm ảnh ứng với vị trí (i, j) trong ảnh đang dò cạnh

$$G(i,j) = \begin{matrix} F(i-1,j-1) & F(i-1,j) & F(i-1,j+1) \\ F(i,j-1) & F(i,j) & F(i,j+1) \\ F(i+1,j-1) & F(i+1,j) & F(i+1,j+1) \end{matrix}$$

Định nghĩa phép toán * như sau:

$$M*G(i,j) = a.F(i-1,j-1)+b.F(i-1,j)+c.F(i-1,j+1)+d.F(i,j-1)+e.F(i,j)+ \\ +f.F(i,j+1)+g.F(i+1,j-1)+h.F(i+1,j)+i.F(i+1,j+1) \quad (3.4.3)$$

Lúc này việc dò cạnh sẽ được thực hiện một cách khá đơn giản. Một pixel ở vị trí (i,j) trong ảnh mới (ảnh sau khi dò cạnh) có giá trị cường độ xám bằng:

$$F'(i,j) = M*G(i,j) \quad (3.4.4)$$

Áp dụng công thức (3.4.4) cho tất cả các pixel trong ảnh F, ta có ảnh F' là ảnh đã được dò cạnh.

Ví dụ: Mặt nạ tương ứng với Δ^c và Δ^r ở trên là

$$\begin{matrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} \quad \Delta^c \qquad\qquad \begin{matrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} \quad \Delta^r$$

Áp dụng công thức (3.4.4) cho mặt nạ Δ^c:

$$F'(i,j) = d.F(i,j-1)+e.F(i,j) \qquad \text{với } d=-1,\ e=1 \\ = (-1).F(i,j-1)+(1).F(i,j) \\ = F(i,j) - F(i,j-1) \\ = \Delta^c_{ij}$$

Ảnh được dò cạnh sẽ phụ thuộc vào mặt nạ đưa ra. Mặt nạ đưa ra phản ánh được hướng của việc dò cạnh. Chúng ta hoàn toàn có thể dò cạnh theo nhiều hướng.

Ví dụ mặt nạ A

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{matrix}$$

có thể dò cạnh theo hướng ngang và dọc (mặt nạ này do $\Delta^c+\Delta^r$)

Một số trường hợp cần lưu ý :

+ Giá trị F'(i, j) có thể khá lớn (>255) sau khi tính.

Ví dụ với ảnh F như sau :

0	0	0	0
0	200	200	200
0	200	200	200
0	200	200	200
0	200	200	200

Được tính với mặt nạ A trên. Lúc đó F'(1, 1) = (2)200 = 400 >255

Điều này không hợp lý vì giá trị cường độ xám trong một bức ảnh chỉ từ 0 $\rightarrow$ 255. Vì vậy để giải quyết trường hợp này ta chỉ cần nhân tỉ lệ cho tất cả các giá trị cường độ xám trong ảnh sau cho nhỏ hơn 255. Áp dụng cho ví dụ trên, ta nhân ½ cho ảnh F'. Lúc đó F'(1,1) = 400(1/2)=200 <255

+ Tương tự trường hợp trên F'(i, j) có thể âm.

Ví dụ cũng mặt nạ trên cho ảnh F sau:

0	0	0	0
1	1	0	0
1	1	0	0
0	0	0	0

F'(1,2) = (-1)1= -1 <0

Để giải quyết trường hợp này ta chỉ cần cộng thêm một giá trị dương sao cho toàn ảnh có giá trị cường độ xám lớn hơn hoặc bằng không. Áp dụng cho ví dụ trên, cộng 1 cho ảnh F'. Lúc đó F'(1,2) = 0 $\geq$0.

Có hai phương pháp chính để dò cạnh : Mẫu thích hợp nhất (template matching, gọi tắt là TM), Độ dốc khác biệt (differential gradient, gọi tắt là DG) và một số phương pháp khác.

3.4.2.3 Phương pháp TM

3.4.2.3.1 Phương pháp

Tại từng điểm trong quá trình dò cạnh, ta sẽ chọn mặt nạ sao cho giá trị cường độ xám mới của pixel là lớn nhất.

$$F'(i, j) = \max \{ M_i * G(i, j) / i=1,2,3,\ldots,n \} \quad (3.4.5)$$

Giá trị n thường bằng 8 hay 12

Ý tưởng này giống như việc ta đang so từng mặt nạ với ảnh, mặt nạ nào hợp nhất thì chọn. Các mặt nạ M_i thực chất được tạo ra từ 1 mặt nạ bằng cách hoán vị các hệ số xung quanh tâm mặt nạ theo vòng tròn.

Trong phương pháp này, hướng của cạnh là hướng của mặt nạ lớn nhất được chọn.

3.4.2.3.2 Ví dụ

Từ một mặt nạ ban đầu

$$M = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$$

Ta có được 8 mặt nạ bằng cách hoán vị mặt nạ M trên là :

$$M_1 = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad M_2 = \begin{matrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix}$$

$$M_3 = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad M_4 = \begin{matrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix}$$

$$M_5 = \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix} \qquad M_6 = \begin{matrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{matrix}$$

$$M_7 = \begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix} \qquad M_8 = \begin{matrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{matrix}$$

3.4.2.3.3 Một số mặt nạ thường dùng

3.4.2.3.2.1 Mặt nạ Prewitt

$$M = \begin{matrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{matrix}$$

3.4.2.3.2.2 Mặt nạ Kirsch

$$M = \begin{matrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{matrix}$$

3.4.2.3.2.3 Mặt nạ Robinson "3"

$$M = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$$

3.4.2.3.2.4 Mặt nạ Robinson "5"

$$M = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

3.4.2.4 Phương pháp DG

3.4.2.3.1 Phương pháp

Trong phương pháp này ta luôn chọn hai mặt nạ để dò cạnh. Hai mặt nạ này đại diện cho hai hướng ngang và dọc (M_x và M_y). Sau đó tính theo công thức sau :

$$F'(i, j) = \sqrt{(Mx * G(i, j))^2 + (My * G(i, j))^2} \qquad (3.4.6)$$

Trong một số trường hợp ta có thể dùng các công thức khác :

$$F'(i, j) = |M_x * G(i, j)| + |M_y * G(i, j)|$$

hoặc (3.4.7)

$$F'(i, j) = \max(|M_x * G(i, j)|, |M_y * G(i, j)|)$$

Trong phương pháp DG, hướng của cạnh được tính như sau :

Đặt $g_x = M_x * G(i, j)$ và $g_y = M_y * G(i, j)$

$$\theta = \text{arctg}\left(\frac{g_y}{g_x}\right) \qquad (3.4.8)$$

Rõ ràng phương pháp TM tính toán hướng của cạnh ít hơn DG mặc dù DG sẽ chính xác hơn.

3.4.2.3.2 Một số mặt nạ thường dùng

3.4.2.3.2.1 Mặt nạ Roberts

$$M_x = \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix} \qquad M_y = \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$$

3.4.2.3.2.2 Mặt nạ Sobel

$$M_x = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad M_y = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

3.4.2.3.2.3 Mặt nạ Prewittl

$$M_x = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad M_y = \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$$

3.4.2.5 Các phương pháp khác

3.4.2.5.1 Mặt nạ Frei - Chen

Sử dụng 9 mặt nạ 3x3 : $M_1 \rightarrow M_9$

$$M_1 = \begin{matrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{matrix} \qquad M_2 = \begin{matrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{matrix} \qquad M_3 = \begin{matrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{matrix}$$

$$M_4 = \begin{matrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{matrix} \qquad M_5 = \begin{matrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{matrix} \qquad M_6 = \begin{matrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{matrix}$$

$$M_7 = \begin{matrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ -1 & -2 & 1 \end{matrix} \qquad M_8 = \begin{matrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{matrix} \qquad M_9 = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Trong 9 mặt nạ, $M_1 \rightarrow M_4$ thể hiện sự dò cạnh, $M_5 \rightarrow M_8$ thể hiện sự dò đường, M_9 thể hiện trung bình giá trị cường độ xám của pixel đang xét.

Tại mỗi pixel của ảnh F' được tính bởi công thức

$$F'(i, j) = \left(\frac{\sum_{i=1}^{4} (M_i * F)^2}{\sum_{i=1}^{9} (M_i * F)^2} \right)^{1/2} \qquad (3.4.9)$$

Phương pháp tiện lợi ở chổ linh động nhưng bất tiện ở chổ tính toán khá phức tạp.

3.4.2.5.2 Mặt nạ Marr-Hildreth

Là một thuật toán có thể liên kết tất cả các hướng trong quá trình dò cạnh.

Đầu tiên, thuật toán sẽ làm trơn bức ảnh thông qua hàm Gaussian:

$$G(\sigma,x,y) = \sigma^2 . e^{\frac{-(x^2+y^2)}{2\sigma^2}} \qquad (3.4.10)$$

Trong đó σ là phương sai

$$\sigma = \sqrt{\sum_{i-0}^{N} \frac{(x_i - \bar{x})^2}{N-1}}$$

(x,y) là vị trí pixel đang xét

Hàm này được áp đặt vào bức ảnh bằng cách dùng như một mặt nạ (gọi là mặt nạ Gaussian) để làm trơn bức ảnh.

Sau khi làm trơn ảnh ta dùng mặt nạ Laplacian để dò cạnh.

Mặt nạ Laplacian là

1	-2	1
-2	4	-2
1	-2	1

Để tìm mặt nạ Gaussian ta cần tính hàm Gaussian và lưu giá trị trên một lưới gồm nxn pixel (n là số dòng , cột trong mặt nạ. n được chọn sao cho mặt nạ chứa đủ các giá trị khác của hàm Gaussian , thường thì n=$3\sqrt{2}\,\partial$). Mặt nạ Gaussian được tính sao cho vị trí trung tâm của mặt nạ là vị trí (0,0) của hàm Gaussian. Ví dụ với σ =1.2 , ta có mặt nạ Gaussian như sau :

0.0000	0.0000	0.0000	0.0000	0.0001	0.0002	0.0001	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0002	0.0013	0.0039	0.0055	0.0039	0.0013	0.0002	0.0000	0.0000
0.0000	0.0002	0.0027	0.0157	0.0447	0.0632	0.0447	0.0157	0.0027	0.0002	0.0000
0.0000	0.0013	0.0157	0.0895	0.2537	0.3590	0.2537	0.0895	0.0157	0.0013	0.0000
0.0001	0.0039	0.0447	0.2537	0.7190	1.0175	0.7190	0.2537	0.0447	0.0039	0.0001
0.0002	0.0055	0.0632	0.3590	1.0175	1.4400	1.0175	0.3590	0.0632	0.0055	0.0002
0.0001	0.0039	0.0447	0.2537	0.7190	1.0175	0.7190	0.2537	0.0447	0.0039	0.0001
0.0000	0.0013	0.0157	0.0895	0.2537	0.3590	0.2537	0.0895	0.0157	0.0013	0.0000
0.0000	0.0002	0.0027	0.0157	0.0447	0.0632	0.0447	0.0157	0.0027	0.0002	0.0000
0.0000	0.0000	0.0002	0.0013	0.0039	0.0055	0.0039	0.0013	0.0002	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0001	0.0002	0.0001	0.0000	0.0000	0.0000	0.0000

Bây giờ ta nhập hai mặt nạ Laplacian và Gaussian để sinh ra một mặt nạ, gọi là Laplacian-Gaussian:

0.0000	0.0001	0.0008	0.0012	-.0008	-.0029	-.0008	0.0012	0.0008	0.0001	0.0000
0.0001	0.0019	0.0086	0.0131	-.0085	-.0307	-.0085	0.0131	0.0086	0.0019	0.0001
0.0008	0.0086	0.0393	0.0599	-.0390	-.1396	-.0390	0.0599	0.0393	0.0086	0.0008
0.0012	0.0131	0.0599	0.0913	-.0594	-.2128	-.0594	0.0913	0.0599	0.0131	0.0012
-.0008	-.0085	-.0390	-.0594	0.0387	0.1385	0.0387	-.0594	-.0390	-.0085	-.0008
-.0029	-.0307	-.1396	-.2128	0.1385	0.4956	0.1385	-.2128	-.1396	-.0307	-.0029
-.0008	-.0085	-.0390	-.0594	0.0387	0.1385	0.0387	-.0594	-.0390	-.0085	-.0008
0.0012	0.0131	0.0599	0.0913	-.0594	-.2128	-.0594	0.0913	0.0599	0.0131	0.0012
0.0008	0.0086	0.0393	0.0599	-.0390	-.1396	-.0390	0.0599	0.0393	0.0086	0.0008
0.0001	0.0019	0.0086	0.0131	-.0085	-.0307	-.0085	0.0131	0.0086	0.0019	0.0001
0.0000	0.0001	0.0008	0.0012	-.0008	-.0029	-.0008	0.0012	0.0008	0.0001	0.0000

Do mặt nạ trên có giá trị là số thực, dẫn đến tính toán khá phức tạp. Vì thế để dễ dàng tính toán ta cần biến đổi thành mặt nạ gồm các giá trị nguyên như sau :

0	0	0	0	0	-1	0	0	0	0	0
0	0	2	4	-2	-10	-2	4	2	0	0
0	2	13	20	-13	-46	-13	20	13	2	0
0	4	20	30	-19	-71	-19	30	20	4	0
0	-2	-13	-19	13	46	13	-19	-13	-2	0
-1	-10	-46	-71	46	166	46	-71	-46	-10	-1
0	-2	-13	-19	13	46	13	-19	-13	-2	0
0	4	20	30	-19	-71	-19	30	20	4	0
0	2	13	20	-13	-46	-13	20	13	2	0
0	0	2	4	-2	-10	-2	4	2	0	0
0	0	0	0	0	-1	0	0	0	0	0

Tóm lại thuật toán chỉ việc dùng một mặt nạ Laplacian-Gaussian thay cho việc dùng hai mặt nạ để dò cạnh. Phương pháp này cần nhiều thời gian để tính toán vì vậy ta cần lưu ý các ảnh có kích cở lớn khi sử dụng.

3.4.3 Đường

Sự khác nhau giữa đường thẳng và cạnh thẳng sẽ trở nên rõ ràng khi ta thực hiện việc dò cạnh cho ảnh có chứa một đường thẳng mỏng. Sau khi thực hiện việc dò cạnh thì những cạnh thẳng sẽ xuất hiện, chính là nơi gặp nhau của hai vùng còn đường thẳng sẽ trở thành một vùng đối tượng mà có hai cạnh ở hai bên, tức là việc dò cạnh của đường thẳng sẽ cho kết quả là hai đường thẳng song song thay vì chỉ một đường thẳng.

Ví dụ cho ảnh F như sau:

0	0	128	0	0	0	0	0	0	0	0
0	0	128	0	0	0	128	128	128	128	0
0	0	128	0	0	0	128	128	128	128	0
0	0	128	0	0	0	128	128	128	128	0
0	0	128	0	0	0	0	0	0	0	0
0	0	128	0	0	0	0	0	0	0	0

Sau khi áp dụng lọc cạnh theo mặt nạ Robinson "3" ta có ảnh F' sau khi lọc cạnh là:

0	128	-128	0	0	0	0	0	0	0
0	128	-128	0	128	128	0	0	-128	0
0	128	-128	0	128	128	0	0	-128	0
0	128	-128	0	128	128	0	0	-128	0
0	128	-128	0	0	0	0	0	0	0
0	128	-128	0	0	0	0	0	0	0

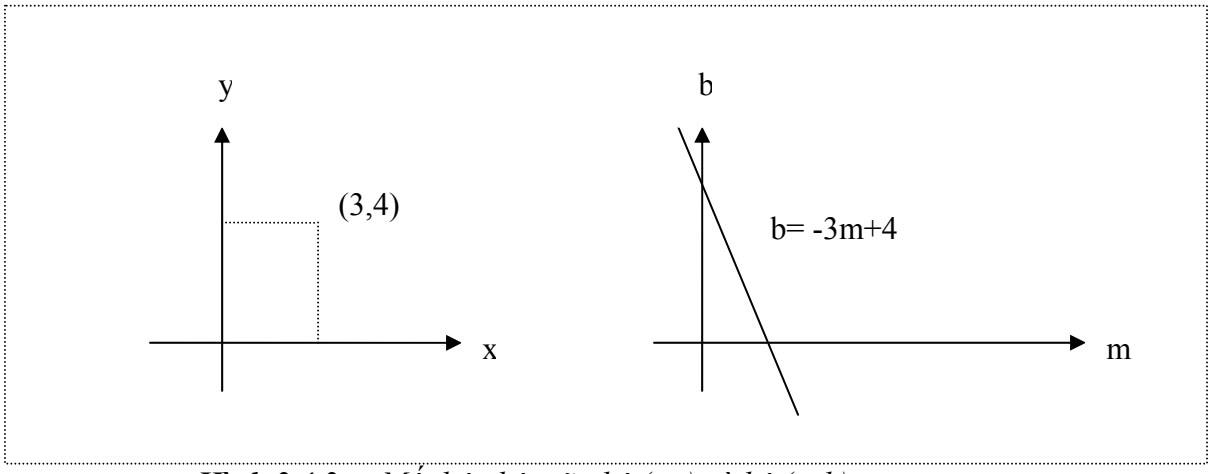
Vì thế việc dò đường cần phải có những phương pháp khác so với dò cạnh . Để dò đường trong ảnh có thể dùng phương pháp Hough. Khái niệm cơ bản để dò đường trong phương pháp Hough là mối liên quan giữa điểm và đường.

Đường thẳng có phương trình là y=mx+b (3.4.11)

Trong đó x,y là toạ độ các điểm trong hệ toạ độ (x,y), m và b là hai tham số.

Phương trình (3.4.11) tương đương phương trình : b= -mx+y (3.4.12)

Lúc này nếu ta xem (m,b) là toạ độ các điểm và x,y là hai tham số thì phương trình (3.4.12) sẽ là phương trình đường thẳng trong không gian (m,b)(gọi là không gian Hough). Như vậy một điểm trong hệ toạ độ (x,y) sẽ tương ứng với một đường thẳng trong hệ toạ độ (m,b).



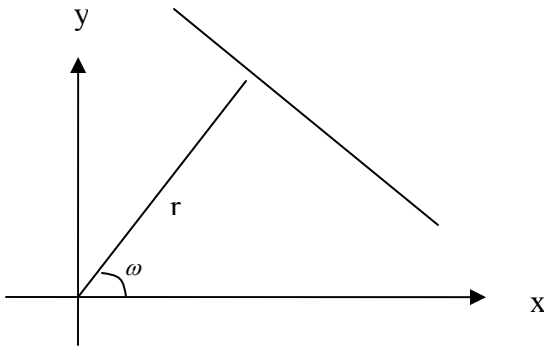
Hình 3.4.3.a: *Mối liên hệ giữa hệ (x,y) và hệ (m,b)*

Chú ý rằng có một phép tương hổ giữa hai không gian trên, tức là những điểm thẳng hàng trong hệ tọa độ (x,y) thì tương ứng có những đường thẳng đồng qui tại một điểm trong hệ (m,b). Nhưng có một trường hợp đặt biệt, nếu các điểm này nằm trên đường thẳng song song với trục Oy trong hệ tọa độ (x,y) thì tương ứng sẽ là những đường thẳng song song với nhau mà không đồng qui. Trong các loại hình ảnh thì trường hợp này không phải ít. Vì vậy người ta muốn chuyển sang hệ tọa độ (r,ω) thay cho (m,b) trong không gian Hough bằng cách đổi biến như sau:

$$\sin(\omega) = \frac{-1}{\sqrt{m^2+1}}, \quad \cos(\omega) = \frac{m}{\sqrt{m^2+1}}, \quad r = \frac{-b}{\sqrt{m^2+1}}$$

Lúc đó phương trình (3.4.12) sẽ thành phương trình:

$$r = x.\cos(\omega) + y.\sin(\omega) \qquad (3.4.13)$$



Hình 3.4.3.b **:** *Minh họa r và ω trong hệ (x,y)*

Phương trình đường cong (3.4.13) khắc phục được trường hợp đặt biệt của phương trình (3.4.12). Tóm lại trong không gian Hough, một điểm biến thành một đường cong và các điểm thẳng hàng trong không gian (x,y) sẽ tương ứng có những đường cong giao nhau tại một điểm trong không gian (r,ω).

Trong ứng dụng phương pháp Hough dùng để lọc đường, ta thực hiện theo hai cách sau :

Cách 1:

Trước hết áp dụng phương pháp Hough để chuyển ảnh trong hệ (x,y) sang hệ (r,ω), trong không gian Hough này ta tìm những đỉnh (là vị trí đồng qui của các đường cong trong hệ (r,ω)). Từ đó suy ra hai tham số r, ω. Sau đó dùng hai tham số này để tìm những điểm trong ảnh gốc tạo thành đường thẳng cần dò. Chú ý ω chỉ xét từ $0 \to 180$ do tính chất đối xứng của đường thẳng.

Cách 2:

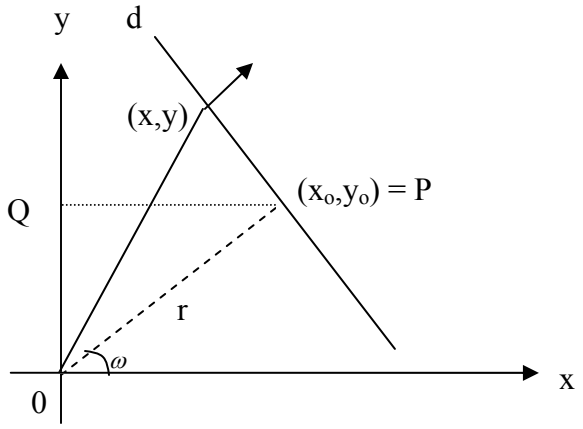
Để xác định các đường thẳng, đầu tiên ta cần xác định hướng của những đường thẳng thông qua mặt nạ bằng công thức tương tự (3.4.8)sau :

$$\theta = \text{arctg}(\ g_y / g_x\ ) \qquad (3.4.14)$$

Từ θ này ta tìm được r bằng công thức (3.4.13). Sau đó dùng r, ω tìm được xác định đường thẳng là tập hợp các điểm có cùng r, ω.

Cải tiến cách 2:

Trong quá trình tính ω và r, ta phải dùng các công thức có chứa các hàm lượng giác vì vậy khá phức tạp. Người ta chuyển sang một cách tính r khác đơn giản hơn.



Hình 3.4.3.c **:** *Thể hiện cách chuyển sang cách tính đơn giản hơn*

Ta có:

$$\text{tg}(\omega) = \frac{y_o}{x_0} \Rightarrow \frac{g_y}{g_x} = \frac{y_o}{x_o} \qquad (1)$$

Và PQ $\perp$ d $\Rightarrow (x_o, y_o).(x - x_o, y - y_o) = 0$

$$\Leftrightarrow (x - x_o)x_o + (y - y_o)y_o = 0 \qquad (2)$$

Từ (1) và (2) suy ra : $x_o = v.g_x, y_o = v.g_y$

$$\text{với} \qquad v = \frac{x.g_x + y.g_y}{g_x^2 + g_y^2}$$

Mặt khác do ΔOPQ vuông nên r=$\sqrt{x_o^2 + y_o^2}$

$$\Rightarrow r = \frac{x.g_x + y.g_y}{\left(g_x^2 + g_y^2\right)^{1/2}}$$

Từ đó ta có hai tham số tg(ω) và r một cách đơn giản hơn cách 2 ban đầu.

3.5 Thao tác hình học

Biến đổi hình học dựa trên sự thay đổi vị trí của những pixel trên ảnh. Nói một cách khác, nó làm dịch chuyển các pixel trên đến những vị trí khác trên ảnh. Các thao tác hình học gồm:

- Lấy vùng (windowing)
- Tịnh tiến (translation)
- Co (scaling)
- Quay (rotation)
- Biến dạng (warp)

3.5.1 Lấy vùng (windowing)

Là lấy một vùng hình chữ nhật nhỏ chứa trong ảnh gốc.

Ảnh ban đầu Ảnh được lấy ra

Hình 3.5.1 **:** *Minh họa thao tác lấy vùng*

3.5.2 Tịnh tiến (translation)

Là di chuyển các pixel trong ảnh theo các hướng X hay Y bằng một vài pixel. Trong quá trình tịnh tiến cần phải thêm vào hàng và cột nếu đối tượng tịnh tiến vượt quá biên chứa ảnh.

Tịnh tiến dương là tịnh tiến theo hướng làm tăng chỉ số hàng và cột. Ngược lại, tịnh tiến âm là tịnh tiến theo hướng làm giảm chỉ số hàng và cột.

Ảnh ban đầu Ảnh đã tịnh tiến âm

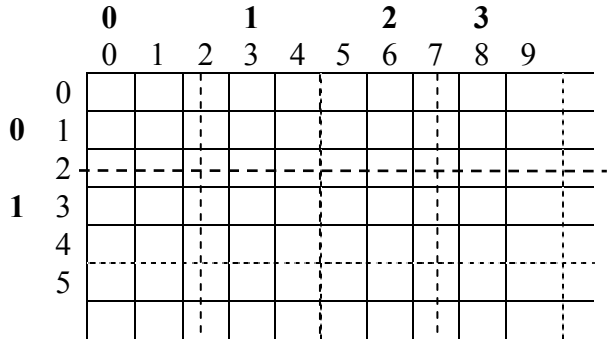
Hình 3.5.2 **:** *Minh họa thao tác tịnh tiến*

3.5.3 Co (scaling)

Là làm cho đối tượng trong ảnh lớn hơn (phóng to) hay nhỏ hơn (thu nhỏ) ban đầu. Thao tác này không chỉ làm cho số pixel trong ảnh thay đổi mà còn ảnh hưởng đến kích cở của mỗi pixel.

Nếu ta co theo hệ số có dạng 2^n thì tương đối đơn giản. Ví dụ co theo hệ số là 2 thì một pixel sẽ biến thành một khối 2x2 pixel trong ảnh mới.

Nhưng nếu co theo một hệ số tùy ý thì không đơn giản vì một pixel trong ảnh nguồn không dễ dàng ánh xạ thành một số pixel trong ảnh mới. Ví dụ ảnh có kích thước 256x256 pixel được co thành ảnh có kích thước 100x100 pixel. Như vậy hệ số co là 1/2.56, nghĩa là một đơn vị diện tích trong ảnh mới sẽ tương ứng $2.56^2 = 6.5536$ đơn vị diện tích trong ảnh gốc.



Hình 3.5.3.a: *Minh họa vị trí tọa độ của ảnh 256x256 (đường liền) và ảnh 100x100 (đường gạch đứt).*

	1.0	1.0	0.56
1.0	A	B	C
1.0	D	E	F
0.56	G	H	I

Hình 3.5.3.b *: Minh họa pixel tại vị trí (0,0) trong ảnh co (100x100) và $A \rightarrow I$ là vị trí trong ảnh gốc*

Giá trị tại điểm (0,0) trong ảnh mới sẽ bằng

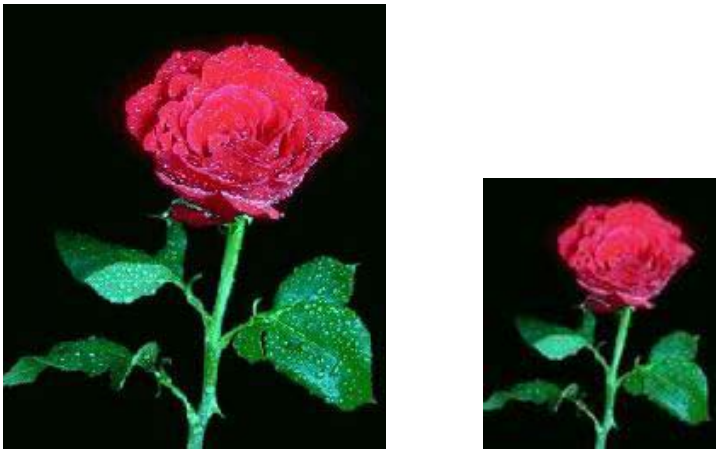
$$F'(0,0) = w_A.A+w_B.B+w_C.C+w_D.D+w_E.E+w_F.F+w_G.G+w_H.H+w_I.I$$

Trong đó: A,B,…..,I là giá trị mức độ xám tương ứng vị trí trong ảnh gốc.
w_A, w_B….,w_I là phân số giữa diện tích vùng tương ứng và tổng diện tích vùng.
Ứng với ví dụ này ta có :

$$W_A = 1/2.56^2$$
$$W_I = 0.56^2/2.56^2$$

Chú ý ta luôn có :

$$\sum_{i=A,B,...I} w_i = 1$$

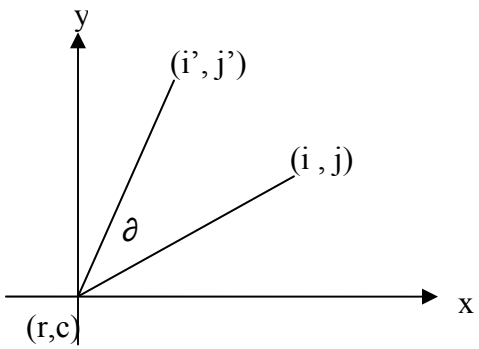


Ảnh ban đầu Ảnh đã co

Hình 3.5.3.c **:** *Minh họa thao tác co*

3.5.4 Quay (rotation)

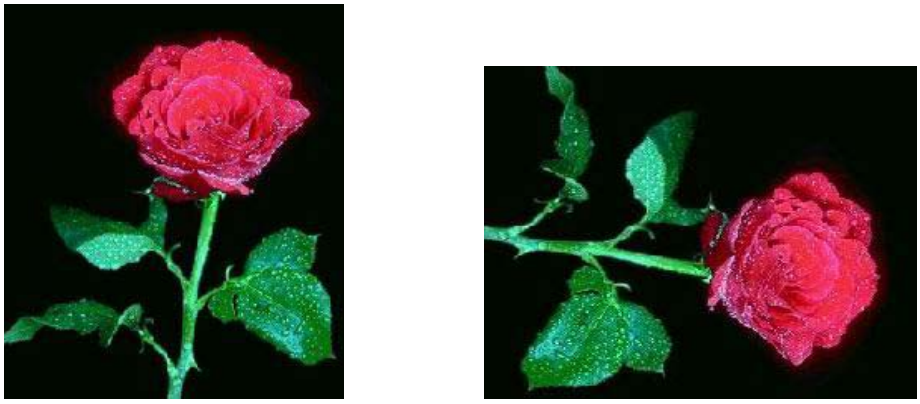
Là quay các điểm trong ảnh một góc quanh một điểm nào đó, điểm này thường là tâm của ảnh hơn là điểm (0,0).



Hình 3.5.4.a : *Minh họa thao tác quay một điểm (i , j) quanh điểm (r,c)*

Khi cần quay điểm (i, j) trên ảnh quanh điểm (r,c) một góc ∂ và (i, j) cách (r,c) một khoảng R thì điểm mới (i', j') sau khi quay được tính bằng công thức sau:

$$j' = j.\cos(\partial) - i.\sin(\partial) + c$$
$$i' = j.\sin(\partial) + i.\cos(\partial) + r$$



Ảnh ban đầu — Ảnh sau khi quay một góc 90 độ

Hình 3.5.4.b **:** *Minh họa thao tác quay*

3.5.5 Biến dạng (warp)

Là làm thay đổi hình dạng của một ảnh theo một ánh xạ:

Ảnh nguồn — Ảnh biến dạng

$$(x,y) \longrightarrow (x',y') = (G_r(x,y), G_c(x,y))$$

với

$$G_r(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{ij} x^i y^i$$

$$G_c(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} b_{ij} x^i y^j$$

Trong đó n,m là bậc biến dạng

a_{ij},b_{ij} là hệ số biến dạng

3.6 Điểm nhiễu (Noise)

Những điểm nhiễu là những pixel có mức độ xám thay đổi một cách ngẫu nhiên bởi việc số hóa, chuyển giao hay xử lý trên một ảnh. Bởi vì những điểm lân cận này được phát sinh một cách ngẫu nhiên nên nó ảnh hưởng không tốt đến ảnh.

Có 3 loại điểm nhiễu quan trọng:

+ Điểm nhiễu độc lập (signal-independent noise) : những điểm này có được là do có một số điểm trong ảnh có giá trị cường độ xám ngẫu nhiên phát sinh thêm hoặc mất đi.

+ Điểm nhiễu phụ thuộc (signal-dependent noise) : những điểm này có được là do mỗi giá trị cường độ xám của pixel ngẫu nhiên tăng lên hoặc giảm đi.

+ Điểm nhiễu muối tiêu (salt and pepper noise) : trường hợp này thường xuất hiện ở những ảnh trắng đen. Nó xảy ra khi một số điểm trắng biến thành đen và ngược lại. Điều này xảy ra là do khi đặt ngưỡng (Threshold) một ảnh.

Tất nhiên, điểm nhiễu có thể do nhiều nguồn gây ra và nhiều loại điểm nhiễu có thể xuất hiện trong cùng một ảnh.

Mặc dù chúng ta không thể khử hết tất cả các điểm nhiễu có thể có, nhưng ta có thể làm giảm sự nhiễu này bằng cách xử lý trơn ảnh (smoothing). Việc xử lý trơn ảnh là tìm ảnh trung bình (là ảnh có giá trị cường độ xám của một pixel là giá trị cường độ xám trung bình của các pixel cùng vị trí ảnh của nhiều ảnh tương tự nhau). Bức ảnh trung bình này không mất hết các điểm nhiễu nhưng nó có thể làm cho bức ảnh dễ dàng thao tác hơn.

Có nhiều khả năng là ta không thể có được nhiều ảnh tương tự để có thể tìm ảnh trung bình. Vì vậy ta có thể làm trơn ảnh bằng cách tìm giá trị cường độ xám trung bình của một pixel từ một vùng nhỏ lân cận (thường là vùng ảnh nhỏ 3x3 pixel) trong ảnh. Giá trị trung bình này thường được tính thông qua mặt nạ :

1	1	1
1	1	1
1	1	1

(a)

1	1	1
1	2	1
1	1	1

(b)

1	2	1
2	4	2
1	2	1

(c)

Bằng cách nhân tương ứng các mặt nạ này vào vùng đang xét, sau đó chia 9 nếu dùng mặt nạ (a), chia 10 nếu dùng mặt nạ (b), chia 16 nếu dùng mặt nạ (c). Nhưng cách này có khuynh hướng làm mờ đi những cạnh thẳng, đường thẳng và những đóm trong ảnh.

Một cách khác để làm giảm sự nhiễu mà có thể giảm sự làm mờ cạnh là ta đặt ngưỡng. Giá trị cường độ xám mới bằng:

$$F'(i,j) = \begin{cases} \dfrac{\left(\sum_{r=i-1}^{i+1}\sum_{c=j-1}^{j+1} F(r,c)\right) - F(i,j)}{8} & \left|\dfrac{\left(\sum_{r=i-1}^{i+1}\sum_{c=j-1}^{j+1} F(r,c)\right) - 2F(i,j)}{8}\right| < T \\ F(i,j) & \neq \end{cases}$$

Một cách khác nữa mà không làm thay đổi gì đến cạnh trong ảnh là lọc trung bình (median filter). Giá trị trung bình này được tính thông qua một vùng nhỏ lân cận của pixel đang xét và nó sẽ thay thế giá trị giá trị cường độ xám của pixel đang xét. Giá trị trung bình này được tính bằng cách chọn phần tử giữa của mảng các giá trị đã được sắp xếp. Mảng này gồm các giá trị của lân cận điểm đang xét và giá trị điểm đang xét. Vùng nhỏ lân cận điểm đang xét gồm 5 hay 13 pixel và có hình dạng :

	*	
*	(*)	*
	*	

5

		*		
	*	(*)	*	
*	*	*	*	*
	*	*	*	
		*		

13

Trong đó điểm được đóng ngoặc chính là điểm đang xét.

CHƯƠNG 2: NHẬN DẠNG - PHÂN LỚP - ĐẾM ĐỐI TƯỢNG

1 Nhận dạng và phân lớp đối tượng

Một vấn đề nổi bật của thị giác máy tính là làm cho máy tính có thể nhận dạng được đối tượng. Để có thể nhận dạng được các đối tượng trong ảnh, chúng ta cần phải thực hiện ba bước :

+ Tìm các đặc trưng cho đối tượng.
+ Xác định lớp cho các đối tượng.
+ Loại ra các đối tượng không biết.

Tìm các đặc trưng cho đối tượng tức là cần thu thập và tìm kiếm các thông tin liên quan đến đối tượng trong ảnh. Xác định lớp cho đối tượng là dựa vào các đặc trưng tìm ra mà ta xếp đối tượng đó vào lớp thích hợp nhất. Chúng ta cần áp dụng nhiều thao tác trên ảnh mà các chương trước đã nêu để tìm ra các dữ liệu liên quan.

Việc phân lớp và nhận dạng đối tượng có thể dựa vào một số phương pháp thông thường như: thống kê (statistical), xác suất (decision), so mẫu đối tượng (template matching), nhận dạng dựa vào cấu trúc(structural). Không có một phương pháp nào luôn đúng hay sai trong việc phân lớp và nhận dạng đối tượng. Tùy theo vấn đề bài toán mà ta áp dụng các phương pháp trên.

1.1 Đặc trưng (Feature)

Một đặc trưng là một sự đo lường được thực hiện trên một ảnh hay một vùng của ảnh. Ví dụ diện tích, chu vi, hình dạng tròn, số lỗ trong đối tượng,….là những đặc trưng. Đặc trưng thường thể hiện các đặc điểm của đối tượng trong ảnh.Vì thế đặc trưng được dùng để phân biệt các đối tượng. Tức là những đối tượng nào có cùng đặc trưng thì có thể cùng một lớp. Nhưng không phải đặc trưng nào cũng hữu ích trong việc phân biệt đối tượng. Ví dụ để nhận dạng các con chó, cừu, ngựa trong một ảnh thì ta không thể dùng số chân làm đặc trưng để phân biệt. Còn nếu ta sử dụng bờm làm đặc trưng để phân biệt thì ngựa sẽ được nhận ra trong ảnh. Nhưng vấn đề là ta làm cách nào xác định những cái bờm trong ảnh, việc này thật sự không dễ dàng. Vì vậy chúng ta cần tìm những đặc trưng khác dễ xác định, dễ tính toán hơn để làm đặc trưng nhận dạng. Những tính chất liên quan đến hình học thì dễ dàng sử dụng trong vấn đề này. Tóm lại, việc nhận dạng đối tượng thì khó khăn hơn việc phân lớp, vấn đề tìm ra đặc trưng nhận dạng phù hợp là quan trọng để có thể nhận dạng đối tượng.

Công việc chọn đặc trưng liên quan đến việc loại ra những đặc trưng không có lợi cho đến khi còn lại những đặc trưng dễ sử dụng và hữu ích. Vì vậy chúng ta phải cố gắng dùng tất cả những đặc trưng có thể trong việc chọn ra đặc trưng tốt nhất. Nhưng công việc này không khả thi, chúng ta cần phải rút ngắn quá trình thực hiện. Bằng cách tính toán những đặc trưng của những đối tượng đã biết, chúng ta có thể tìm ra những đặc trưng thích hợp là những đặc trưng có giá trị tương tự với nhau trong cùng một lớp. Ví dụ tất cả những đồng xu thì có cùng chu vi và diện tích, những cây bút chì thì có cùng một hình dạng cơ bản. Một giá trị đo lường thì hữu ích cho vấn đề này là phương sai (variance) được tính bằng công thức:

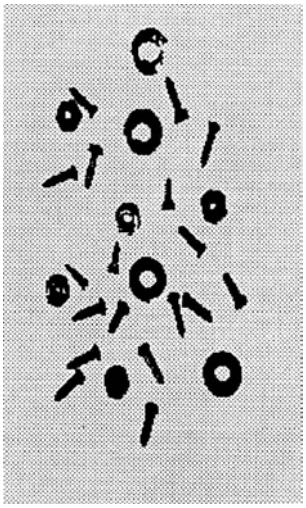
$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)$$

Trong đó: N là số đối tượng được xét

x_i là giá trị đo được của đối tượng thứ i

$\bar{x}$ là trung bình của x_i với i=1,…,N

Nếu phương sai nhỏ thì đặc trưng ứng với phương sai đó tốt bởi vì nó có nghĩa là đặc trưng này có giá trị ổn định cho các đối tượng trong cùng một lớp.



Hình 1.1.a **:** *Minh họa ảnh gồm một vài đinh ốc, vòng đệm và đai ốc*

Khảo sát một bức ảnh gồm một vài đinh ốc, vòng đệm và đai ốc. Như vậy nghĩa là có ba lớp đối tượng trong ảnh. Giả sử rằng bức ảnh này đã được phân tích và nhận dạng. Chúng ta sẽ khảo sát những đặc trưng nào hữu ích để nhận dạng được chúng. Những đặc trưng này được tính toán cho mỗi đối tượng và sau đó phương sai của từng lớp cũng được tính toán. Xét các đặc trưng diện tích, chu vi và hình dáng tròn của ảnh trên cho từng đối tượng, ta có bảng dữ liệu về giá trị trung bình (kỳ vọng) và phương sai của từng lớp như sau:

Phương sai - Kỳ vọng						
Lớp	Diện tích		Chu vi		Hình dáng tròn	
Đinh ốc	492.4	116.9	77.56	58.25	0.2723	2.35
Vòng đệm	2317	375.5	197	123.7	1.867	3.45
Đai ốc	167.7	223.7	124.1	71.89	0.4485	1.91

Để có thể so sánh sự khác nhau của các đặc trưng trên đối với các lớp đối tượng, ta cần tìm ra một giá trị có thể thể hiện sự khác nhau này. Nhìn vào bảng dữ liệu trên, ta thấy các đặc trưng có phương sai lớn thì kỳ vọng cũng lớn. Từ điều này ta có thể tìm ra một con số có thể thể hiện sự khác nhau về đặc trưng của các lớp. Con số này gọi là hệ số của sự thay đổi (coefficient of variantion) được tính bằng công thức :

$$V=\frac{\sigma}{\bar{x}}$$

Trong đó: σ là độ lệch tiêu chuẩn, tức là căn bậc hai của phương sai

$\bar{x}$ là kỳ vọng

Giá trị V cho ba lớp trên được thể hiện dưới bảng sau:

Hệ số của sự thay đổi (V)			
Lớp	Diện tích	Chu vi	Hình dáng tròn
Đinh ốc	0.1898	0.1512	0.2218
Vòng đệm	0.1282	0.1135	0.3957
Đai ốc	0.0579	0.1549	0.3504

Từ bảng này cho thấy một cách chọn khá đơn giản: Đặc trưng tốt nhất để phân biệt đai ốc là diện tích, phân biệt vòng đệm là chu vi, còn để phân biệt đinh ốc là hình dạng tròn bởi vì các giá trị V của những lớp này trong các đặc trưng tương ứng khác biệt so với những lớp còn lại.

Một con số khác cũng hữu ích để xác định khoảng cách kỳ vọng của hai lớp là

$$V_{ij}=\frac{\left|\overline{x_i}-\overline{x_j}\right|}{\sqrt{\sigma_i^2+\sigma_j^2}}$$

Với ví dụ ảnh trên ta có bảng dữ liệu so sánh các V_{ij}:

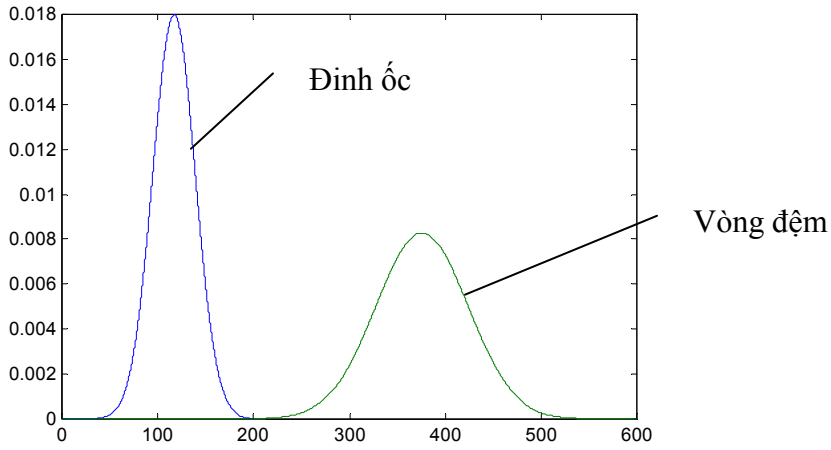
	Đặc trưng diện tích			Đặc trưng hình dáng tròn		
Lớp	Đinh ốc	Vòng đệm	Đai ốc	Đinh ốc	Vòng đệm	Đai ốc
Đinh ốc	0.00	4.88	4.16	0.00	0.75	0.52
Vòng đệm	4.88	0.00	3.05	0.75	0.00	1.01
Đai ốc	4.16	3.05	0.00	0.52	1.01	0.00

Nếu đặc trưng nào có khoảng cách kỳ vọng của hai lớp lớn thì đặc trưng đó có thể phân biệt được hai đối tượng. Ngược lại nếu khoảng cách kỳ vọng nhỏ thì đặc trưng đó không tốt cho việc phân biệt giữa hai lớp đó. Với ví dụ đưa ra ở trên, thì rõ ràng diện tích thì tốt cho việc phân biệt đinh ốc và vòng đệm (vì khoảng cách khá lớn : 4.88). Ngược lại, đặc trưng hình dáng tròn không tốt cho việc phân biệt đinh ốc và vòng đệm (vì khoảng cách đó khá nhỏ : 0.75).

Tiếp theo, chúng ta áp dụng một cách khác để có thể chọn ra các đặc trưng có thể phân biệt tốt các lớp đối tượng một cách tổng quát hơn. Chúng ta hoàn toàn có thể giả sử rằng các giá trị của ba đặc trưng diện tích, chu vi, hình dáng tròn cho ảnh trên trên có phân phối chuẩn $N(\bar{x},\sigma^2)$ (normally distributed), và hàm mật độ xác suất là:

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$

Phương trình đường cong này có dạng hình chuông cho mỗi đặc trưng x.

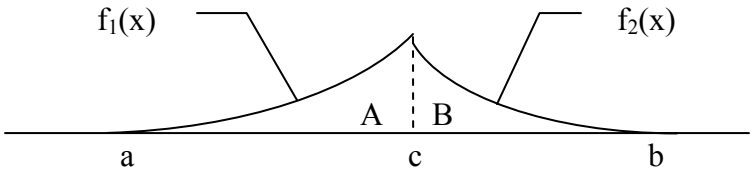


Hình 1.1.b **:** *Đường cong hình chuông của đỉnh ốc và vòng đệm với đặc trưng diện tích*

Một đặc trưng tốt nếu hàm mật độ xác suất của nó có dạng đường cong hình chuông cao và hẹp, vì điều này đồng nghĩa với việc có phương sai nhỏ. Một đặc trưng mà có thể phân biệt các lớp với nhau khi các đường cong hình chuông của các lớp không bị chồng chất lên nhau hoặc ít chồng chất. Ví dụ trong hình 1.1.b trên, đặc trưng diện tích của lớp đỉnh ốc và lớp vòng đệm có thể phân biệt được các đối tượng thuộc hai lớp này vì chúng có đường cong hình chuông không bị chồng chất. Ví dụ nếu một đối tượng có diện tích 121 đơn vị diện tích thì đối tượng đó sẽ là đỉnh ốc vì diện tích của nó quá nhỏ so với vòng đệm. Còn nếu xét đặc trưng hình dáng tròn cho lớp đỉnh ốc và vòng đệm thì rõ ràng đặc trưng này không tốt để phân biệt hai lớp đối tượng này vì hai đường cong hình chuông của hai lớp này chồng lên nhau khá nhiều từ 1.24 $\rightarrow$ 3 đơn vị giá trị đặc trưng hình dáng tròn.

Vì vậy để xác định được đặc trưng nào có thể phân biệt các lớp, đơn giản là chúng ta vẽ các cặp đường cong cho các cặp lớp đối tượng cần phân biệt rồi xem sự chồng chất của các cặp hình đó như thế nào. Nhưng việc so sánh này chỉ thực hiện cho từng đặc trưng của từng cặp lớp đối tượng chứ không nên thực hiện tất cả các đặc trưng cho tất cả các lớp.

Nhưng làm cách nào có thể biết được các đường cong hình chuông có chồng chất lên nhau như thế nào ? Đường cong hình chuông này có phương trình khá phức tạp. Vì vậy ta không dễ dàng xét sự chồng chất của các đường cong này. Mặt khác, nếu chồng chất thì nhiều hay ít . Vì nếu ít thì đặc trưng đang xét vẫn thích hợp để phân biệt hai lớp này. Cho nên ta cần tính diện tích trùng lắp của hai diện tích tạo bởi hai đường cong hình chuông với trục hoành. Điều này khá phức tạp nếu ta áp dụng phương pháp bình thường, vì vậy ta cần áp dụng tính chất tính diện tích tương tự trong tích phân. Nghĩa là diện tích từ một đường cong đến trục hoành xấp xỉ tổng diện tích các hình chữ nhật mà chiều rộng là một khoảng khá nhỏ d, chiều cao là giá trị f(x+id) như hình sau mô tả :



Hình 1.1.c **:** *Minh họa cách tìm diện tích*

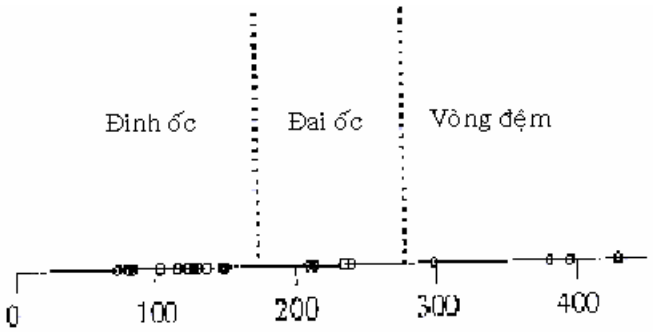
Áp dụng cách tính diện tích này để tìm diện tích trùng lắp của hai đường cong hình chuông của hai lớp như sau: Diện tích trùng lắp chính là tổng của hai diện tích A và B. Trong đó A là diện tích của đường cong $f_1(x)$ ($f_1(x)$ là đường cong thuộc lớp 2) đến trục hoành từ a $\rightarrow$ c (với a là giá trị đặc trưng nhỏ nhất của lớp thứ 2, c là giá trị đặc trưng mà hai đường cong giao nhau) và B là diện tích của đường cong $f_2(x)$ ($f_2(x)$ là đường cong của lớp 1)đến trục hoành từ c $\rightarrow$ b (với b là giá trị đặc trưng lớn nhất của lớp thứ 1). Áp dụng cách này cho ta có bảng dữ liệu diện tích trùng lắp của ảnh các con ốc nêu trên khi xét về đặc trưng diện tích:

Diện tích trùng lắp			
Lớp	Đinh ốc	Vòng đệm	Đai ốc
Đinh ốc	1.0	0.0002	0.0023
Vòng đệm	0.0002	1.0	0.011
Đai ốc	0.0023	0.011	1.0

Nhìn vào bảng này ta thấy rõ là đinh ốc và vòng đệm có diện tích trùng lắp khá nhỏ 0.0002, điều này suy ra đặc trưng diện tích thích hợp để phân biệt đinh ốc và vòng đệm.

1.2 Phân tích mẫu thống kê

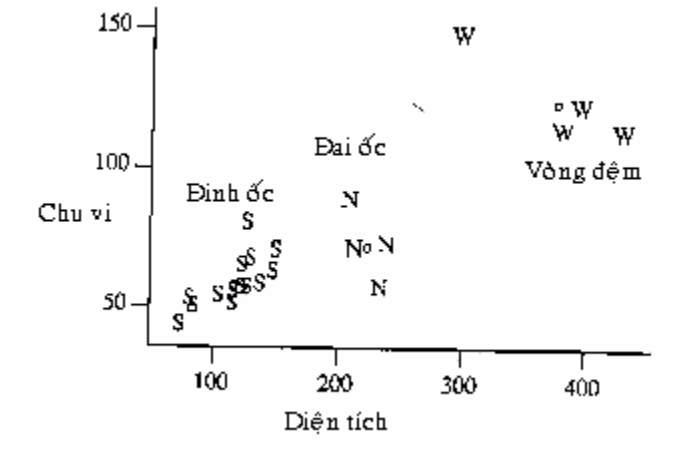
Như phần trên đã đề cập, tìm đặc trưng để phân biệt các lớp đối tượng là bước đầu tiên và quan trọng nhất để phân lớp một tập các đối tượng. Khảo sát lại bức ảnh các con ốc ở trên, chúng ta có thể chỉ sử dụng diện tích làm đặc trưng là đủ để phân biệt các đối tượng.



Hình 1.2.a : *Minh họa các giá trị diện tích của các đối tượng trong ảnh các con ốc trên một trục*

Giá trị diện tích của các đối tượng trong ảnh được thể hiện trên một trục. Nếu diện tích thuộc đoạn 0 đến 175 thì đối tượng là đinh ốc và nếu diện tích thuộc đoạn 176 đến 275 thì đối tượng là đai ốc, và nếu diện tích lớn hơn 275 thì đối tượng là vòng đệm. Cách phân biệt dùng một đặc trưng giống như ta đang xét trên không gian một chiều.

Nếu ta sử dụng nhiều đặc trưng hơn để phân biệt thì có lẽ các đối tượng sẽ được phân biệt một cách chắc chắn hơn, chính xác hơn. Nhưng bằng cách nào ta có thể lưu trữ và thao tác với nhiều đặc trưng? Lúc này ta sử dụng vectơ đặc trưng (là vectơ mà mỗi thành phần tọa độ là một đặc trưng) để thể hiện được nhiều đặc trưng khác nhau cho một đối tượng trong không gian nhiều chiều (gọi là không gian đặc trưng). Số chiều của không gian này là số đặc trưng của vector đặc trưng. Ví dụ nếu ta dùng cả diện tích và chu vi để phân lớp trong ảnh các con ốc trên thì việc này giống như ta đang thao tác trong không gian hai chiều mà mỗi vectơ gồm hai thành phần ứng với diện tích và chu vi. Chẳng hạn ta có: (148,62.2) là vectơ thuộc lớp đinh ốc, (380,113.4) là vectơ thuộc lớp vòng đệm. Hình dưới đây thể hiện các đối tượng trong ảnh trên trong không gian hai chiều (diện tích, chu vi):



Hình 1.2.b **:** *Minh họa các vị trí của đối tượng trong ảnh các con ốc trong không gian hai chiều (Diện tích , Chu vi)*

Hình này cho thấy rõ các đối tượng cùng một lớp sẽ tập trung thành một nhóm gần nhau. Từ điều này, ta đưa ra một phương pháp để nhận diện và phân lớp là phương pháp chọn lớp gần nhất (nearest neighor). Nghĩa là nếu hai vectơ đặc trưng của hai đối tượng có khoảng cách khá nhỏ trong không gian đặc trưng thì có thể hai đối tượng này thuộc cùng một lớp. Khoảng cách giữa hai vector đặc trưng sẽ được định nghĩa là khoảng cách Euclidian, tức là:

$$d = \|x - y\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

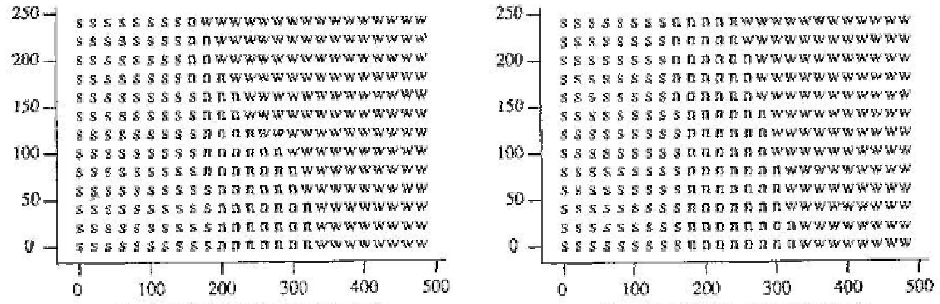
Với : $x=(x_1,...,x_n)$, $y=(y_1,...,y_n)$ là hai vectơ đặc trưng của hai lớp

n là số chiều của không gian đặc trưng hay số đặc trưng được khảo sát

Vậy nếu xét một đối tượng chưa biết thuộc lớp nào, ta cần phân lớp cho đối tượng đó thì ta chỉ cần tìm khoảng cách giữa vectơ đặc trưng của đối tượng chưa biết này và tất cả các

vectơ đặc trưng của đối tượng đã được phân lớp. Đối tượng chưa biết sẽ thuộc lớp mà có đối tượng có khoảng cách nhỏ nhất trong tất cả các khoảng cách vừa tìm. Phương pháp này tính toán khá nhiều. Vì vậy có một phương pháp khác để giảm sự tính toán, đó là phương pháp trọng lượng trung bình gần nhất (nearest centroid). Vectơ đặc trưng trung bình của một lớp là một vectơ đặc trưng có thành phần là các giá trị trung bình của một đặc trưng trong cùng lớp, kí hiệu là $\vec{c} = (\overline{x_1},.....,\overline{x_n})$.

Phương pháp trọng lượng trung bình gần nhất là tính toán các khoảng cách giữa vectơ đặc trưng của đối tượng không biết và vectơ đặc trưng trung bình của các lớp đã biết. Nếu khoảng cách nào nhỏ nhất thì đối tượng chưa biết sẽ thuộc lớp tương ứng. Phương pháp này tính toán ít hơn và nhanh hơn so với phương pháp trên vì chỉ cần tính một vài giá trị khoảng cách. Hình vẽ mô tả vị trí các đối tượng thuộc lớp nào trong không gian đặc trưng hai chiều diện tích và chu vi của hai phương pháp trên đối với ảnh các con ốc:



Hình 1.2.c *: Minh họa vị trí các đối tượng thuộc lớp nào trong không gian đặc trưng hai chiều (Diện tích , Chu vi). Với s : Đinh ốc , n : Đai ốc , w : Vòng đệm (a)Phương pháp chọn lớp gần nhất. (b)Phương pháp trọng lượng trung bình gần nhất*

Hai phương pháp trên thường được sử dụng khi biết được các lớp trong ảnh và chỉ để nhận diện các đối tượng chưa biết. Nếu số lớp trong ảnh chưa biết thì làm cách nào phân lớp các đối tượng? Một cách xử lý được đưa ra gọi là tự động phân lớp (autoclustering). Cách này dựa trên việc sử dụng một ngưỡng (threshold) cho các giá trị khoảng cách của các vectơ đặc trưng. Khi một vectơ đặc trưng của một đối tượng chưa biết được xét để phân lớp thì ta cần tính khoảng cách giữa vectơ đặc trưng mới này và các vectơ đặc trưng của tất cả các đối tượng đã được phân lớp. Lúc này, nếu khoảng cách d nhỏ hơn giá trị ngưỡng d_t của lớp đã tồn tại thì đối tượng mới sẽ thuộc lớp đó. Ngược lại nếu khoảng cách d lớn hơn d_t thì đối tượng chưa biết trở thành đối tượng đầu tiên của lớp mới. Giá trị ngưỡng d_t thì ảnh hưởng đến kết quả phân lớp. Nếu d_t nhỏ thì sẽ cho ra nhiều lớp hơn số lớp thực tế. Nếu d_t lớn thì cho ít lớp hơn số lớp thực tế. Vì vậy việc xác định d_t khá quan trọng, để xác định được d_t ta cần hỗ trợ cho máy tính bằng cách "huấn luyện " cho máy tính có thể tự đưa ra giá trị d_t phù hợp cho các lớp quen thuộc.Tóm lại, với phương pháp này ta có thể áp dụng để có được các

đối tượng đầu tiên của các lớp. Từ đó ta có thể nhận diện và phân lớp được đối tượng trong ảnh.

1.3 Phương pháp xác suất

Một câu hỏi đặt ra là việc phân lớp luôn đúng hay không? Tất nhiên, trong quá trình phân lớp có thể sẽ bị nhầm lẫn. Mục tiêu của chúng ta là phải làm sao để lỗi này giảm xuống nhỏ nhất hay khả năng đặt đúng lớp là lớn nhất. Để làm được điều này ta có thể sử dụng lý thuyết xác suất.

Xác suất của một biến cố A là khả năng thường xảy ra của biến cố A trong không gian mẫu được thể hiện bằng một con số từ 0 đến 1. Kí hiệu p(A). Nếu biến cố không bao giờ xảy ra thì có xác suất là 0; nếu biến cố đó chắc chắn xảy ra thì có xác suất là 1; còn nếu xác suất trong khoảng (0,1) là khả năng có thể xảy ra của biến cố. Ví dụ xác suất lấy một con tây trong một bộ bài 52 lá là $\frac{12}{52} = \frac{4}{13}$. Xét bức ảnh các con ốc của phần trước, có một vài lớp xuất hiện trong ảnh nhiều hơn các lớp khác, chẳng hạn đinh ốc có nhiều hơn các loại khác. Điều này nghĩa là xác suất xảy ra của lớp đinh ốc lớn hơn của các lớp khác. Xác suất của một đối tượng có thuộc lớp C_i là $p(C_i)$. Ví dụ nếu ảnh có 1000 đối tượng, trong đó có 200 đối tượng thuộc lớp C_i thì xác suất của một đối tượng thuôc vào lớp C_i là $\frac{200}{1000} = 0.2$. Tổng của xác suất của tất cả các lớp trong ảnh là 1.

Cũng như phần phần 1.1, chúng ta xét một đặc trưng để phân lớp. Nhưng ở đây chúng ta sử dụng xác suất để lựa chọn lớp cho đối tượng. Xác suất của đối tượng có đặc trưng x đối với lớp C_i là $p\left(x/C_i\right)$. $p\left(x/C_i\right)$ này thể hiện mức độ tốt hay không tốt khi sử dụng đặc trưng x để phân biệt cho lớp C_i. Vì vậy nó chính là giá trị của hàm mật độ xác suất f(x) đối với lớp C_i .

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}.e^{\frac{-(x-\bar{x})^2}{2\sigma^2}}$$

Để thực hiện việc phân lớp dựa trên ý tưởng xác suất xảy ra lớn nhất, ta cần tính $p\left(C_i/x\right)$ là xác suất để một đối tượng thuộc C_i khi xét đặc trưng x. Sau đó đặt đối tượng vào lớp có xác suất $p\left(C_i/x\right)$ lớn nhất. Nhưng giá trị $p\left(C_i/x\right)$ thì không dễ dàng tính toán từ các dữ liệu cho trước. Vì thế ta cần dùng công thức Bayes để tính :

$$p\left(C_i/x\right) = \frac{p\left(x/C_i\right).p(C_i)}{p(x)}$$

Bởi vì $p\left(x/C_i\right)$ và $p(C_i)$ thì dễ dàng tính từ dữ liệu cho trước hơn $p\left(C_i/x\right)$, còn giá trị p(x) thì không quan trọng vì nó sẽ không cần để so sánh các $p\left(C_i/x\right)$ với nhau. Cách phân lớp như vậy phân lớp có khả năng lớn nhất (hay gọi là phân lớp Bayes).

Ví dụ quan sát lại bức ảnh các con ốc

Gọi: C_1 là lớp đinh ốc.

C_2 là lớp vòng đệm.

Ta cần phân biệt đối tượng chưa biết có diện tích 148 thuộc lớp C_1 hay C_2. Để làm được điều này ta cần so sánh $p\left(C_1/148\right)$ và $p\left(C_2/148\right)$. Nếu giá trị nào lớn hơn thì đối tượng chưa biết sẽ thuộc lớp tương ứng. Ta có:

$p(C_1) = 0.85$, $p(C_2) = 0.15$

$p\left(148/C_1\right) = 0.0068$, $p\left(148/C_2\right) < 0.000001$

Suy ra $p\left(C_1/148\right) = \frac{(0.0068).(0.85)}{p(x)} = \frac{0.00578}{p(x)}$

$p\left(C_2/148\right) < \frac{(0.000001).(0.15)}{p(x0} = \frac{0.00000015}{p(x)}$

$\Rightarrow p\left(C_1/148\right) > p\left(C_2/148\right)$

Vậy đối tượng có diện tích 148 thuộc lớp C_1.

Nếu chúng ta chỉ phân lớp cho trường hợp đơn giản là cho hai lớp và chỉ dùng một đặc trưng thì ta có thể làm đơn giản hơn.

Vì chỉ có hai lớp nên ta có $p(C_2) = 1 - p(C_1)$. Giả sử rằng :

$$p\left(C_1/x\right) = p\left(C_2/x\right)$$

$$\Leftrightarrow \quad \frac{p\left(x/C_1\right).p(C_1)}{p(x)} = \frac{p\left(x/C_2\right).p(C_2)}{p(x)}$$

$$\Leftrightarrow \quad \frac{p\left(x/C_2\right)}{p\left(x/C_1\right)} = \frac{p(C_1)}{p(C_2)}$$

$$\Leftrightarrow \quad \frac{p\left(x/C_2\right)}{p\left(x/C_1\right)} = \frac{p(C_1)}{1 - p(C_1)}$$

Đặt $\frac{p\left(x/C_2\right)}{p\left(x/C_1\right)} = \Lambda(x), \frac{p(C_1)}{1-p(C_1)} = T$

Ta thấy rằng T chính là ngưỡng để phân biệt giữa hai lớp C_1,C_2 trong phương pháp phân lớp Bayes. Cụ thể là nếu $\Lambda(x) \le T$ thì đối tượng có đặc trưng x thuộc lớp C_1, ngược lại $\Lambda(x) > T$ thì đối tượng có đặc trưng x thuộc lớp C_2. Một cách tổng quát ta định nghĩa hàm k(x) để thể hiện phương pháp này như sau:

$$k(x) = \begin{cases} C_1 & \Lambda(x) \le T \\ C_2 & \Lambda(x) > T \end{cases}$$

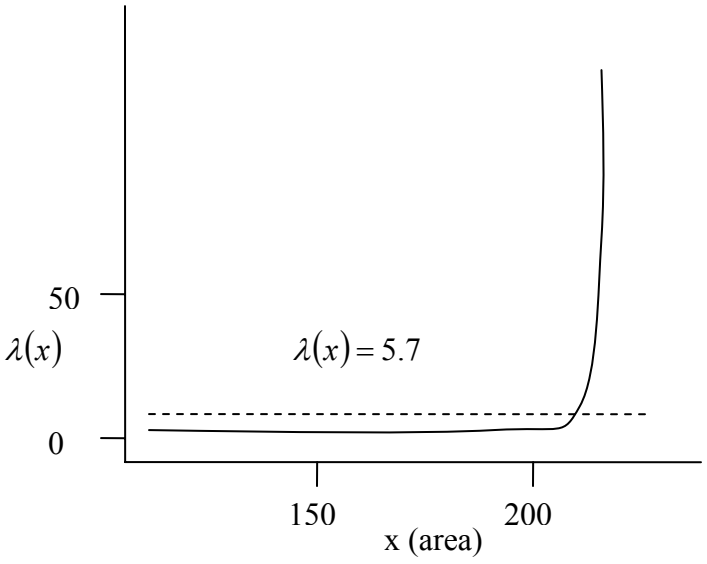
Xét lại ví dụ trên, ta có:

$$\Lambda(148) = \frac{0.000001}{0.0068} = 0.000147$$

$$T = \frac{0.85}{0.15} = 5.7$$

$$\Rightarrow \Lambda(148) < T$$

Vậy đối tượng thuộc lớp C_1



Hình 1.3.a **:** *Minh họa cách phân biệt hai lớp đinh ốc và vòng đệm trong phương pháp phân lớp Bayes*

Vì đặc trưng diện tích thì tốt cho việc phân lớp giữa đinh ốc và vòng đệm nên đồ thị (Λ, x) sẽ phân biệt hai lớp một cách rõ ràng. Còn nếu ta xét trên đặc trưng hình dáng tròn thì sẽ không phân biệt hai lớp rõ ràng. Trong phương pháp này, có thể xảy ra hai kiểu lỗi:

+ Lỗi 1: đưa đối tượng vào lớp 2 trong khi nó thuộc lớp 1.

+ Lỗi 2: đưa đối tượng vào lớp 1 trong khi nó thuộc lớp 2.

Gọi ε_1 là xác suất xảy ra lỗi 1.

ε_2 là xác suất xảy ra lỗi 2

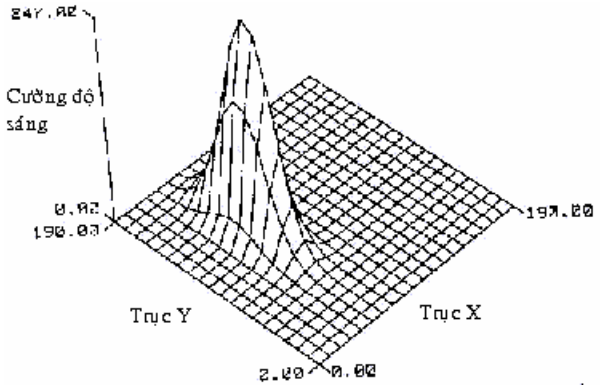
Khi đó để có độ chính xác hơn khi xảy ra lỗi, giá trị T cần được tính như sau:

$$T = \frac{\varepsilon_1 . p(C_1)}{\varepsilon_2 . (1 - p(C_1))}$$

Mở rộng phương pháp Bayes, ta có thể dùng nhiều đặc trưng hơn để phân lớp tương tự với phương pháp thống kê.Ví dụ với hai đặc trưng thì hàm mật độ xác suất được tính bằng công thức sau:

$$f(x,y) = \frac{1}{\sigma_x . \sqrt{2\pi}} . e^{\frac{-(x-\bar{x})^2}{2\sigma_x^2}} . \frac{1}{\sigma_y . \sqrt{2\pi}} . e^{\frac{-(x-\bar{x})^2}{2\sigma_y^2}}$$

Hàm này có hình dạng là một mặt cong hình chuông trong không gian ba chiều như hình sau:

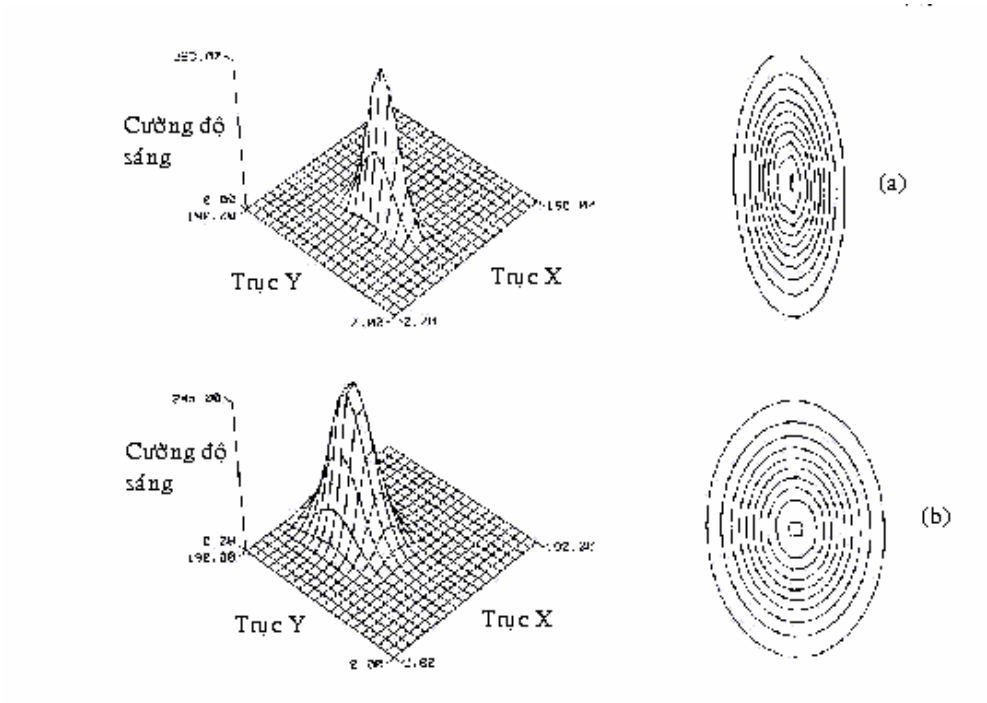


Hình 1.3.b **:** *Minh họa hình dạng của hàm mật độ xác xuất theo đặc trưng diện tích và chu vi*

Chúng ta có thể tính toán tương tự bằng công thức Bayes nhưng tính toán này trong không gian nhiều chiều để thực hiện thì không tốt. Một cách khác để phân lớp trong trường hợp này được thảo luận dưới đây. Mỗi một lớp được thể hiện bởi một mặt cong hình chuông (tức hàm mật độ xác suất của nó). Cũng như phương pháp lát cắt được dùng trong việc vẽ bản đồ, người ta thể hiện các mặt cong hình chuông thành những đường tròn đồng tâm.

Ví dụ phân lớp cho hai lớp A và B có các dữ liệu trong bảng sau:

	Đặc trưng 1(x)		Đặc trưng 2(y)	
Lớp	Kỳ vọng	Độ lệch tiêu chuẩn	Kỳ vọng	Độ lệch tiêu chuẩn
A	178.9	38.2	158.25	15.3
B	243.7	45.95	171.89	31.14



Hình 1.3.c **:** *Minh họa việc thể hiện các mặt cong hình chuông thành những đường tròn đồng tâm*
(a)Mặt cong hình chuông đường tròn đồng tâm của lớp A
(b)Mặt cong hình chuông đường tròn đồng tâm của lớp B

Người ta định nghĩa hàm d(x,y) bằng hiệu của hai hàm mật độ xác suất của hai lớp:

$$d(x,y) = f_2(x,y) - f_1(x,y)$$

Phương trình đường cong d(x,y)=0 có nghĩa như đường cong đi qua các giao điểm của hai hình lát cắt của hai mặt cong hình chuông. Lúc này rõ ràng đường cong như một đường biên giới chia ra hai lớp, nếu d(x,y)<0 thì đối tượng đang xét sẽ thuộc lớp 1; ngược lại sẽ thuộc lớp 2.

Một cách tổng quát ta định nghĩa hàm k cho phương pháp này như sau:

$$k(x,y) = \begin{cases} C_1 & d(x,y) \le 0 \\ C_2 & d(x,y) > 0 \end{cases}$$

$d(x,y) = f_B(x,y)-f_A(x,y)$
$= 0.0002.x^2-0.0144x+0.0032y^2-0.998y+72.11$
Xét một đối tượng có hai đặc trưng (x,y)=(181.2,160.7).
Lúc đó d(181.2,160.7)=-0.159<0
Suy ra đối tượng này thuộc lớp A.

1.4 Phương pháp so mẫu đối tượng (Template Matching)

1.4.1 Đối tượng mẫu

- Một đối tượng mẫu mang các đặc trưng trung bình của một lớp đối tượng, và được sử dụng để nhận biết các đối tượng cùng thuộc lớp đó.
- Các đối tượng mẫu thường được sử dụng để nhận dạng chữ in, số, và các đối tượng đơn giản, nhỏ khác.
- Phương pháp so mẫu thường được sử dụng cho ảnh Bi-level.

1.4.2 So mẫu trên ảnh Bi-level

1.4.2.1 Độ liên kết chuẩn(Normalized Match Index)

- Đếm số pixel đối tượng tương ứng giữa ảnh mẫu và vùng ảnh đang được xem xét(có cùng giá trị cường độ là 0). Ký hiệu: N_1.
- Đếm số pixel đối tượng không tương ứng giữa ảnh mẫu và vùng ảnh đang được xem xét(không cùng giá trị cường độ). Ký hiệu: N_2.
- Độ liên kết chuẩn được tính theo công thức:

$$N = \frac{N_1 - N_2}{N_1 + N_2}$$

- Ví dụ minh hoạ:

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	. . . .
1 1 0 0 0 0 1 1	1 1 1 1 1 1 1 1	. . - - - - - . .
1 0 1 1 1 1 0 1	1 1 0 1 1 1 0 1	. - - . . . + .
1 0 0 0 0 0 0 1	1 1 1 0 1 0 1 1	. - - + - + - .
1 0 1 1 1 1 1 1	1 1 1 1 0 1 1 1	. - . . - . . .
1 0 1 1 1 1 1 1	1 1 1 0 1 0 1 1	. - . - . - - .
1 1 0 0 0 0 1 1	1 1 0 1 1 1 0 1	. . + - - - - .
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	. . . .
Ảnh mẫu	Vùng ảnh dữ liệu	Độ liên kết: (4-20)/(4+20)

Hình 1.4.2.1: *Ví dụ xác định độ liên kết chuẩn*

1.4.2.2 Phương pháp

- Di chuyển ảnh mẫu đến tất cả các vị trí có thể trong ảnh lớn.
- Tại mỗi vị trí độ liên kết chuẩn được xác định để quyết định vùng đang xét có thuộc cùng lớp với đối tượng mẫu không.

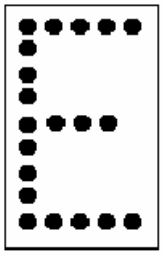
- Phương pháp này đối chiếu từng pixel. Do đó những đối tượng khác nhau về kích cỡ, phương hướng sẽ xem như thuộc lớp khác nhau.

1.5 Phương pháp nhận dạng dựa vào cấu trúc (Structural method)

- Phương pháp này dựa trên ý tưởng là: các đối tượng được kết cấu từ các thành phần nhỏ và đơn giản hơn theo các luật nhất định. Quá trình nhận ra một đối tượng trong ảnh chính là xác định các thành phần, thiết lập mối quan hệ giữa các thành phần, sau đó so với các mẫu đã được biết trước.
- Phương pháp này sử dụng các thuật ngữ tương đối để mô tả quan hệ giữa các thành phần của đối tượng (dài hơn, ngắn hơn,…).

1.5.1 Một ví dụ cụ thể

- Các thành phần tạo nên chữ E gồm một đoạn thẳng đứng và ba đoạn thẳng nằm ngang.



***Hình 1.5.1**: Ký tự E*

- Mối liên hệ giữa các thành phần tạo nên ký tự E có thể được mô tả như sau:Tại mỗi đầu đoạn thẳng đứng gặp 2 đoạn thẳng nằm ngang ngắn hơn theo cùng một hướng; đoạn thẳng nằm ngang thứ 3 gặp trung điểm đoạn thẳng đứng theo cùng hướng 2 đoạn thẳng nằm ngang trước.

1.5.2 Mô tả các thành phần cơ bản và các quan hệ

- Các thành phần cơ bản tạo nên đối tượng bao gồm các đoạn thẳng và các đường cong nhỏ.

1.5.2.1 Mô tả các thành phần cơ bản và quan hệ giữa chúng theo dạng đồ thị.

1.5.2.1.1 Ý tưởng

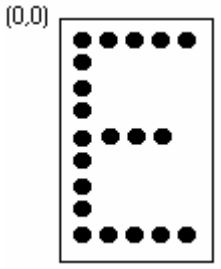
- Trong nhận dạng mẫu theo cấu trúc, cấu trúc dữ liệu được sử dụng là đồ thị (Graph). Các nút đại diện cho các thành phần và các cạnh đại diện cho mối quan hệ giữa chúng.
- Đối tượng mẫu được mô tả trước dưới dạng đồ thị mẫu. Để nhận biết đối tượng trong ảnh dữ liệu, trước hết các thành phần được định vị cùng với mối liên hệ của chúng để thiết lập một dạng đồ thị, sau đó so đồ thị này với đồ thị mẫu.

1.5.2.1.2 Ví dụ

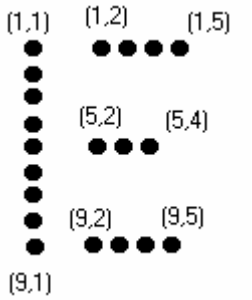
- Xét lại ví dụ ký tự E, các thành phần đoạn thẳng được phân lớp theo chiều dài và phương hướng của chúng.

- Sử dụng các thuật ngữ tương đối để mô tả các thành phần. Chiều dài có thể là dài (LONG- nếu lớn hơn ½ kích cỡ đối tượng), ngắn (SHORT- nếu nhỏ hơn hoặc bằng ½ kích cỡ đối tượng). Các hướng có thể là thẳng đứng (VERTICAL - gần 90^0), nằm ngang (HORIZONTAL- gần 0^0).

- Giả sử ký tự E có toạ độ như sau:



- Các đoạn thẳng cấu thành ký tự E:



- Đoạn thẳng đầu tiên từ (1,1) đến (9,1) được phân lớp là thẳng đứng (VERTICAL - gần 0^0). Để quyết định sự phân lớp về chiều dài, trước hết xác định chiều dài của hình chữ nhật bao đóng nhỏ nhất là 10-0 +1=11, suy ra đoạn thẳng đầu tiên có tính chất dài (LONG – do 9 > 11/2).

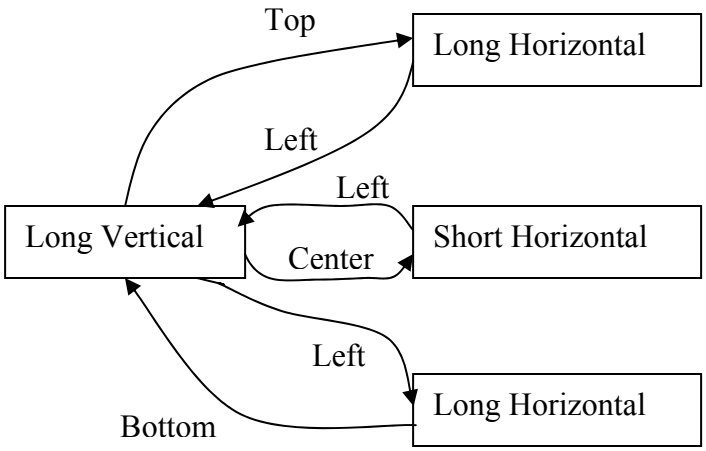
- Xác định tương tự cho các đoạn thẳng còn lại, ta được kết quả sau:

Đoạn thẳng	Chiều dài	Hướng
(1,1)-(9,1)	LONG	VERTICAL
(1,2)-(1,5)	LONG	HORIZONTAL
(5,2)-(5,4)	SHORT	HORIZONTAL
(9,2)-(9,5)	LONG	HORIZONTAL

- Mỗi thành phần phần bây giờ được xem như một nút (node) trong đồ thị.

- Bước kế tiếp là chỉ ra mối quan hệ giữa các thành phần, và mã hoá chúng như các cạnh. Các quan hệ trong trường hợp này được định nghĩa là vị trí tương đối nơi các đường thẳng gặp nhau: TOP, BOTTOM, LEFT, RIGHT, CENTER, mối quan hệ này không đối xứng.

- Lược đồ mô tả các quan hệ:



Hình 1.5.2.1.2: *Lược đồ mô tả mối quan hệ giữa các thành phần*

- Các bước thiết lập đồ thị:
  + Mỗi thành phần được cấp phát như một nút.
  + Phân lớp kiểu cho mỗi nút (Vd: SHORT VERTICAL) và lưu kết quả cho mỗi nút.
  + Đối với mỗi nút A, xác định các nút liên kết với A(ký hiệu: B), phân lớp sự liên kết theo mỗi hướng (từ A đến B, từ B đến A).
- Bước cuối cùng là xác định xem đồ thị vừa thiết lập với đồ thị mẫu có đồng dạng hay không. Một nút A_i trong đồ thị A tương ứng với nút B_i trong đồ thị B nếu chúng có cùng thuộc tính và liên kết những nút các nút khác theo cùng quan hệ.

1.5.2.2 Mô tả các quan hệ theo cú pháp (Syntatic)

1.5.2.1.1 Định nghĩa văn phạm để mô tả đối tượng

Có thể sử dụng ngôn ngữ hình thức để mô tả các thành phần cơ bản của đối tượng và mối quan hệ giữa chúng.

Văn phạm (grammar) là bộ bốn G =(N, $\sum$, P, S) với:

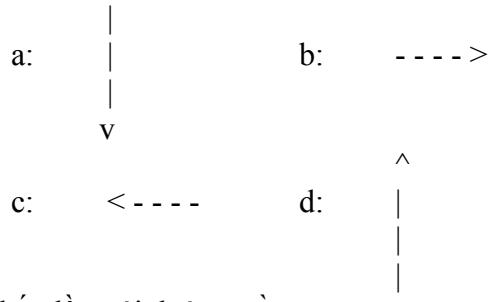
- N: là tập hữu hạn các ký hiệu không cuối (nonterminal symbol - được ký hiệu bằng chữ Latinh hoa).
- T: là tập hữu hạn các ký hiệu cuối (nonterminal symbol- được ký hiệu bằng chữ Latinh thường) và $N \cap T = \phi$.
- S : là ký hiệu đầu tiên, và $S \in N$.
- $\sum$: là tập các qui tắc, là tập con của $(N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$ (các qui tắc được viết dưới dạng $\alpha \to \beta$)

Từ một ký tự bắt đầu, theo các qui tắc qui tắc sinh, ta sẽ có được một chuỗi bao gồm toàn ký hiệu cuối, kết quả đó gọi là ngôn ngữ được sinh bởi văn phạm tương ứng.

1.5.2.1.2 Ví dụ

Xét ví dụ thiết kế văn phạm cho số 3:

- Định nghĩa các ký hiệu cuối:



- Số 3 bắt đầu với đường nằm ngang:

S → b
|
B

- Tiếp theo là đường thẳng dọc:

B → a
|
C

- Kế đến là đường nằm ngang, và phần còn lại bên dưới:

C → .
/ \
c D

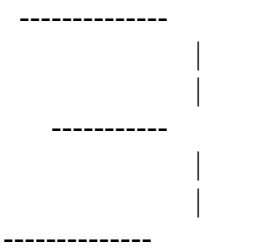
- Bên dưới đường nằm ngang là đường thẳng đứng:

D → a
|
E

- Cuối cùng,

E → c

Văn phạm này sinh ra các ký tự có dạng sau:

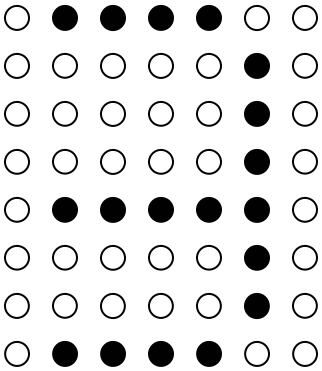


1.5.3 Nhận biết các thành phần (Identifying Components)

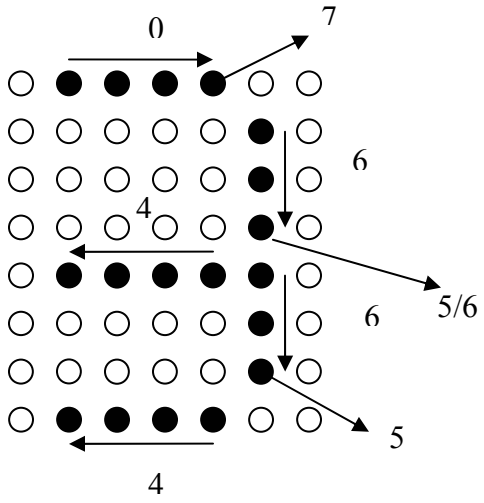
1.5.3.1 Dựa vào mã theo dạng xích(Chain Code)

1.5.3.1.1 Nhận biết đường thẳng

- Ý tưởng: những pixel kề nhau nằm trên cùng một đường thẳng sẽ có hướng (mã theo dạng xích) giống nhau.
- Ví dụ đối tượng sau:



- Mã hoá theo dạng xích, ta được:



- Những pixel kề nhau có cùng mã (trừ pixel cuối cùng) được xem như nằm trên cùng một đường thẳng. Giá trị mã đó sẽ phân lớp kiểu đường thẳng: nếu mã có giá trị 0 - đại diện cho đường thẳng nằm ngang từ trái qua phải,…Kết quả này được sử dụng cho việc nhận dạng theo cú pháp (chẳng hạn giá trị 0 tương ứng ký hiệu b ở ví dụ trên).

- Tuy nhiên phương pháp này chỉ có thể thể hiện được 8 dạng (hướng) đường thẳng, đồng thời rất nhạy cảm với sự xuất hiện của một vài pixel được đặt sai vị trí.

1.5.3.1.2 Nhận biết đường cong

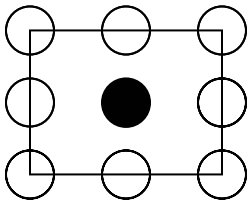
- Ý tưởng: một đường cong bao gồm các pixel kề nhau có hướng (mã theo dạng xích) tăng hoặc giảm liên tục.

- Ví dụ, dãy mã sau: 00111223333443222211000 bao gồm hai đường cong được đại diện bởi 2 dãy: 0011122333344 và 322211000.

1.5.3.2 Dựa vào tính chất dây cung (Chord Property)

1.5.3.2.1 Khái niệm hình vuông lân cận 8

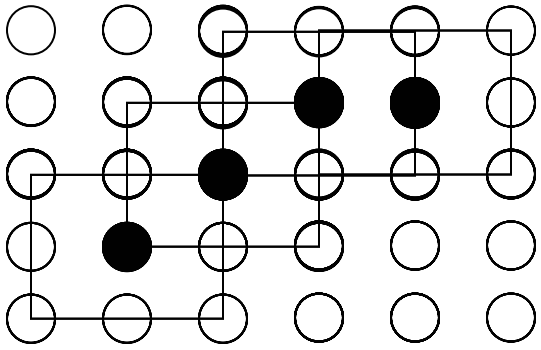
Hình vuông lân cận 8 của pixel P chính là hình vuông được tạo ra từ 8 lân cận của P.



Hình 1.5.3.2.1: *Hình vuông lân cận 8*

1.5.3.2.2 Khái niệm dãy hình lân cận 8

Cho dãy các pixel $P_0[i_0,j_0]$, $P_1[i_1,j_1]$, $P_2[i_2,j_2]$, … $P[i_nn,j_n]$ thỏa điều kiện pixel P_k $[i_k,j_k]$ l à lân cận 8 của pixel P_{k+1} $[i_{k+1},j_{k+1}]$ với $\forall k : 0 \le k \le n-1$ $P_0, P_1, ..., P_n$. Khi đó, dãy hình lân cận 8 chính là hội các hình vuông lân cận 8 của các pixel P_i với $0 \le i \le n$.



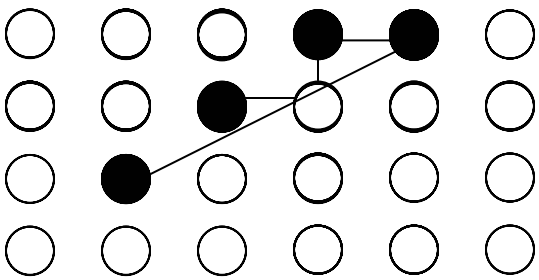
Hình 1.5.3.2.2: *Dãy hình lân cận 8*

1.5.3.2.3 Thao tác nhận biết đoạn thẳng dựa vào tính chất dây cung

- Tính chất dây cung: mọi đường thẳng nối hai điểm bất kỳ phải nằm bên trong dãy hình lân cận 8 của các pixel thuộc đường thẳng đó.
- Thao tác thực hiện: chọn 1 pixel bắt đầu, và thêm 1 lân cận của nó tập hợp. Một pixel sẽ được thêm vào tập hợp nếu thoả tính chất dây cung.
- Tuy nhiên, phương pháp này sẽ mất rất nhiều thời gian với những đoạn thẳng dài do nó kiểm tra tất cả các cặp pixel có thể có trong tập hợp.

1.5.3.2.4 Cải tiến

- Một pixel được thêm vào tập hợp nếu thoả điều kiện khoảng cách theo chiều ngang và theo chiều dọc từ tất cả các pixel trong tập hợp đến đường thẳng tạo bởi nó và điểm đầu phải nhỏ hơn hoặc bằng 1 đơn vị.



Hình 1.5.3.2.4

- Phương pháp này cho phép nhận biết các đường thẳng có hệ số gốc bất kỳ với thời gian thực hiện nhanh hơn.

1.6 Phương pháp bao đóng

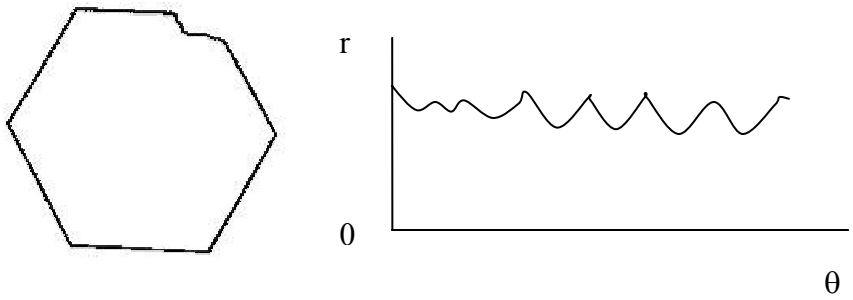
Bao đóng của các đối tượng trong ảnh cũng có thể dùng để nhận dạng và phân lớp các đối tượng trong ảnh. Phương pháp bao đóng là sử dụng bao đóng của các đối tượng trong ảnh để có thể nhận diện chúng. Phương pháp này thuận lợi để làm giảm sự tính toán đối với những ảnh có số pixel của đối tượng quá lớn so với số pixel của bao đóng đối tượng đó. Nhưng phương pháp này lại không thể xử lý được các đối tượng bị chồng lên nhau hay tiếp xúc nhau.

Ý tưởng của phương pháp này cũng gần giống với phương pháp so mẫu đối tượng (temp matching) đã nêu trong phần 1.4. Phương pháp so mẫu đối tượng thì tính toán khá nhiều. Ví dụ với ảnh 256x256 pixel, để nhận dạng một đối tượng có kích thước 30x30 pixel thì ta cần 256x256x30x30x360≈20000 triệu phép so sánh vì phải xét ở tất cả các hướng nên phải so sánh 360 lần tại một vị trí trong ảnh. Trong khi đó phương pháp bao đóng có số phép so sánh ít hơn nhiều.

Phương pháp bao đóng cũng dùng một bao đóng mẫu để nhận dạng các đối tượng có bao đóng giống nó. Nhưng sự so sánh giữa đối tượng mẫu này với các đối tượng trong ảnh không giống phương pháp so mẫu đối tượng. Sự so sánh giữa đối tượng mẫu và đối tượng trong ảnh trong phương pháp bao đóng là sự so sánh giữa các đồ thị (r,θ). Đồ thị (r,θ)được định nghĩa như sau:

$r : [0,360] \to \Re$

$\theta \mapsto r(\theta)$ = khoảng cách từ trọng tâm của đối tượng đến một điểm trên bao đóng ứng với góc θ



Hình 1.6.a: *Ảnh đối tượng và đồ thị (r,θ) tương ứng*

Gọi: B là đồ thị (r,θ) của đối tượng trong ảnh
T là đồ thị (r,θ) của đối tượng mẫu

Để so sánh hai đồ thị B và T, ta chỉ cần tính giá trị D_α sau:

$$D_\alpha = \sum_{\theta=0}^{360} [r_B(\theta) - r_T(\theta+\alpha)]^2$$

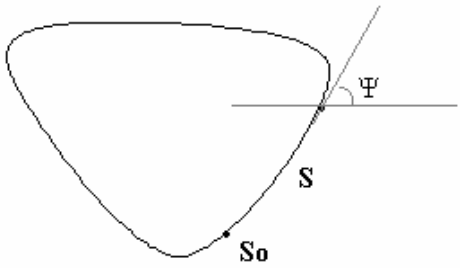
hoặc

$$D_\alpha = \sum_{\theta=0}^{360} |r_B(\theta) - r_T(\theta+\alpha)|$$

Giá trị D_α là giá trị thể hiện giống nhau của B và T ứng với góc $\alpha(\alpha \in [0,360])$. Giá trị α có ý nghĩa như ta xoay đối tượng mẫu trước khi so sánh với đối tượng trong ảnh. Nếu tồn tại một α sao cho D_α khá nhỏ thì hai đồ thị B và T khá giống nhau. Điều này tức là đối tượng đang xét trong ảnh khá giống với đối tượng mẫu. Như vậy ta có thể xếp đối tượng đang xét này cùng lớp với đối tượng mẫu.

Phương pháp bao đóng này chỉ cần so sánh 360x360≈100000 cho ảnh có kích thước bất kỳ. Đồng thời ta có thể giảm giá trị r khi r quá lớn bằng cách thay r bằng $r/r_{\max}$.

Trường hợp gây ra lỗi khi sử dụng phương pháp bao đóng là khi đối tượng trong ảnh bị méo mó, biến dạng. Lúc đó đồ thị (r,θ) thể hiện bao đóng cho đối tượng này sẽ thay đổi vì đường cong r(θ) phụ thuộc vào vị trí chọn trọng tâm ban đầu và bao đóng đối tượng. Để khắc phục lỗi này, chúng ta có một đồ thị khác thay thế cho đồ thị (r,θ) đó là đồ thị (s,ψ). Với s là chiều dài tính từ điểm đang xét đến điểm ban đầu trên bao đóng; còn ψ là hướng của tiếp tuyến so với trục hoành tại điểm có chiều dài s.



Hình 1.6.b: *Minh họa s và* ψ

Cách này thuận lợi hơn cách dùng đồ thị (r,θ) ở chổ nó không cần tìm trọng tâm của đối tượng nên vẫn áp dụng được để so sánh với đối tượng mẫu mặc dù đối tượng đang xét có hơi bị móp méo so với mẫu. Nhưng cách này lại so sánh khá nhiều. Vì vậy để cải tiến người ta thay đồ thị (s,ψ) bằng đồ thị $(s,\Delta\psi)$:

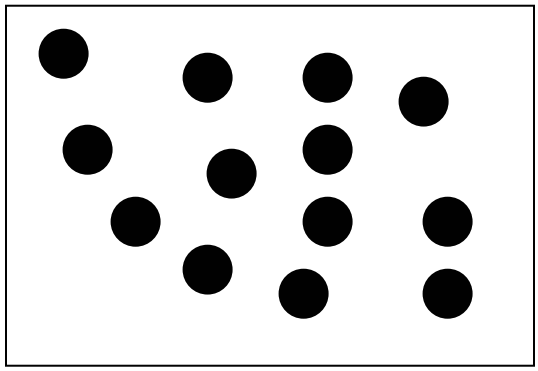
$$\Delta\psi = \psi - 2\pi.s/p$$

Trong đó p là chu vi của bao đóng. Đồ thị $(s,\Delta\psi)$ thực chất là đồ thị thể hiện sự khác nhau giữa đối tượng đang xét và một hình tròn có chu vi bằng với chu vi của đối tượng đang xét. Lúc này ta chỉ cần so sánh hai đồ thị $(s,\Delta\psi)$ của B và T . Do $\Delta\psi$ đã tuần hoàn theo s nên số phép so sánh đã giảm khá nhiều.

2 Đếm đối tượng

2.1 Đếm số đối tượng trên một ảnh đơn giản

- Xét một ảnh Grey-level bao gồm các đối tượng cùng loại, không có sự chồng chất lên nhau, và ảnh xem như không bị nhiễu.



Hình 2.1: *Ảnh đơn giản*

- Thao tác đếm số đối tượng trong ảnh được thực hiện đơn giản theo các bước sau:

+ Threshold ảnh đó để được ảnh Bi-level tương ứng, trong đó các đối tượng được phân biệt với màu nền.

+ Khởi gán biến đếm bằng 0.

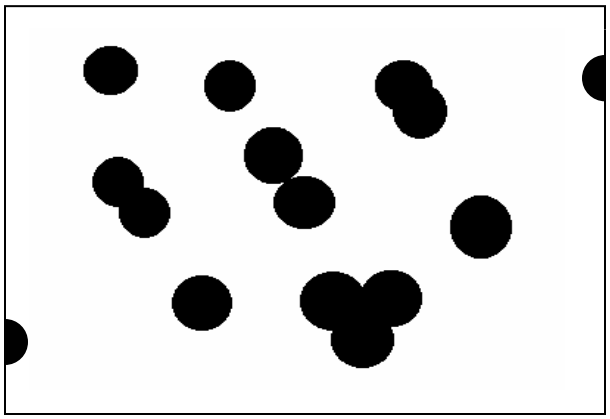
+ Tìm vùng đối tượng, đánh dấu vùng đó, tăng biến đếm lên 1, xoá vùng đó.

+ Bước thứ 3 được lặp lại cho đến khi không còn vùng nào được tìm thấy, khi đó giá trị biến đếm chính là số đối tượng thuộc ảnh.

2.2 Đếm số đối tượng trên ảnh phức tạp hơn

- Xét ảnh có các thuộc tính tương tự như trên nhưng trong đó các đối tượng có thể nằm chồng lên nhau. Khi đó, việc đếm số đối tượng trong ảnh phụ thuộc rất nhiều vào hình dạng đối tượng. Do đó, tùy theo trường hợp cụ thể mà ta có cách giải quyết vấn đề khác nhau.

- Xét ảnh bao gồm các đối tượng tròn có thể nằm chồng lên nhau.

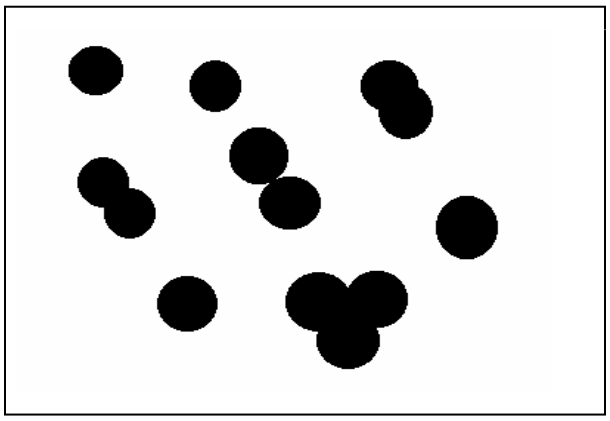


Hình 2.2.a: *Ảnh có các đối tượng chồng lên nhau*

- Do hình dạng của đối tượng rất quan trọng, nên thông thường các đối tượng nằm trên biên của ảnh thường được xoá đi. Điều này có thể được thực hiện theo cách sau:

+ Duyệt dòng trên nhất, dòng trái nhất và cột trái nhất, cột phải nhất của ảnh để tìm pixel đối tượng.

+ Bất kỳ pixel nào được tìm thấy, thì vùng đối tượng chứa pixel đó sẽ được đánh dấu và bị xoá đi.

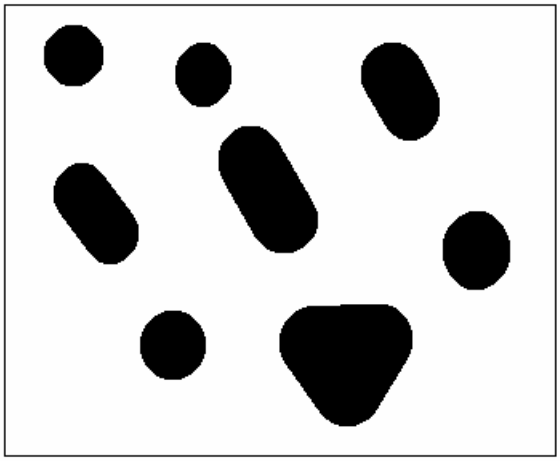


Hình 2.2.b: *Xoá đi các đối tượng thuộc biên ảnh*

- Bây giờ có thể đếm số đối tượng trong ảnh dựa vào các phương pháp sau.

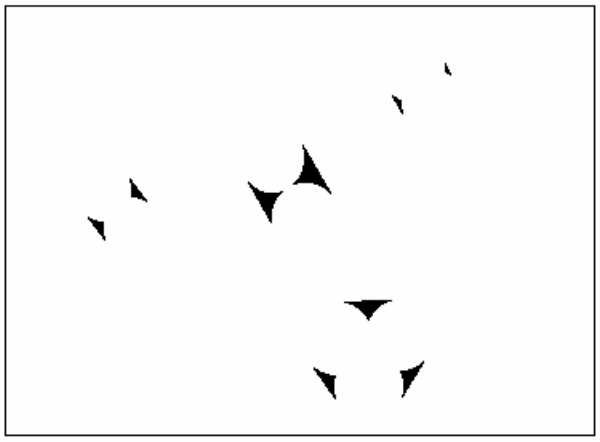
2.2.1 Đếm số đối tượng dựa vào bao lồi

- Từng vùng đối tượng được định vị và đánh dấu.
- Tìm bao lồi của từng vùng, lưu tất cả vào trong một ảnh (ảnh SECR – smallest enclosing convex region).



Hình 2.2.1.a: *Tìm bao lồi của đối tượng*

- Các vùng lõm (concave) được xác định bằng cách lấy ảnh bao lồi trừ cho ảnh gốc ban đầu . Thao tác trừ này có nghĩa là các pixel của đối tượng trong ảnh trừ sẽ là các pixel nền trong ảnh bị trừ.



Hình 2.2.1.b: *Các vùng lõm còn lại*

- Tính tỷ số diện tích từng vùng trong ảnh ban đầu so với diện tích của một đối tượng mẫu. Những vùng có tỷ số này nhỏ hơn hoặc bằng 1 được đếm là một đối tượng, còn những vùng có tỷ số này lớn hơn một có thể được đếm như sau:

Tỷ số diện tích	Số vùng lõm	Đếm
1-2	1-2	2
	3-4	3
2-3	1-2	3
	3-4	3
	5-6	4
3-4	1-6	4
	7-8	5

2.2.2 Đếm số đối tượng dựa vào phương pháp so mẫu

- Đối với những đối tượng tròn như trong ví dụ trên thì phương pháp so mẫu đối tượng rất có hiệu quả.
- Một mẫu có thể được lấy từ chính ảnh gốc và nên là mẫu điển hình cho lớp đối tượng đang xét. Trong ảnh đang xét, những đối tượng riêng rẽ sẽ được xét dựa vào các số liệu thống kê (diện tích, chu vi,..).
- Mẫu được di chuyển đến tất cả vị trí có thể trong ảnh dữ liệu, để tìm vùng tương ứng thích hợp (có chỉ số liên kết chuẩn lớn hơn 0). Khi tìm được vùng tương ứng biến đếm được tăng lên, đồng thời vùng đó cũng được đánh dấu và xoá đi.
- Kết quả của biến đếm chính là số đối tượng trong ảnh.

2.2.3 Đếm số đối tượng chồng nhau dựa vào phương pháp phân chia đối tượng (watershed method)

Đối với ảnh ví dụ trên, ta cũng có thể đếm số đối tượng dựa vào phương pháp phân chia đối tượng (watershed method), khi đó phương pháp được thực hiện như sau:

- Trước hết, xác định khoảng cách ngắn nhất từ mỗi pixel đối tượng đến pixel nền.
- Xác định vị trí của các pixel có giá trị lớn nhất K(gọi là đỉnh của vùng)đánh dấu chúng.
- Bắt đầu từ các pixel này, ta thực hiện việc mở rộng vùng sao cho các vùng đó không liên kết với nhau. Tức là thực hiện thao tác sau:

+ Đánh dấu các pixel có giá trị K

+ Đánh dấu những pixel thoả điều kiện:

- Kề với pixel có giá trị K.
- Có giá trị K-1.
- Không liên kết hai pixel thuộc các vùng không liên kết.

+ Giảm K xuống 1 giá trị.

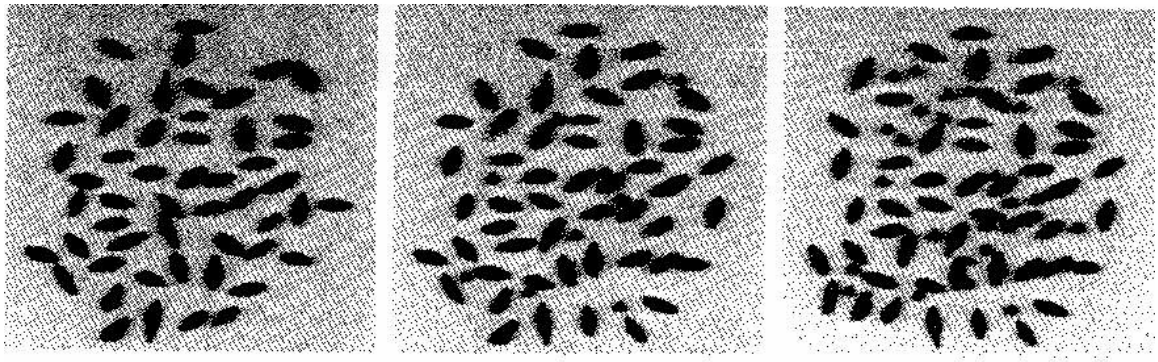
+ Thực hiện các bước 2 và 3 cho đến khi K=0.

Phương pháp phân chia đối tượng được áp dụng cho từng vùng một trong ảnh. Chúng ta thu được ảnh bao gồm các vùng không liên kết với nhau. Sau đó, việc đếm số đối tượng trong vùng trở nên đơn giản.

Tuy nhiên, trong một vài trường hợp hai đối tượng chồng lên nhau có thể tạo ra 3 đỉnh, và sẽ dẫn đến kết quả không chính xác: số đối tượng được đếm là 3.

2.3 Phân lớp các hạt trong ảnh

Ứng dụng này được dùng trong nông nghiệp. Ích lợi của nó là có thể tách các hạt ngũ cốc. Thực tế các hạt sạn thường bị trộn lẫn với các hạt ngũ cốc vì vậy chúng ta cần đếm số hạt ngũ cốc và hạt sạn để tìm tỷ lệ giữa chúng. Từ đó có thể giúp ích trong việc thống kê sau một vụ mùa.



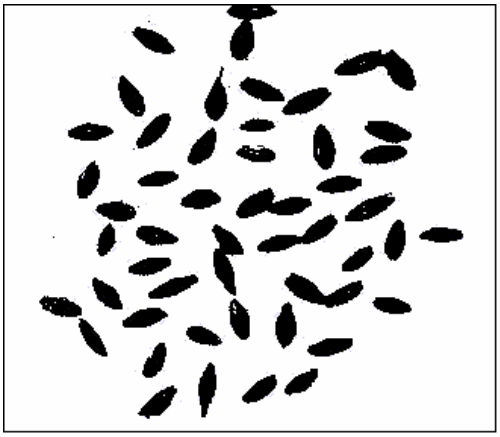
Ảnh các hạt ngũ cốc	Ảnh các hạt ngũ cốc Và các hạt sạn	Ảnh các hạt ngũ cốc và nhiều hạt sạn hơn
(i)	(ii)	(iii)

***Hình 2.3.a**: Ảnh minh họa*

Những hạt sạn thông thường khá nhỏ so với những hạt ngũ cốc, đồng thời hình dạng của chúng cũng khác nhau. Vì vậy các đặc trưng diện tích, chu vi là những đặc trưng thích hợp để có phân biệt.

Hình 2.3.a (i) trên là ảnh chỉ gồm những hạt ngũ cốc mà không có những hạt sạn. Những ảnh như vậy rất có lợi trong việc tính toán và thống kê những thuộc tính của những hạt ngũ cốc, có thể giúp ta có một số thông tin hữu ích để phân biệt và đếm những hạt ngũ cốc.

Khảo sát ảnh 2.3.a (i) để đếm số hạt ngũ cốc .Trước hết ta cần đặt ngưỡng cho ảnh này.



Hình 2.3.b: *Ảnh Bi-level của ảnh 2.3.a (i)*

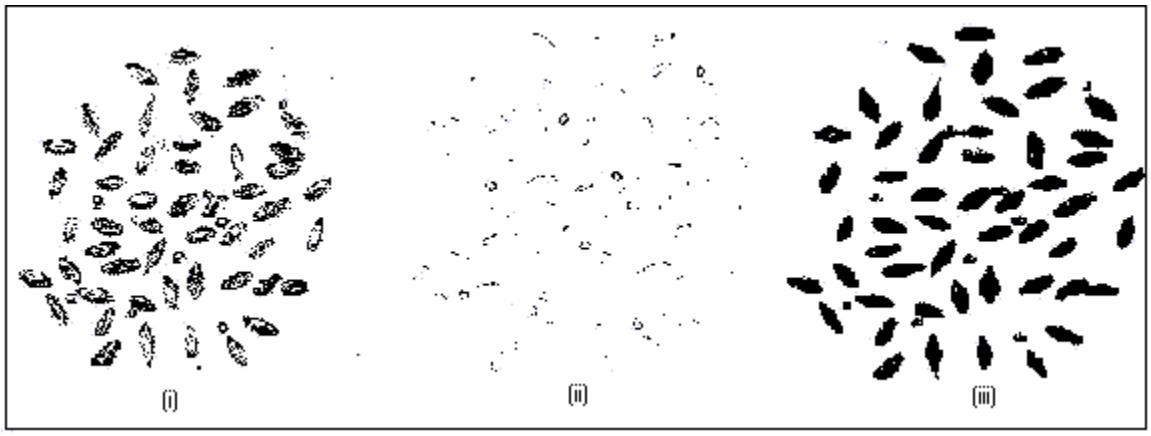
Do khi chụp ảnh, các hạt ngũ cốc tạo ra bóng của nó trong ảnh. Sự xuất hiện của những bóng này tạo ra những vùng tối trong ảnh, làm ảnh sau khi đặt ngưỡng có hiện tượng hai hạt nối dính nhau thành một đối tượng. Vì vậy khi tính diện tích của các đối tượng trong ảnh để phân biệt và đếm thì diện tích của các đối tượng xuất hiện hai nhóm phân biệt :

+ Nhóm 1 có diện tích 800 $\rightarrow$1300 pixel.

+ Nhóm 2 có diện tích 2100 $\rightarrow$ 2800 pixel.

Vì vậy khi đếm số hạt ngũ cốc, nếu đối tượng thuộc nhóm 1 thì được đếm là 1 hạt còn nếu đối tượng thuộc nhóm 2 thì được đếm là 2 hạt.

Nhưng khi xét đến ảnh có những hạt sạn như hình 2.3.a (ii) và 2.3.a (iii). Sau khi đặt ngưỡng cho ảnh thì xuất hiện những đối tượng có diện tích thuộc nhóm 2 nhưng thực chất chỉ có một hạt ngũ cốc. Vì đối tượng này có được là do một hạt ngũ cốc, bóng của nó và một hạt sạn đứng gần nhau tạo thành. Vì vậy làm cho việc đếm số hạt ngũ cốc trong hình có hạt sạn không dễ dàng. Để có thể đếm được các hạt ngũ cốc trong những hình này có một phương pháp như sau:



Hình 2.3.c*: Hình minh họa phương pháp đếm hạt ngũ cốc*

Xét ảnh 2.3.a (ii) là ảnh gồm những hạt ngũ cốc và hạt sạn. Trước hết ta lọc cạnh ảnh này (bằng mặt nạ Sobel chẳng hạn) cho ra ảnh 2.3.c (i) trên. Việc lọc cạnh này có ý nghĩa loại trừ bóng của hạt ngũ cốc. Sau đó đặt ngưỡng cho ảnh 2.3.c (i) bằng giá trị T_{new} chứ không phải giá trị T_{is} thông thường như đã thực hiện ở chương 1.

$$T_{new} = T_{is}+(255-T_{is})/4$$

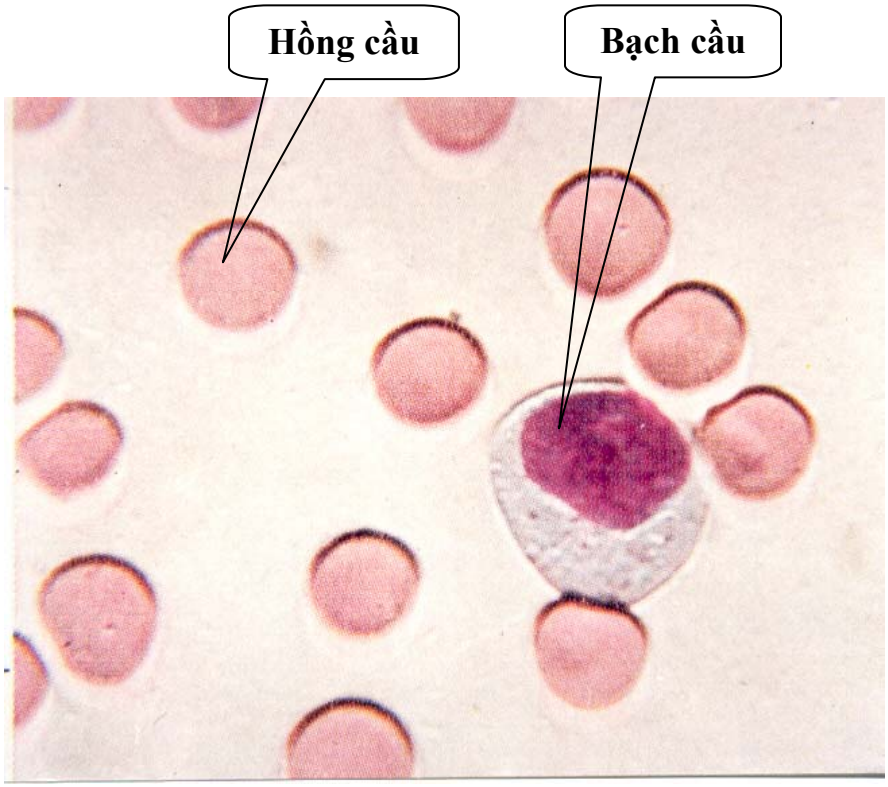
Việc đặt ngưỡng này nhằm tìm ra những cạnh xung quanh hạt là nơi có cường độ sáng đậm nhất. Ảnh 2.3.c (ii) là ảnh sau khi đặt ngưỡng ảnh 2.3.c (i). Cuối cùng ta dùng ảnh đã đặt ngưỡng của ảnh 2.3.a (ii) ban đầu trừ cho ảnh 2.3.c (ii). Thao tác trừ ở đây là những vị trí pixel đen trong ảnh 2.3.c (ii) sẽ tương ứng là những điểm trắng trong ảnh đã đặt ngưỡng của ảnh 2.3.a (ii). Ảnh 2.3.c (iii) là kết quả sau khi trừ. Ảnh 2.3.c (iii) này là ảnh đã tách riêng các hạt sạn ra với các hạt ngũ cốc. Lúc đó ta có thể dễ dàng đếm được các hạt ngũ cốc. Bảng dữ liệu sau là kết quả đếm các hạt trong ba bức ảnh 2.3.a

Lớp	Ngũ cốc	Sạn	Tỷ lệ(sạn/ngũ cốc)
Ảnh 2.3.a (i)	52	0	0
Ảnh 2.3.a (ii)	51	9	0.18
Ảnh 2.3.a (iii)	49	25	0.51

CHƯƠNG 3: THUẬT TOÁN ĐẾM SỐ LƯỢNG BẠCH CẦU - HỒNG CẦU VÀ ĐÁNH GIÁ

1 Bài toán

- Yêu cầu bài toán: Nhận dạng và phân lớp để có thể đếm số Bạch cầu (BC)và Hồng cầu (HC) trong một ảnh bitmap.
- Giới thiệu ảnh:



Hình 1*: *Minh hoạ ảnh Bạch cầu và Hồng cầu

- Một số đặc điểm của các đối tượng Hồng cầu và Bạch cầu trong ảnh :
  - Cả hai loại đối tượng đều có hình dạng gần giống với hình tròn.
  - Bạch cầu to hơn nhiều so với Hồng cầu.
  - Màu của Bạch cầu luôn đậm hơn Hồng cầu và màu Hồng cầu đậm hơn so với nền của ảnh.
  - Các đối tượng trong ảnh có thể dính với nhau hoặc chồng chất lên nhau.
  - Bạch cầu có màng bao quanh nhân.
- Phân tích ảnh :
  - Thuận lợi :
    - Màu sắc của Bạch cầu và Hồng cầu trong ảnh khác nhau.

- Diện tích của Bạch cầu và Hồng cầu trong ảnh khác nhau.
- Các đối tượng có hình dạng khá tròn
- Bạch cầu có thể dính với Hồng cầu chứ không chồng chất lên nhau và sự dính nhau này có biên giữa các đối tượng khá rõ ràng

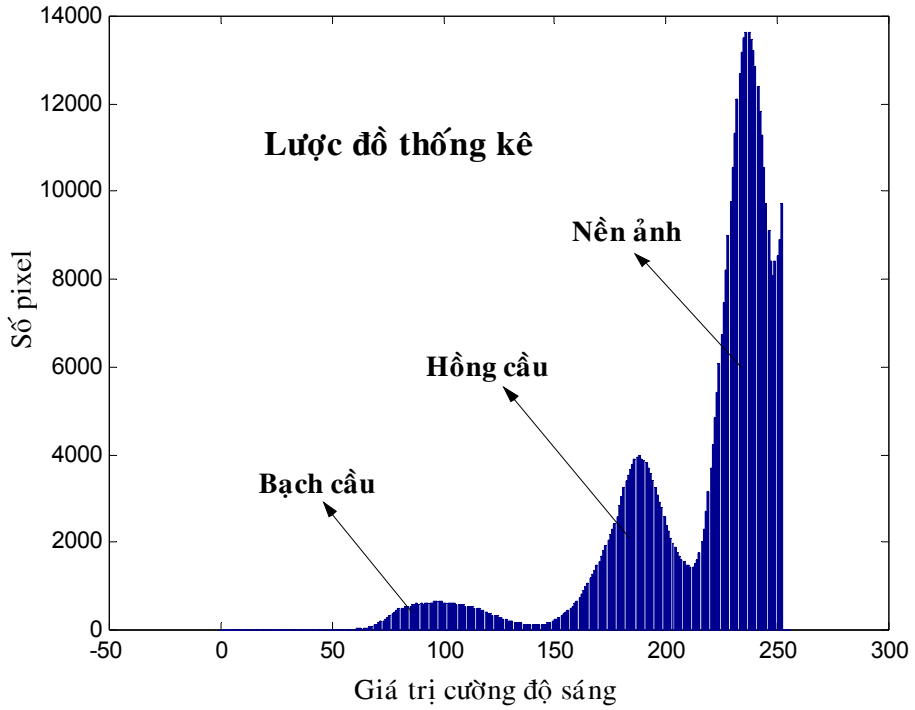
- Khó khăn :
  - Ảnh đôi khi bị nhiễu.
  - Các đối tượng dính nhau giữa HC và BC , giữa HC và HC , giữa BC và BC
  - Một số đối tượng ở biên của ảnh chỉ bằng một phần của các đối tượng do việc chia khung trong thao tác lấy mẫu máu và chụp ảnh.

2 Hướng giải quyết

Dựa vào một số đặc điểm của Bạch cầu và Hồng Cầu, chúng ta có thể đưa ra một số hướng giải quyết như sau:

- Dựa vào đặc trưng màu sắc khá khác nhau giữa Bạch cầu, Hồng cầu và nền ảnh, dẫn đến giá trị cường độ sáng của chúng cũng khác nhau khá rõ rệt trong ảnh Greylevel. Từ đó, ta có thể chọn được giá trị ngưỡng Threshold để có thể lọc ra ảnh Bi-level chỉ có Bạch cầu và ảnh Bi-level gồm Bạch Cầu và Hồng cầu bằng cách dựa vào Biểu đồ thống kê.

  Trong Biểu đồ thống kê của các ảnh đều có hình dạng của ba quả núi. Tương tự như sau:



Hình 2 : *Minh họa Lược đồ thống kê của một ảnh Bạch cầu và Hồng cầu*

Do đó ta có thể chọn giá trị ngưỡng Threshold ở các giá trị cường độ sáng ứng với các nơi trũng trong lược đồ.

- Do Bạch cầu chỉ có thể dính với Hồng cầu chứ không chồng chất lên nhau và sự dính nhau này có biên giữa các đối tượng khá rõ ràng. Từ đặc điểm này ta có thể tách rời được Bạch cầu và Hồng cầu khi chúng dính nhau bằng cách lấy ảnh Bi-level gồm Bạch cầu và Hồng cầu trừ cho ảnh Bi-level chỉ gồm cạnh của các đối tượng (ảnh này là ảnh đặt ngưỡng của ảnh lọc cạnh của ảnh ban đầu).Thao tác trừ này có nghĩa là các pixel của đối tượng trong ảnh trừ sẽ là các pixel nền trong ảnh bị trừ.
- Vì các đối tượng trong ảnh có hình dạng tròn nên ta có thể áp dụng phương pháp bao lồi (trong chương 2)để đếm khi chúng dính nhau hoặc chồng chất lên nhau.
- Để xử lý các đối tượng không đạt ở biên (là những đối tượng nhỏ hơn một nửa đối tượng thông thường), ta loại bỏ chúng bằng cách xác định diện tích của các đối tượng đó rồi so sánh để loại bỏ trong khi thực hiện thao tác đếm.

2.1 Thuật toán tổng quan

Thuật toán tổng quan gồm các bước cơ bản sau:

1. Đọc ảnh Bitmap và tìm ảnh Greylevel.
2. Tìm ảnh Bi-level chỉ có Bạch cầu và ảnh Bi-level gồm Bạch cầu và Hồng cầu.
3. Tách Bạch Cầu và Hồng cầu ra trong ảnh Bi-level gồm Bạch cầu và Hồng cầu.
4. Triệt tiêu Bạch cầu trong ảnh vừa tách .
5. Đếm Bạch cầu và Hồng cầu trong các ảnh tìm được.

2.2 Thuật toán chi tiết

Tùy theo điều kiện của các ảnh ta có các trường hợp sau :

2.2.1 Trường hợp 1

Trường hợp khi các đối tượng trong ảnh rời nhau (không dính nhau hoặc chồng chất lên nhau) gồm các bước chính sau:

1. Đọc ảnh Bitmap để lấy dữ liệu và chuyển sang ảnh Greylevel (ảnh ban đầu).
2. Đặt ngưỡng Threshold dựa vào Biểu đồ thống kê để lọc ra ảnh Bi-level chỉ có Bạch cầu (gọi là ảnh 1) và ảnh Bi-level gồm Bạch Cầu và Hồng cầu (gọi là ảnh 2).
3. Đếm số Bạch cầu trong ảnh 1 và số Bạch cầu và Hồng cầu trong ảnh 2 đồng thời loại bỏ các đối tượng không đạt. Suy ra số Hồng cầu.

2.2.2 Trường hợp 2

Trường hợp các đối tượng trong ảnh có biên của đối tượng rõ ràng (Khi bỏ biên của các đối tượng thì chúng sẽ rời nhau)

1. Đọc ảnh Bitmap để lấy dữ liệu và chuyển sang ảnh Greylevel (gọi là ảnh ban đầu).

2. Đặt ngưỡng Threshold dựa vào Biểu đồ thống kê để lọc ra ảnh Bi-level chỉ có Bạch cầu (gọi là ảnh 1) và ảnh Bi-level gồm Bạch Cầu và Hồng cầu (gọi là ảnh 2).
3. Lọc cạnh ảnh ban đầu và đặt ngưỡng Threshold để có ảnh Bi-level chỉ gồm cạnh của các đối tượng (gọi là ảnh 3).
4. Tách Bạch cầu và Hồng cầu ra khi chúng liên kết nhau bằng cách lấy ảnh 1 trừ ảnh 3 và ảnh 2 trừ ảnh 3.
5. Đếm số Bạch cầu trong ảnh 1 và số Bạch cầu và Hồng cầu trong ảnh 2 đồng thời loại bỏ các đối tượng không đạt. Suy ra số Hồng cầu.

2.2.3 Trường hợp 3

Trong trường hợp bất kỳ gồm các bước sau:

1. Đọc ảnh Bitmap để lấy dữ liệu và chuyển sang ảnh Greylevel (gọi là ảnh ban đầu).
2. Đặt ngưỡng Threshold dựa vào Biểu đồ thống kê để lọc ra ảnh Bi-level chỉ có Bạch cầu (gọi là ảnh 1) và ảnh Bi-level gồm Bạch Cầu và Hồng cầu (gọi là ảnh 2).
3. Lọc cạnh ảnh ban đầu và đặt ngưỡng Threshold để có ảnh Bi-level chỉ gồm cạnh của các đối tượng (gọi là ảnh 3).
4. Tách Bạch cầu và Hồng cầu ra khi chúng liên kết nhau bằng cách lấy ảnh 2 trừ ảnh 3.
5. Triệt tiêu Bạch Cầu trong ảnh 2
6. Đếm số Bạch cầu trong ảnh 1 và số Hồng cầu trong ảnh 2 bằng phương pháp bao lồi đồng thời loại bỏ các đối tượng không đạt.

3 Đánh giá thuật toán

Thời gian thực hiện trung bình:

- Thời gian thực hiện trung bình của thuật toán trong trường hợp 1 : 1 phút 45 giây.
- Thời gian thực hiện trung bình của thuật toán trong trường hợp 2 : 1 phút 51 giây.
- Thời gian thực hiện trung bình của thuật toán trong trường hợp 3 : 2 phút 37 giây.

Thuật toán chạy nhanh hay chậm ít phụ thuộc vào kích thước ảnh mà chủ yếu phụ thuộc vào sự phức tạp của ảnh (ảnh bị nhiễu nhiều hay ít, các đối tượng Bạch cầu và Hồng cầu nhiều hay ít).

4 Mô tả cài đặt

Chương trình được cài đặt thông qua năm tập tin chính gồm:

1. **BmpImage.java**
2. **SimpleImage.java**
3. **Algorithm.java**
4. **Application.java**
5. **ExampleFileFilter.java**

Mỗi tập tin này chứa một lớp chính có tên lớp ứng với tên tập tin. Các lớp chính này có chức năng:

- **Lớp BmpImage** : là lớp có thành phần dữ liệu ứng với các thành phần dữ liệu trong ảnh Bitmap(tiêu đề, bảng màu và dữ liệu). Lớp này được dùng để giúp cho việc đọc và ghi ảnh Bitmap.
- **Lớp SimpleImage** : là lớp có thành phần dữ liệu là chiều rộng, chiều cao và một ma trận chứa dữ liệu hình ảnh. Lớp có các phương thức có chức năng thực hiện mọi thao tác trên ảnh.
- **Lớp Algorithm** : là lớp chỉ chứa các phương thức có chức năng thực hiện các thuật toán của chương trình.
- **Lớp Application** : là lớp thể hiện phần giao diện của chương trình. Lớp này được gọi để chạy chương trình.
- **Lớp ExampleFileFilter** : là lớp trợ giúp cho việc tìm kiếm các tập tin có đuôi .bmp(ảnh Bitmap).

Lớp SimpleImage là lớp quan trọng nhất của chương trình. Lớp này gồm các phương thức chính như sau:

-long[] histogram(int bin_width) : Phương thức có chức năng tìm các giá trị trong biểu đồ thống kê tuỳ theo độ rộng bin_width truyền vào.

-void smoothHisto(long[] histo,int width) : Phương thức có chức năng làm trơn biểu đồ thống kê theo độ rộng width truyền vào.

-void thresholdImage(int t) : Phương thức có chức năng đặt ngưỡng Threshold ảnh theo giá trị ngưỡng t truyền vào.

-void threshIs() : Phương thức cũng có chức năng đặt ngưỡng Threshold với giá trị ngưỡng được tìm thông qua phương pháp chọn lặp nhiều lần (Iterative Selection)

-void thresholdBHC(SimpleImage bc,SimpleImage hc) : Phương thức có chức năng đặt ngưỡng Threshold sao cho ảnh bc chỉ chứa Bạch cầu và ảnh hc chứa Bạch cầu và Hồng cầu.

-void thresholdEgde() : Phương thức có chức năng đặt ngưỡng cho ảnh đã lọc cạnh.

-long area(int value) : Phương thức có chức năng tính diện tích của đối tượng có giá trị value.

-void minus(SimpleImage z) : Phương thức có chức năng trừ cho ảnh z (tức là những điểm thuộc đối tượng trên ảnh z sẽ là nền trong ảnh gọi phương thức).

-void edgeSobel() : Phương thức có chức năng lọc cạnh theo mặt nạ Sobel.

-void delBC(SimpleImage simgBC) : Phương thức có chức năng triệt tiêu đối tượng Bạch cầu trong ảnh gọi phương thức thông qua vị trí Bạch cầu của ảnh simgBC truyền vào.

-int processNoise(int oldVal,int newVal,int iseed,int jseed,long temp) : Phương thức có chức năng kiểm tra và xử lý các vùng không phải là đối tượng Bạch cầu và Hồng cầu

-void deleteNoiseHC() : Phương thức có chức năng xóa tất cả các vùng có diện tích nhỏ hơn Hồng cầu (chúng được xem là vùng nhiễu).

-void deleteNoiseBC() : Phương thức có chức năng xóa tất cả các vùng có diện tích nhỏ hơn Bạch cầu (chúng được xem là nhiễu).

-void fillHole() : Phương thức có chức năng làm đầy các lỗ trống bên trong các đối tượng.

-long areaCell() : Phương thức có chức năng tìm diện tích đối tượng mẫu (đối tượng thông thường)trong ảnh.

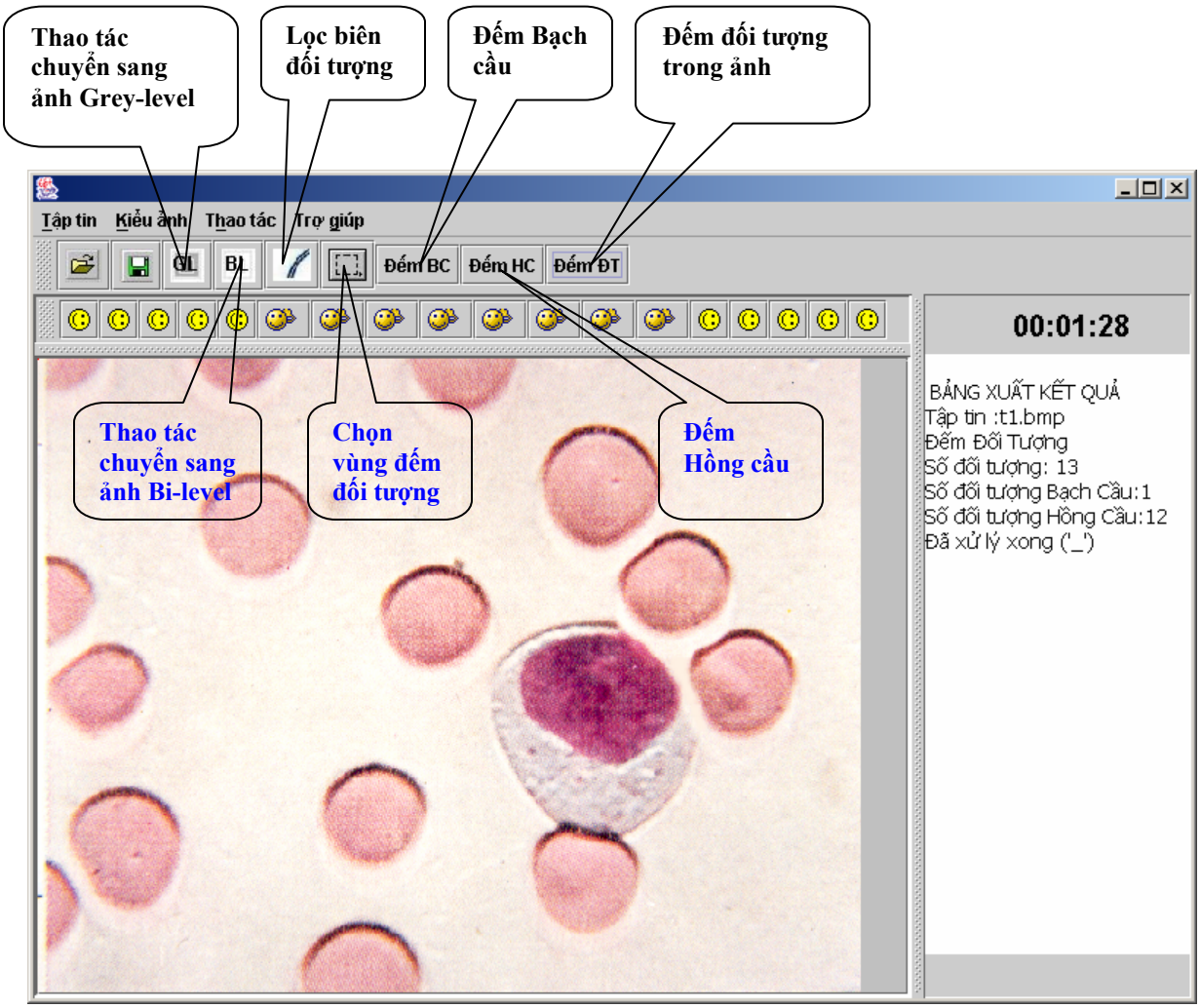
-int countObject() : Phương thức có chức năng đếm số đối tượng trong ảnh (khi ảnh có các đối tượng rời nhau).

-void convexity(int val) : Phương thức có chức năng tìm bao lồi cho các đối tượng trong ảnh.

-int indexTable(double fraction, int concavities) : Phương thức có chức năng bảng chỉ mục để xác định số đối tượng của một vùng, được sử dụng trong phương pháp bao lồi.

-int countOverlap(SimpleImage bdDN,SimpleImage bdFH) : Phương thức có chức năng thực hiện việc đếm đối tượng bằng phương pháp bao lồi.

5 Giao diện chương trình ứng dụng



6 Hạn chế và hướng phát triển

6.1 Hạn chế

Chương trình còn nhiều hạn chế như sau:

- Chỉ có thể thực hiện được cho các ảnh Bạch cầu và Hồng cầu trong thực tế.
- Do thiếu phương tiện cùng kỹ thuật chuyên môn nên chưa có nhiều ảnh để kiểm tra tính đúng đắn của thuật toán.
- Cho kết quả sai đối với một số đối tượng không rõ ràng (về màu sắc, diện tích, hình dạng tròn).

6.2 Hướng phát triển

Nếu được tiếp tục mở rộng, chương trình có thể được xây dựng thành một phần mềm hoàn chỉnh đếm chính xác hơn, nhanh hơn số lượng hồng cầu, bạch cầu trên mẫu máu; từ đó đưa ra kết quả sơ lược cho việc chẩn đoán bệnh.

Từ những kiến thức cơ bản của Thị Giác Máy Tính, có thể lập trình cho máy tính có khả năng nhận biết, phân lớp và đếm các đối tượng khác.

Có thể kết hợp với logic mờ và thuật toán di truyền để phân lớp đối tượng.

TÀI LIỆU THAM KHẢO

Trong quá trình thực hiện đề tài chúng em đã tham khảo một số tài liệu sau:

[1] J.R.Parker. Practical Computer Vision Using C. Yale University Press, 1995.

[2] Ramesh Jain, Rangachar Kasturi, Brian G.Schunck. Machine Vision. MIT Press and McGraw-Hill, Inc, 1995.

[3] E.R.Davies. Machine Vision:Theory, Algorithms, Practicalities. Academic Press, 1997.

[3] C.Wayne Brown, Barry J.Shepherd. Graphic File Formats. Academic Press, 1995.

[4] Hoàng Ngọc Giao. Lập trình Java như thế nào? NXB Thống kê-Hà Nội, 1998.