

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



# **XỬ LÝ ẢNH**

*(Dùng cho sinh viên hệ đào tạo đại học từ xa)*

**Lưu hành nội bộ**

**HÀ NỘI - 2006**

# XỬ LÝ ẢNH

Biên soạn : PGS.TS NGUYỄN QUANG HOAN

# LỜI NÓI ĐẦU

Trong các trường đại học, cao đẳng, xử lý ảnh đã trở thành một môn học chuyên ngành của sinh viên các ngành Công nghệ Thông tin, Viễn thông. Giáo trình và tài liệu về lĩnh vực này ở nước ta còn đang hạn chế. Để đáp ứng kịp thời cho đào tạo từ xa, Học viện Công nghệ Bưu chính Viễn thông cố gắng kịp thời biên soạn tài liệu này cho sinh viên, đặc biệt hệ Đào tạo từ xa học tập. Trong quá trình biên soạn, chúng tôi có tham khảo các tài liệu của Đại học Bách khoa Hà nội [1] giáo trình gần gũi về tính công nghệ với Học viện. Một số giáo trình khác của Đại học Quốc gia thành phố Hồ Chí Minh [2], tài liệu trên mạng và các tài liệu nước ngoài bằng tiếng Anh [5, 6, 7] cũng được tham khảo và giới thiệu để sinh viên đào tạo từ xa đọc thêm.

Tài liệu này nhằm hướng dẫn và giới thiệu những kiến thức cơ bản, các khái niệm, định nghĩa tóm tắt. Một số thuật ngữ được chú giải bằng tiếng Anh để học viên đọc bằng tiếng Anh dễ dàng, tránh hiểu nhầm khi chuyển sang tiếng Việt.

Tài liệu gồm các chương sau:

- Chương 1. Nhập môn xử lý ảnh
- Chương 2. Thu nhận ảnh
- Chương 3. Xử lý nâng cao chất lượng ảnh
- Chương 4. Các phương pháp phát hiện biên ảnh
- Chương 5. Phân vùng ảnh
- Chương 6. Nhận dạng ảnh
- Chương 7. Nén dữ liệu ảnh

Còn nhiều vấn đề khác như các công cụ toán học, kỹ thuật biến đổi ảnh, truyền ảnh, các phần mềm xử lý v.v... chưa đề cập được trong phạm vi tài liệu này. Đề nghị các bạn đọc tìm hiểu thêm sau khi đã có những kiến thức cơ bản này.

Tuy có tham gia giảng dạy môn xử lý ảnh ở cấp Đại học một số năm, nhiều lớp có trình độ khác nhau; chủ nhiệm một số đề tài nghiên cứu Cơ bản Nhà nước, đề tài cấp Bộ liên quan nhưng “Xử lý ảnh” là môn học có sự kết hợp nhiều giữa nhiều lĩnh vực khoa học và công nghệ nên có thể coi là môn học khó. Nhiều cố gắng để cập nhật kiến thức nhưng thời gian, điều kiện, khả năng có hạn nên tài liệu chắc chắn còn nhiều thiếu sót. Chúng tôi mong nhận được nhiều ý kiến đóng góp để tài liệu được hoàn thiện hơn cho các lần tái bản sau.

**Hà Nội, tháng 12 năm 2006**

**Tác giả**

# CHƯƠNG 1: NHẬP MÔN XỬ LÝ ẢNH

**Học xong phần này sinh viên có thể nắm được:**

1. Ý nghĩa, mục đích môn học; các ứng dụng của xử lý ảnh trong công nghiệp, đời sống
2. Các thành phần: xử lý và phân tích ảnh. Các khối chức năng, ý nghĩa của nó trong quy trình xử lý và phân tích ảnh.
3. Các khái niệm, định nghĩa ảnh số hóa. Tổng quan về biểu diễn ảnh.
4. Một số công cụ xử lý ảnh. Các vấn đề đặt ra với xử lý ảnh.

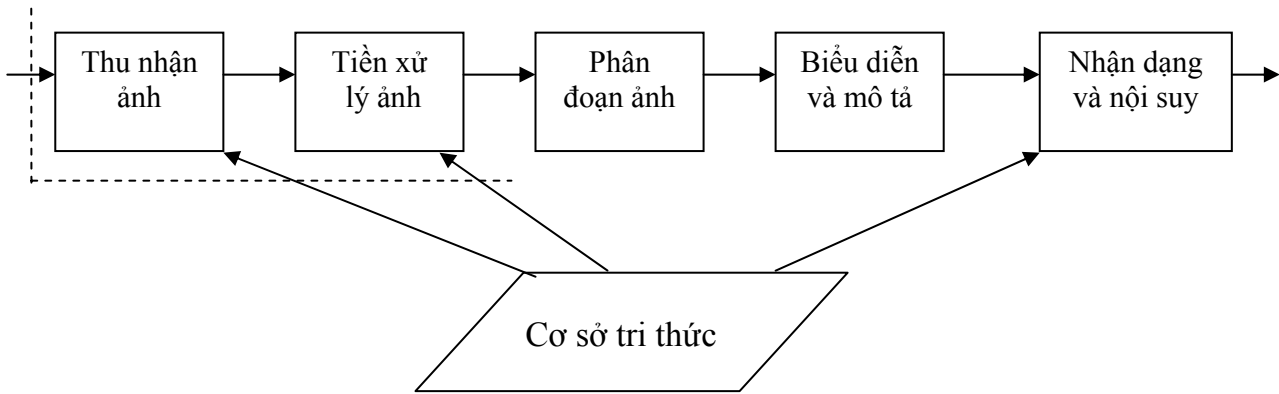
## 1.1 GIỚI THIỆU VỀ HỆ THỐNG XỬ LÝ ẢNH

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số là một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn... Thứ hai, các công cụ toán như Đại số tuyến tính, Sác xuất, thống kê. Một số kiến thức cần thiết như Trí tuệ nhân tạo, Mạng nơ ron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền qua cáp từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh số thuận lợi. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu nhiều kết quả khả quan.

Để dễ tưởng tượng, xét các bước cần thiết trong xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kính CCIR). Gần đây, với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. (Máy ảnh số hiện nay là một thí dụ gần gũi). Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh; có thể quét từ ảnh chụp bằng máy quét ảnh. Hình 1.1 dưới đây mô tả các bước cơ bản trong xử lý ảnh.



**Hình 1.1** Các bước cơ bản trong xử lý ảnh

Sơ đồ này bao gồm các thành phần sau:

**a) Phần thu nhận ảnh (Image Acquisition)**

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hoá (như loại CCD – Charge Coupled Device) là loại photodiode tạo cường độ sáng tại mỗi điểm ảnh.

Camera thường dùng là loại quét dòng ; ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh)

**b) Tiền xử lý (Image Processing)**

Sau bộ thu nhận, ảnh có thể nhiều độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

**c) Phân đoạn (Segmentation) hay phân vùng ảnh**

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu, chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

**d) Biểu diễn ảnh (Image Representation)**

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (*Feature Selection*) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

**e) Nhận dạng và nội suy ảnh (Image Recognition and Interpretation)**

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại. Có nhiều cách phân loại ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản:

- Nhận dạng theo tham số.
- Nhận dạng theo cấu trúc.

Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người...

#### **f) Cơ sở tri thức (Knowledge Base)**

Như đã nói ở trên, ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiều. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy. Trong tài liệu, chương 6 về nhận dạng ảnh có nêu một vài ví dụ về cách sử dụng các cơ sở tri thức đó.

#### **g) Mô tả (biểu diễn ảnh)**

Từ Hình 1.1, ảnh sau khi số hoá sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh thô đó được đặc tả (biểu diễn) lại (hay đơn giản là mã hoá) theo các đặc điểm của ảnh được gọi là các đặc trưng ảnh (*Image Features*) như: biên ảnh (*Boundary*), vùng ảnh (*Region*). Một số phương pháp biểu diễn thường dùng:

- Biểu diễn bằng mã chạy (Run-Length Code)
- Biểu diễn bằng mã xích (Chain-Code)
- Biểu diễn bằng mã tứ phân (Quad-Tree Code)

#### **Biểu diễn bằng mã chạy**

Phương pháp này thường biểu diễn cho vùng ảnh và áp dụng cho ảnh nhị phân. Một vùng ảnh  $R$  có thể mã hoá đơn giản nhờ một ma trận nhị phân:

$$U(m, n) = 1 \quad \text{nếu } (m, n) \text{ thuộc } R$$
$$U(m, n) = 0 \quad \text{nếu } (m, n) \text{ không thuộc } R$$

Trong đó:  $U(m, n)$  là hàm mô tả mức xám ảnh tại tọa độ  $(m, n)$ . Với cách biểu diễn trên, một vùng ảnh được mô tả bằng một tập các chuỗi số 0 hoặc 1. Giả sử chúng ta mô tả ảnh nhị phân của một vùng ảnh được thể hiện theo tọa độ  $(x, y)$  theo các chiều  $x$  và  $y$  và đặc tả chỉ đối với giá trị "1" khi đó dạng mô tả có thể là:  $(x, y)r$ ; trong đó  $(x, y)$  là tọa độ,  $r$  là số lượng các bit có giá trị "1" liên tục theo chiều ngang hoặc dọc.

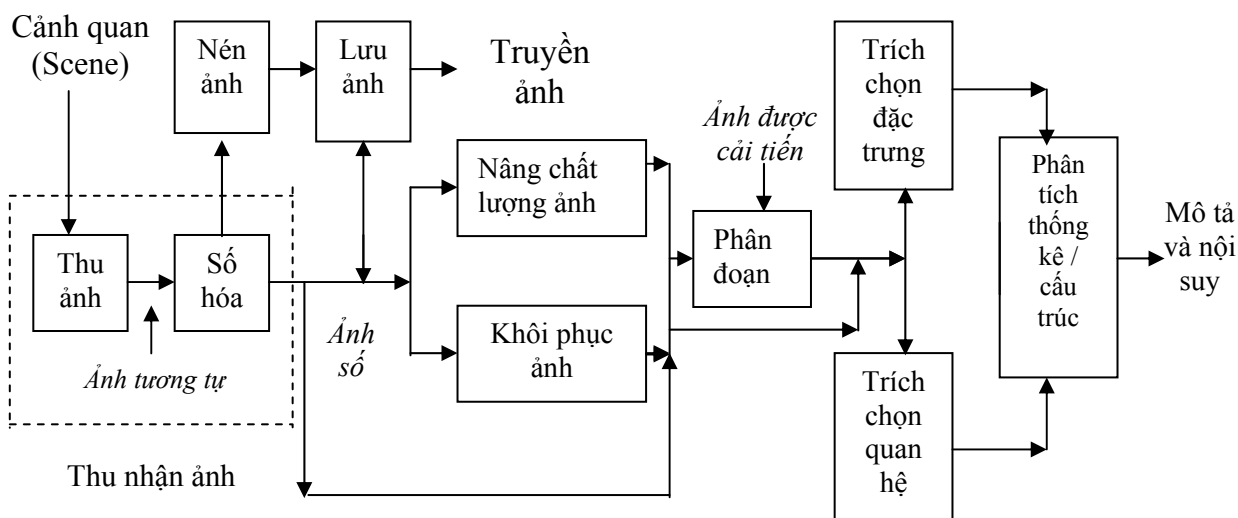
#### **Biểu diễn bằng mã xích**

Phương pháp này thường dùng để biểu diễn đường biên ảnh. Một đường bất kỳ được chia thành các đoạn nhỏ. Nối các điểm chia, ta có các đoạn thẳng kế tiếp được gán hướng cho đoạn thẳng đó tạo thành một dây xích gồm các đoạn. Các hướng có thể chọn 4, 8, 12, 24,... mỗi hướng được mã hoá theo số thập phân hoặc số nhị phân thành mã của hướng.

**Biểu diễn bằng mã tứ phân**

Phương pháp mã tứ phân được dùng để mã hoá cho vùng ảnh. Vùng ảnh đầu tiên được chia làm bốn phần thường là bằng nhau. Nếu mỗi vùng đã đồng nhất (chứa toàn điểm đen (1) hay trắng (0)), thì gán cho vùng đó một mã và không chia tiếp. Các vùng không đồng nhất được chia tiếp làm bốn phần theo thủ tục trên cho đến khi tất cả các vùng đều đồng nhất. Các mã phân chia thành các vùng con tạo thành một cây phân chia các vùng đồng nhất.

Trên đây là các thành phần cơ bản trong các khâu xử lý ảnh. Trong thực tế, các quá trình sử dụng ảnh số không nhất thiết phải qua hết các khâu đó tùy theo đặc điểm ứng dụng. Hình 1.2 cho sơ đồ phân tích và xử lý ảnh và lưu trữ thông tin giữa các khối một cách khá đầy đủ. Ảnh sau khi được số hóa được nén, lưu lại để truyền cho các hệ thống khác sử dụng hoặc để xử lý tiếp theo. Mặt khác, ảnh sau khi số hóa có thể bỏ qua công đoạn nâng cao chất lượng (khi ảnh đủ chất lượng theo một yêu cầu nào đó) để chuyển tới khâu *phân đoạn* hoặc bỏ tiếp khâu *phân đoạn* chuyển trực tiếp tới khâu *trích chọn đặc trưng*. Hình 1.2 cũng chia các nhánh song song như: nâng cao chất lượng ảnh có hai nhánh phân biệt: nâng cao chất lượng ảnh (tăng độ sáng, độ tương phản, lọc nhiễu) hoặc khôi phục ảnh (hồi phục lại ảnh thật khi ảnh nhận được bị méo) v.v...



**Hình 1.2** Sơ đồ phân tích và xử lý ảnh và lưu trữ thông tin giữa các khối

## 1.2 NHỮNG VẤN ĐỀ CƠ BẢN TRONG HỆ THỐNG XỬ LÝ ẢNH

### 1.2.1 Điểm ảnh (Picture Element)

Gốc của ảnh (ảnh tự nhiên) là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính (số), ảnh cần phải được số hoá. Số hoá ảnh là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức xám). Khoảng cách giữa các điểm ảnh đó được thiết lập sao cho mắt người không phân biệt được ranh giới giữa

chúng. Mỗi một điểm như vậy gọi là điểm ảnh (PEL: Picture Element) hay gọi tắt là Pixel. Trong khuôn khổ ảnh hai chiều, mỗi pixel ứng với cặp tọa độ  $(x, y)$ .

**Định nghĩa:**

Điểm ảnh (Pixel) là một phần tử của ảnh số tại tọa độ  $(x, y)$  với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh đó được chọn thích hợp sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh.

### 1.2.2 Độ phân giải của ảnh

**Định nghĩa:** Độ phân giải (Resolution) của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị.

Theo định nghĩa, khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bố, đó chính là độ phân giải và được phân bố theo trục  $x$  và  $y$  trong không gian hai chiều.

Ví dụ: Độ phân giải của ảnh trên màn hình CGA (Color Graphic Adaptor) là một lưới điểm theo chiều ngang màn hình: 320 điểm chiều dọc \* 200 điểm ảnh (320\*200). Rõ ràng, cùng màn hình CGA 12" ta nhận thấy mịn hơn màn hình CGA 17" độ phân giải 320\*200. Lý do: cùng một mật độ (độ phân giải) nhưng diện tích màn hình rộng hơn thì độ mịn (liên tục của các điểm) kém hơn.

### 1.2.3 Mức xám của ảnh

Một điểm ảnh (pixel) có hai đặc trưng cơ bản là vị trí  $(x, y)$  của điểm ảnh và độ xám của nó. Dưới đây chúng ta xem xét một số khái niệm và thuật ngữ thường dùng trong xử lý ảnh.

**a) Định nghĩa:** Mức xám của điểm ảnh là cường độ sáng của nó được gán bằng giá trị số tại điểm đó.

**b) Các thang giá trị mức xám thông thường:** 16, 32, 64, 128, 256 (Mức 256 là mức phổ dụng. Lý do: từ kỹ thuật máy tính dùng 1 byte (8 bit) để biểu diễn mức xám: Mức xám dùng 1 byte biểu diễn:  $2^8=256$  mức, tức là từ 0 đến 255).

**c) Ảnh đen trắng:** là ảnh có hai màu đen, trắng (không chứa màu khác) với mức xám ở các điểm ảnh có thể khác nhau.

**d) Ảnh nhị phân:** ảnh chỉ có 2 mức đen trắng phân biệt tức dùng 1 bit mô tả  $2^1$  mức khác nhau. Nói cách khác: mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1.

**e) Ảnh màu:** trong khuôn khổ lý thuyết ba màu (Red, Blue, Green) để tạo nên thế giới màu, người ta thường dùng 3 byte để mô tả mức màu, khi đó các giá trị màu:  $2^{8*3}=2^{24}\approx 16,7$  triệu màu.

Chúng ta cần đọc thêm sách tham khảo để có thêm thông tin về các khái niệm này.

### 1.2.4 Định nghĩa ảnh số

Ảnh số là tập hợp các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật.

### 1.2.5 Quan hệ giữa các điểm ảnh



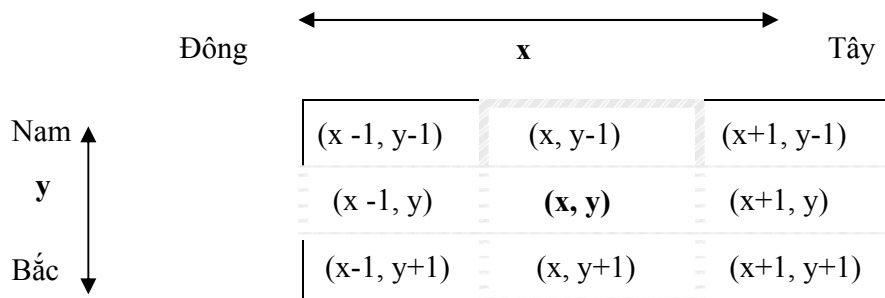
Một ảnh số giả sử được biểu diễn bằng hàm  $f(x, y)$ . Tập con các điểm ảnh là  $S$ ; cặp điểm ảnh có quan hệ với nhau ký hiệu là  $p, q$ . Chúng ta nêu một số các khái niệm sau.

**a) Các lân cận của điểm ảnh (Image Neighbors)**

\* Giả sử có điểm ảnh  $p$  tại tọa độ  $(x, y)$ .  $p$  có 4 điểm lân cận gần nhất theo chiều đứng và ngang (có thể coi như lân cận 4 hướng chính: Đông, Tây, Nam, Bắc).

$$\{(x-1, y); (x, y-1); (x, y+1); (x+1, y)\} = N_4(p)$$

trong đó: số  $1$  là giá trị logic;  $N_4(p)$  tập 4 điểm lân cận của  $p$ .



**Hình 1.3 Lân cận các điểm ảnh của tọa độ  $(x,y)$**

\* Các lân cận chéo: Các điểm lân cận chéo  $N_P(p)$  (Có thể coi lân cận chéo là 4 hướng: Đông-Nam, Đông-Bắc, Tây-Nam, Tây-Bắc)

$$N_P(p) = \{(x+1, y+1); (x+1, y-1); (x-1, y+1); (x-1, y-1)\}$$

\* Tập kết hợp:  $N_8(p) = N_4(p) + N_P(p)$  là tập hợp 8 lân cận của điểm ảnh  $p$ .

\* Chú ý: Nếu  $(x, y)$  nằm ở biên (mép) ảnh; một số điểm sẽ nằm ngoài ảnh.

**b) Các mối liên kết điểm ảnh.**

Các mối liên kết được sử dụng để xác định giới hạn (*Boundaries*) của đối tượng vật thể hoặc xác định vùng trong một ảnh. Một liên kết được đặc trưng bởi tính liên kề giữa các điểm và mức xám của chúng.

Giả sử  $V$  là tập các giá trị mức xám. Một ảnh có các giá trị cường độ sáng từ thang mức xám từ 32 đến 64 được mô tả như sau :

$$V = \{32, 33, \dots, 63, 64\}.$$

Có 3 loại liên kết.

\* **Liên kết 4:** Hai điểm ảnh  $p$  và  $q$  được nói là liên kết 4 với các giá trị cường độ sáng  $V$  nếu  $q$  nằm trong một các lân cận của  $p$ , tức  $q$  thuộc  $N_4(p)$

\* **Liên kết 8:** Hai điểm ảnh  $p$  và  $q$  nằm trong một các lân cận 8 của  $p$ , tức  $q$  thuộc  $N_8(p)$

\* **Liên kết  $m$  (liên kết hỗn hợp):** Hai điểm ảnh  $p$  và  $q$  với các giá trị cường độ sáng  $V$  được nói là liên kết  $m$  nếu.

1.  $q$  thuộc  $N_4(p)$  hoặc
2.  $q$  thuộc  $N_P(p)$

**c) Đo khoảng cách giữa các điểm ảnh.**

**Định nghĩa:** Khoảng cách  $D(p, q)$  giữa hai điểm ảnh  $p$  tọa độ  $(x, y)$ ,  $q$  tọa độ  $(s, t)$  là hàm khoảng cách (Distance) hoặc Metric nếu:

$$1. D(p, q) \geq 0 \text{ (Với } D(p, q) = 0 \text{ nếu và chỉ nếu } p = q)$$

$$2. D(p, q) = D(q, p)$$

$$3. D(p, z) \leq D(p, q) + D(q, z); z \text{ là một điểm ảnh khác.}$$

**Khoảng cách Euclide:** Khoảng cách Euclide giữa hai điểm ảnh  $p(x, y)$  và  $q(s, t)$  được định nghĩa như sau:

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$$

**Khoảng cách khối:** Khoảng cách  $D_4(p, q)$  được gọi là khoảng cách khối đồ thị (City-Block Distance) và được xác định như sau:

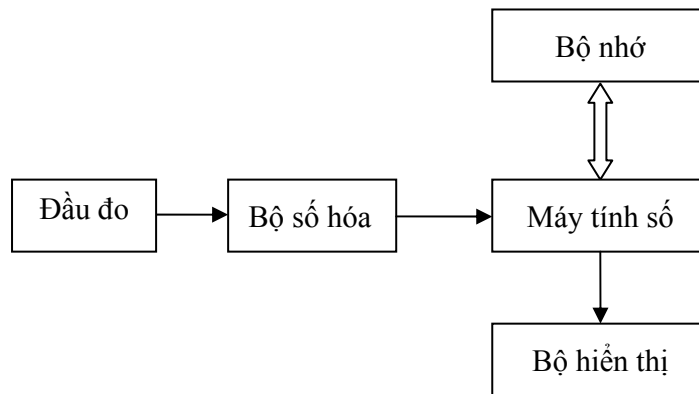
$$D_4(p, q) = |x - s| + |y - t|$$

Giá trị khoảng cách giữa các điểm ảnh  $r$ : giá trị bán kính  $r$  giữa điểm ảnh từ tâm điểm ảnh đến tâm điểm ảnh  $q$  khác. Ví dụ: Màn hình CGA 12" (12" \* 2,54cm = 30,48cm = 304,8mm) độ phân giải 320\*200; tỷ lệ 4/3 (Chiều dài/Chiều rộng). Theo định lý Pitago về tam giác vuông, đường chéo sẽ lấy tỷ lệ 5 phần (5/4/3: đường chéo/chiều dài/chiều rộng màn hình); khi đó độ dài thật là (305/244/183) chiều rộng màn hình 183mm ứng với màn hình CGA 200 điểm ảnh theo chiều dọc. Như vậy, khoảng cách điểm ảnh lân cận của CGA 12" là  $\approx 1$ mm.

Khoảng cách  $D_8(p, q)$  còn gọi là khoảng cách bàn cờ (Chess-Board Distance) giữa điểm ảnh  $p, q$  được xác định như sau:

$$D_8(p, q) = \max(|x - s|, |y - t|)$$

### 1.2.6 Các thành phần cơ bản của hệ thống xử lý ảnh



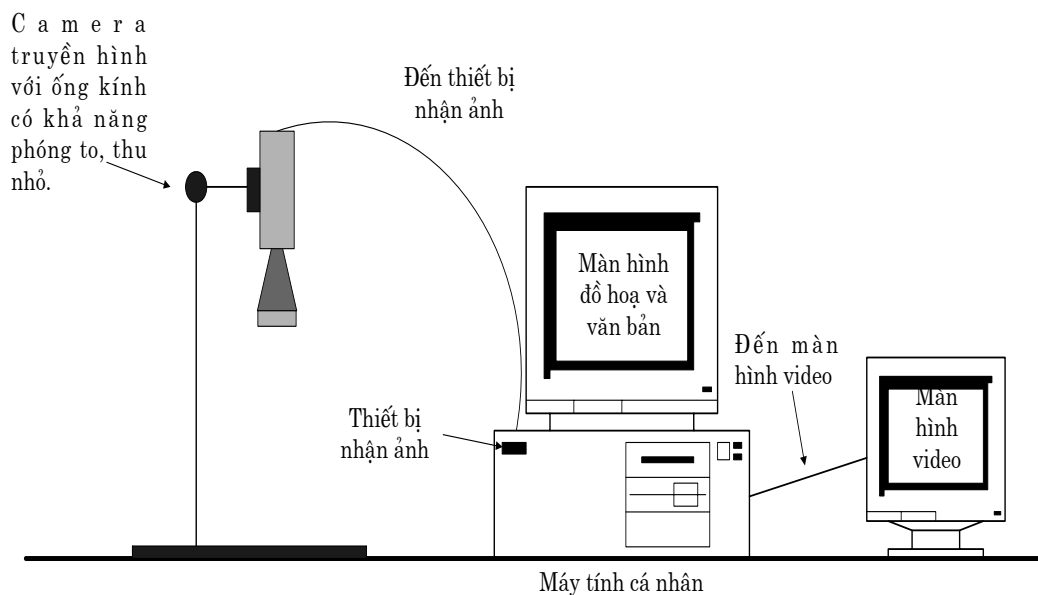
**Hình 1.3 Các thành phần chính của hệ thống xử lý ảnh.**

Theo quan điểm của *quy trình* xử lý, chúng ta đã thể hiện các khối cơ bản trên Hình 1.1, các khối chi tiết và luồng thông tin trên Hình 1.2. Theo quan điểm của *hệ thống* xử lý trên máy tính số, hệ thống gồm các đầu đo (thu nhận ảnh); bộ số hóa; máy tính số; Bộ hiển thị; Bộ nhớ. Các thành phần này không nhắc lại ở đây (đọc thêm giáo trình cấu trúc máy tính).

Một hệ thống xử lý ảnh cơ bản có thể gồm: máy tính cá nhân kèm theo vi mạch chuyên đổi đồ họa VGA hoặc SVGA, đĩa chứa các ảnh dùng để kiểm tra các thuật toán và một màn hình có hỗ trợ VGA hoặc SVGA. Nếu điều kiện cho phép, nên có một hệ thống như Hình 1.4. bao gồm

một máy tính PC kèm theo thiết bị xử lý ảnh. Nối với cổng vào của thiết bị thu nhận ảnh là một video camera, và cổng ra nối với một màn hình. Thực tế, phần lớn các nghiên cứu của chúng ta được đưa ra trên ảnh mức xám (ảnh đen trắng). Bởi vậy, hệ thống sẽ bao gồm một thiết bị xử lý ảnh đen trắng và một màn hình đen trắng.

Ảnh mức xám được áp dụng trong nhiều lĩnh vực như sinh vật học hoặc trong công nghiệp. Thực tế chỉ ra rằng bất kỳ ứng dụng nào trên ảnh, mức xám cũng ứng dụng được trên ảnh màu. Với lý do đó, hệ thống ban đầu nên chỉ bao gồm các thiết bị thu nhận và hiển thị ảnh đen trắng. Với ảnh màu, nên sử dụng một hệ thống mới như Hình 1.3, trừ trường hợp bạn cần một camera TV màu và một màn hình đa tần số (ví dụ như NEC MultiSync, Sony Multiscan, hoặc Mitsubishi Diamond Scan) để hiển thị ảnh màu. Nếu khả năng hạn chế, có thể dùng PC kèm theo vi mạch VGA và màn hình VGA, để dựng ảnh được.



**Hình 1.4 Một hệ thống xử lý ảnh.**

## 1.3 NHỮNG VẤN ĐỀ KHÁC TRONG XỬ LÝ ẢNH

### 1.3.1 Biến đổi ảnh (Image Transform)

Trong xử lý ảnh do số điểm ảnh lớn các tính toán nhiều (độ phức tạp tính toán cao) đòi hỏi dung lượng bộ nhớ lớn, thời gian tính toán lâu. Các phương pháp khoa học kinh điển áp dụng cho xử lý ảnh hầu hết khó khả thi. Người ta sử dụng các phép toán tương đương hoặc biến đổi sang miền xử lý khác để dễ tính toán. Sau khi xử lý dễ dàng hơn được thực hiện, dùng biến đổi ngược để đưa về miền xác định ban đầu, các biến đổi thường gặp trong xử lý ảnh gồm:

- Biến đổi Fourier, Cosin, Sin
- Biến đổi (mô tả) ảnh bằng tích chập, tích Kronecker (theo xử lý số tín hiệu [3])
- Các biến đổi khác như KL (Karhunen Loeve), Hadamard

Một số các công cụ sắc xuất thông kê cũng được sử dụng trong xử lý ảnh. Do khuôn khổ tài liệu hướng dẫn có hạn, sinh viên đọc thêm các tài liệu [1, 2, 3, 4, 5] để nắm được các phương pháp biến đổi và một số phương pháp khác được nêu ở đây.

### 1.3.2 Nén ảnh

Ảnh dù ở dạng nào vẫn chiếm không gian nhớ rất lớn. Khi mô tả ảnh người ta đã đưa kỹ thuật nén ảnh vào. Các giai đoạn nén ảnh có thể chia ra thế hệ 1, thế hệ 2. Hiện nay, các chuẩn MPEG được dùng với ảnh đang phát huy hiệu quả. Một số phương pháp, thuật toán nén đề cập ở các chương 7.

### CÂU HỎI ÔN TẬP

1. Trình bày các thành phần và lưu đồ thông tin giữa các khối trong quá trình xử lý ảnh.
2. Nêu khái niệm và định nghĩa điểm ảnh.
3. Thế nào là độ phân giải ảnh, cho ví dụ?
4. Trình bày định nghĩa mức xám, cho ví dụ.
5. Nêu quan hệ giữa các điểm ảnh.
6. Trình bày về khoảng cách đo và phân loại khoảng cách giữa các điểm ảnh.
7. Nêu ý nghĩa của các phép biến đổi ảnh, liệt kê một số phép biến đổi và cho ví dụ.

## CHƯƠNG 2: THU NHẬN ẢNH

**Học xong phần này sinh viên có thể nắm được:**

1. Các thiết bị thu nhận ảnh;
2. Lấy mẫu và lượng tử hoá
3. Một số phương pháp biểu diễn ảnh
4. Các định dạng ảnh cơ bản
5. Kỹ thuật tái nhận ảnh
6. Khái niệm ảnh đen trắng và ảnh màu

### 2.1 CÁC THIẾT BỊ THU NHẬN ẢNH

Hai thành phần cho công đoạn này là linh kiện nhạy với phổ năng lượng điện từ trường, loại thứ nhất tạo tín hiệu điện ở đầu ra tỷ lệ với mức năng lượng mà bộ cảm biến (đại diện là camera); loại thứ hai là bộ số hoá.

#### 2.1.1 Bộ cảm biến ảnh

Máy chụp ảnh, camera có thể ghi lại hình ảnh (phim trong máy chụp, vidicon trong camera truyền hình). Có nhiều loại máy cảm biến (Sensor) làm việc với ánh sáng nhìn thấy và hồng ngoại như: Micro Densitometers, Image Dissector, Camera Divicon, linh kiện quang điện bằng bán dẫn. Các loại cảm biến bằng chụp ảnh phải số hoá là phim âm bản hoặc chụp ảnh. Camera divicon và linh kiện bán dẫn quang điện có thể cho ảnh ghi trên băng từ có thể số hoá. Trong Micro Densitometer phim và ảnh chụp được gắn trên mặt phẳng hoặc cuộn quang trống. Việc quét ảnh thông qua tia sáng (ví dụ tia Laser) trên ảnh đồng thời dịch chuyển mặt phim hoặc quang trống tương đối theo tia sáng. Trường hợp dùng phim, tia sáng đi qua phim.

Bây giờ chúng ta đề cập đến tất cả các khối trong hệ thống.

#### **Thiết bị nhận ảnh.**

Chức năng của thiết bị này là số hóa một băng tần số cơ bản của tín hiệu truyền hình cung cấp từ một camera, hoặc từ một đầu máy VCR. Ảnh số sau đó được lưu trữ trong bộ đệm chính. Bộ đệm này có khả năng được địa chỉ hóa (nhờ một PC) đến từng điểm bằng phần mềm. Thông thường thiết bị này có nhiều chương trình con điều khiển để có thể lập trình được thông qua ngôn ngữ C. Khi mua một thiết bị cần chú ý các điểm sau:

1. Thiết bị có khả năng số hóa ảnh ít nhất 8 bit (256 mức xám) và ảnh thu được phải có kích thước ít nhất là 512×512 điểm hoặc hơn.
2. Thiết bị phải chứa một bộ đệm ảnh để lưu trữ một hoặc nhiều ảnh có độ phân giải 512×512 điểm ảnh.
3. Thiết bị phải được kèm một bộ đầy đủ thư viện các chương trình con có khả năng giao diện với các chương trình C viết bằng Turbo C hoặc Microsoft C.
4. Sổ tay hướng dẫn sử dụng phải được kèm theo, gồm cả dạng chứa trên đĩa và khi in.

5. Một số thiết bị cho phép tùy chọn sử dụng cả hai chế độ văn bản và đồ họa trên cùng một màn hình hoặc hai màn hình riêng biệt. Mặc dù chi tiết này là không cần thiết, nhưng nó sẽ rất có giá trị trong trường hợp bị giới hạn về không gian lắp đặt hoặc khả năng tài chính.

### **Camera.**

Tổng quát có hai kiểu camera: kiểu camera dùng đèn chân không và kiểu camera chỉ dùng bán dẫn. Đặc biệt là trong lĩnh vực này, camera bán dẫn thường hay được dùng hơn camera đèn chân không. Camera bán dẫn cũng được gọi là CCD camera do dùng các thanh ghi dịch đặc biệt gọi là thiết bị gộp (Charge-Coupled Devices- CCDs). Các CCD này chuyển các tín hiệu ảnh sang từ bộ cảm nhận ánh sáng hỗ trợ ở phía trước camera thành các tín hiệu điện mà sau đó được mã hóa thành tín hiệu TV. Loại camera chất lượng cao cho tín hiệu ít nhiễu và có độ nhạy cao với ánh sáng. Khi chọn camera cần chú ý đến các thấu kính từ 18 đến 108 mm. Sau đây là danh sách các nhà sản xuất:

1. Pulnix America Inc, 770 Lucerne Drive, Sunnyvale, CA 84086. Tel. 408-773-1550; fax 408-737-2966.
2. Sony Corp. of America, Component Products Co., 10833 Valley View St., Cypress, CA 90630. Fax 714-737-4285.
3. Parasonic, industrial camera division: 201-329-6674.
4. JVC Professional: 1-800-JVC-5825.

### **Màn hình video.**

Một số nhà sản xuất (như Sony) sản xuất các loại màn hình đen trắng chất lượng cao. Nên sử dụng loại màn hình chất lượng cao, vì màn hình chất lượng thấp có thể làm bạn nhầm lẫn kết quả. Một màn hình 9 inch là đủ cho yêu cầu làm việc. Để hiển thị ảnh màu, nên dùng một màn hình đa hệ.

### **Máy tính.**

Cần có một máy tính P4 hoặc cấu hình cao hơn. Để chắc chắn, các máy này phải có sẵn các khe cắm cho phần xử lý ảnh. Các chương trình thiết kế và lọc ảnh có thể chạy trên bất kỳ hệ thống nào. Các chương trình con hiển thị ảnh dựng vi mạch VGA và có sẵn trên đĩa kèm theo. Các chương trình con hiển thị ảnh cũng hỗ trợ cho hầu hết các vi mạch SVGA.

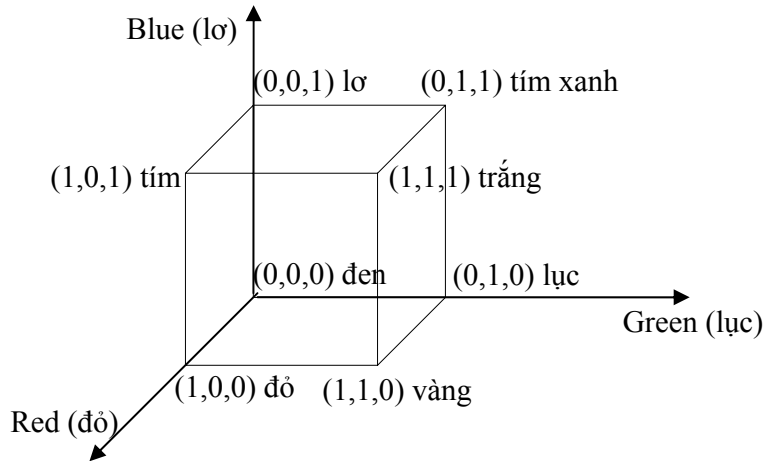
## **2.1.2 Hệ tọa độ màu**

### **a) Khái niệm**

Tổ chức quốc tế về chuẩn hóa màu CIE (Commission Internationale d'Eclairage) đưa ra một số chuẩn để biểu diễn màu. Các hệ này có các chuẩn riêng. Hệ chuẩn màu CIE-rgb dùng 3 màu cơ bản R, G, B và ký hiệu  $RGB_{CIE}$  để phân biệt với các chuẩn khác. Như đã nêu trên, một màu là tổ hợp của các màu cơ bản theo một tỷ lệ nào đó. Như vậy, mỗi pixel ảnh màu ký hiệu  $P_x$ , được viết: ( $T$ : trong công thức dưới đây là ký hiệu chuyển vị)

$$P_x = [red, green, blue]^T$$

Người ta dùng hệ tọa độ ba màu R-G-B (tương ứng với hệ tọa độ x-y-z) để biểu diễn màu như sau:



**Hình 2.1 Hệ tọa độ RGB**

Trong cách biểu diễn này ta có công thức:

$$\text{đỏ} + \text{lục} + \text{lơ} = 1$$

Công thức này gọi là công thức Maxwell. Trong hình trên, tam giác tạo bởi ba đường đứt đoạn gọi là tam giác Maxwell. Màu trắng trong hệ tọa độ này được tính bởi:

$$\text{trắng}_{CIE} = (\text{đỏ}_{CIE} + \text{lục}_{CIE} + \text{lơ}_{CIE}) = 1$$

**b) Biến đổi hệ tọa độ màu**

Hệ tọa độ màu do CIE đề xuất có tác dụng như một hệ quy chiếu và không biểu diễn hết các màu. Trên thực tế, phụ thuộc vào các ứng dụng khác nhau người ta đưa ra các hệ biểu diễn màu khác nhau. Thí dụ:

- Hệ NTSC: dùng 3 màu R, G, B áp dụng cho màn hình màu, ký hiệu  $RGB_{NTSC}$ ;
- Hệ CMY (Cyan Magenta Yellow): thường dùng cho in ảnh màu;
- Hệ YIQ: cho truyền hình màu.

Việc chuyển đổi giữa các không gian biểu diễn màu được thực hiện theo nguyên tắc sau:

Nếu gọi  $\chi$  là không gian biểu diễn các màu ban đầu;  $\chi'$  không gian biểu diễn màu mới

$A$  là ma trận biểu diễn phép biến đổi. Ta có quan hệ sau:

$$\chi' = A\chi$$

Ví dụ, biến đổi hệ tọa độ màu  $RGB_{CIE}$  sang hệ tọa độ màu  $RGB_{NTSC}$  ta có các véc tơ tương ứng:

$$P_x = \begin{bmatrix} R_{CIE} \\ G_{CIE} \\ B_{CIE} \end{bmatrix} \text{ và } P_{x'} = \begin{bmatrix} R_{NTSC} \\ G_{NTSC} \\ B_{NTSC} \end{bmatrix}$$

Công thức chuyển đổi được viết dưới dạng ma trận:

$$\begin{bmatrix} R_{CIE} \\ G_{CIE} \\ B_{CIE} \end{bmatrix} = \begin{bmatrix} 1.167 & -0.146 & -0.151 \\ 0.114 & 0.753 & 0.159 \\ -0.001 & 0.059 & 1.128 \end{bmatrix} \begin{bmatrix} R_{NTSC} \\ G_{NTSC} \\ B_{NTSC} \end{bmatrix}$$

Một số các biến đổi của các hệ tọa độ màu khác, học viên có thể tham khảo các tài liệu [1, 3, 7].

## 2.2 LẤY MẪU VÀ LƯỢNG TỬ HÓA

### 2.2.1 Giới thiệu

Một ảnh  $g(x, y)$  ghi được từ Camera là ảnh liên tục tạo nên mặt phẳng hai chiều. Ảnh cần chuyển sang dạng thích hợp để xử lý bằng máy tính. Phương pháp biến đổi một ảnh (hay một hàm) liên tục trong không gian cũng như theo giá trị thành dạng số rời rạc được gọi là số hoá ảnh. Việc biến đổi này có thể gồm hai bước:

Bước 1: Đo giá trị trên các khoảng không gian gọi là lấy mẫu

Bước 2: Ánh xạ cường độ (hoặc giá trị) đo được thành một số hữu hạn các mức rời rạc gọi là lượng tử hoá.

### 2.2.2 Lấy mẫu

Lấy mẫu là một quá trình, qua đó ảnh được tạo nên trên một vùng có tính liên tục được chuyển thành các giá trị rời rạc theo tọa độ nguyên. Quá trình này gồm 2 lựa chọn:

- Một là: khoảng lấy mẫu.
- Hai là: cách thể hiện dạng mẫu.

Lựa chọn thứ nhất được đảm bảo nhờ lý thuyết lấy mẫu của Shannon. Lựa chọn thứ hai liên quan đến độ đo (Metric) được dùng trong miền rời rạc.

#### Khoảng lấy mẫu (Sampling Interval)

Ảnh lấy mẫu có thể được mô tả như việc lựa chọn một tập các vị trí lấy mẫu trong không gian hai chiều liên tục. Đầu tiên mô tả qua quá trình lấy mẫu một chiều với việc sử dụng hàm delta:

$$\delta(x - x_0) = \begin{cases} 0 & \text{khi } x \neq 0 \\ \infty & \text{khi } x = 0 \end{cases} \quad (2-1)$$

$$\int_{-\infty}^{\infty} \delta(x - x_0) dx = \int_{x_0^-}^{x_0^+} \delta(x - x_0) dx = 1 \quad (2-2)$$

Tiếp theo chúng ta định nghĩa hàm răng lược với các khoảng  $\Delta x$  như sau:

$$Comb(x) = \sum_{r=-\infty}^{\infty} \delta(x - r\Delta x) \quad (2-3)$$

với  $r$  là số nguyên,  $\Delta x$ : khoảng lấy mẫu

Như vậy, hàm răng lược là chuỗi các xung răng lược từ  $(-\infty$  đến  $+\infty)$ . Giả sử hàm một chiều  $g(x)$  được mô tả (gần đúng) bằng  $g(r\Delta x)$  tức là:

$$g(x) \approx g(r\Delta x) \quad (2-4)$$

Khi đó tín hiệu lấy mẫu được mô hình hoá

$$g_s(x) = g(x)comb(x) = g(x) \sum_{r=-\infty}^{\infty} \delta(x - r\Delta x) \quad (2-5)$$



hoặc tương đương

$$g_s(x) = \sum_{r=-\infty}^{\infty} g(r\Delta x)\delta(x - r\Delta x) \quad (2-6)$$

Trong thực tế,  $r$  không thể tính được trong khoảng vô hạn (từ  $-\infty$  đến  $+\infty$ ) mà là một số lượng  $N\Delta x$  mẫu lớn cụ thể. Như vậy, để đơn giản có thể nói hàm liên tục  $g(x)$  có thể biểu diễn trên một miền với độ dài  $N\Delta x$  mẫu thành chuỗi như sau:

$$g(x) \approx \{g(0), g(\Delta x), g(2\Delta x), \dots, g((N-1)\Delta x)\} \quad (2-7)$$

**Chú ý 1:** Khoảng lấy mẫu (Sampling Interval)  $\Delta x$  là một tham số cần phải được chọn đủ nhỏ, thích hợp, nếu không tín hiệu thật không thể khôi phục lại được từ tín hiệu lấy mẫu.

**Chú ý 2:** Từ lý thuyết về xử lý tín hiệu số [5], (2-6) là tích chập trong miền không gian  $x$ . Mặt khác (2-6) tương đương với tích chập trong miền tần số  $\omega$  tức là biến đổi Fourier của  $g_s(x)$  là  $G_s(\omega_x)$ .

$$G_s(\omega_x) = \frac{1}{\Delta x} \sum_{k=-\infty}^{\infty} G(\omega_x - \frac{k}{\Delta x}) \quad (2-8)$$

trong đó  $\omega_x$  là giá trị tần số ứng với giá trị  $x$  trong miền không gian.

Điều kiện khôi phục ảnh lấy mẫu về ảnh thật được phát biểu từ định lý lấy mẫu của Shannon.

#### Định lý lấy mẫu của Shannon [5]

Giả sử  $g(x)$  là một hàm giới hạn giải (Band Limited Function) và biến đổi Fourier của nó là  $G(\omega_x) = 0$  đối với các giá trị  $\omega_x > W_x$ . Khi đó  $g(x)$  có thể được khôi phục lại từ các mẫu được tạo tại các khoảng  $\Delta x$  đều đặn. Tức là

$$\Delta x \leq \frac{1}{2\omega_x} \quad (2-9)$$

Định lý lấy mẫu của Shannon có thể mở rộng cho không gian hai chiều. Hàm răng lược hai chiều khi đó được xác định:

$$comb(x,y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta(x - n\Delta x)(y - m\Delta y) \quad (2-10)$$

Hàm lấy mẫu hai chiều thu được:

$$g_s(x, y) = g(x, y)comb(x, y) = g(x, y) \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta(x - n\Delta x, y - m\Delta y) \quad (2-11)$$

và  $\Delta x, \Delta y$  được chọn thoả mãn các điều kiện tương ứng theo định lý lấy mẫu của Shannon khi đó sẽ:

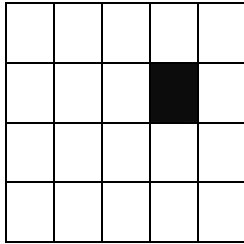
$$\Delta x \leq \frac{1}{2\omega_x}; \Delta y \leq \frac{1}{2\omega_y} \quad (2-12)$$

Tương tự như không gian một chiều, một tín hiệu ảnh hai chiều  $g(x,y)$  có thể xấp xỉ trong khoảng  $[N, M]$  có thể được ước lượng như sau :

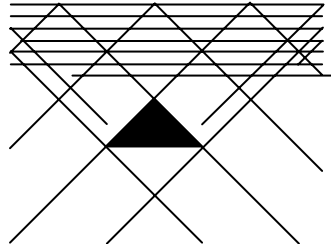
$$g(x, y) \approx g(n\Delta x, m\Delta y) = \{g(0,0), g(0,1), \dots, g(0, N-1), g(1,0), g(1,1), \dots, g(1, N-1), \dots, g(N-1, M-1)\} \quad (2-13)$$

**Các dạng lấy mẫu (Tesselation)**

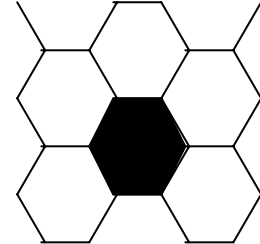
Dạng lấy mẫu (Tesselation) điểm ảnh là cách bài trí các điểm mẫu trong không gian hai chiều. Một số dạng mẫu điểm ảnh được cho là dạng chữ nhật, tam giác, lục giác. Mỗi một mẫu, ngoài việc thể hiện hình dáng còn cho biết đặc điểm liên thông của chúng. Ví dụ, mẫu chữ nhật có liên thông 4 hoặc 8 (nói về các mẫu liền kề); mẫu lục giác có liên thông 6; mẫu tam giác có liên thông 3 hoặc 6.



a) Mẫu điểm ảnh chữ nhật



b) Mẫu điểm ảnh tam giác



c) Mẫu điểm ảnh lục giác

**Hình 2.2 Các dạng mẫu điểm ảnh**

Cần chú ý rằng tài liệu này chỉ xét các mẫu điểm ảnh hình chữ nhật, đặc biệt là dạng hình vuông. Nhiều trường hợp ứng dụng có dùng đến các các mẫu tam giác hoặc lục giác.

**2.2.3 Lượng tử hóa**

Lượng tử hoá là một quá trình lượng hoá tín hiệu thật dùng chung cho các loại xử lý tín hiệu trên cơ sở máy tính. Vấn đề này đã được nghiên cứu kỹ lưỡng và có nhiều lời giải lý thuyết dưới nhiều giả định của các nhà nghiên cứu như Panter và Dite (1951), Max (1960), Panter (1965) [5].

Các giá trị lấy mẫu  $Z$  là một tập các số thực từ giá trị  $Z_{min}$  đến lớn nhất  $Z_{max}$ . Mỗi một số trong các giá trị mẫu  $Z$  cần phải biến đổi thành một tập hữu hạn số bit để máy tính lưu trữ hoặc xử lý.

**Định nghĩa:** Lượng tử hoá là ánh xạ từ các số thực mô tả giá trị lấy mẫu thành một giải hữu hạn các số thực. Nói cách khác, đó là quá trình số hoá biên độ.



**Hình 2.3 Khuông lượng tử theo L mức xám.**



### 2.3.3 Mã tứ phân

Theo phương pháp mã tứ phân, một vùng ảnh coi như bao kín một hình chữ nhật. Vùng này được chia làm 4 vùng con (Quadrant). Nếu một vùng con gồm toàn điểm đen (1) hay toàn điểm trắng (0) thì không cần chia tiếp. Trong trường hợp ngược lại, vùng con gồm cả điểm đen và trắng gọi là vùng không đồng nhất, ta tiếp tục chia thành 4 vùng con tiếp và kiểm tra tính đồng nhất của các vùng con đó. Quá trình chia dừng lại khi mỗi vùng con chỉ chứa thuần nhất điểm đen hoặc điểm trắng. Quá trình đó tạo thành một cây chia theo bốn phần gọi là cây tứ phân. Như vậy, cây biểu diễn ảnh gồm một chuỗi các ký hiệu *b* (black), *w* (white) và *g* (grey) kèm theo ký hiệu mã hóa 4 vùng con. Biểu diễn theo phương pháp này ưu việt hơn so với các phương pháp trên, nhất là so với mã loạt dài. Tuy nhiên, để tính toán số đo các hình như chu vi, mô men là tương đối khó khăn.

## 2.4 CÁC ĐỊNH DẠNG ẢNH CƠ BẢN

### 2.4.1 Khái niệm chung

Ảnh thu được sau quá trình số hóa thường được lưu lại cho các quá trình xử lý tiếp theo hay truyền đi (xem lại Hình 1.2). Trong quá trình phát triển của kỹ thuật xử lý ảnh, tồn tại nhiều định dạng ảnh khác nhau từ ảnh đen trắng (với định dạng IMG), ảnh đa cấp xám cho đến ảnh màu: (BMP, GIF, JPEG...). Tuy các định dạng này khác nhau, song chúng đều tuân theo một cấu trúc chung nhất. Nhìn chung, một tệp ảnh bất kỳ thường bao gồm 3 phần:

- Mào đầu tệp (Header)
- Dữ liệu nén (Data Compression)
- Bảng màu (Palette Color)

#### a) Mào đầu tệp:

Mào đầu tệp là phần chứa các thông tin về kiểu ảnh, kích thước, độ phân giải, số bit dùng cho 1 pixel, cách mã hóa, vị trí bảng màu...

#### b) Dữ liệu nén:

Số liệu ảnh được mã hóa bởi kiểu mã hóa chỉ ra trong phần Header.

#### c) Bảng màu:

Bảng màu không nhất thiết phải có ví dụ khi ảnh là đen trắng. Nếu có, bảng màu cho biết số màu dùng trong ảnh và bảng màu được sử dụng để hiển thị màu của ảnh. Một số các định dạng khác, cấu hình, đặc trưng của từng định dạng và các tham số, học viên có thể tham khảo thêm tài liệu khác (ví dụ [1]).

### 2.4.2 Quy trình đọc một tệp ảnh

Trong quá trình xử lý ảnh, đầu tiên phải tiến hành đọc tệp ảnh và chuyển vào bộ nhớ của máy tính dưới dạng ma trận số liệu ảnh. Khi lưu trữ dưới dạng tệp, ảnh là một khối gồm một số các byte. Để đọc đúng tệp ảnh ta cần hiểu ý nghĩa các phần trong cấu trúc của tệp ảnh như đã nêu trên. Trước tiên, ta cần đọc phần mào đầu (Header) để lấy các thông tin chung và thông tin điều khiển. Việc đọc này sẽ dừng ngay khi ta không gặp được chữ ký (Chữ ký ở đây thường được hiểu

là một mã chỉ ra định dạng ảnh và đời (version) của nó) mong muốn. Dựa vào thông tin điều khiển, ta xác định được vị trí bảng màu và đọc nó vào bộ nhớ. Cuối cùng, ta đọc phần dữ liệu nén.

Sau khi đọc xong các khối dữ liệu ảnh vào bộ nhớ ta tiến hành nén dữ liệu ảnh. Căn cứ vào phương pháp nén chỉ ra trong phần Header ta giải mã được ảnh. Cuối cùng là khâu hiện ảnh. Dựa vào số liệu ảnh đã giải nén, vị trí và kích thước ảnh, cùng sự trợ giúp của bảng màu ảnh được hiện lên trên màn hình.

## 2.5 CÁC KỸ THUẬT TÁI HIỆN ẢNH

### 2.5.1 Kỹ thuật chụp ảnh

Phương pháp sao chụp ảnh là phương pháp đơn giản, giá thành thấp, chất lượng cao. Sau bước chụp là kỹ thuật phòng tối nhằm tăng cường ảnh như mong muốn. Ví dụ kỹ thuật phòng tối như: phóng đại ảnh, thu nhỏ ảnh..., tùy theo ứng dụng. Kỹ thuật chụp ảnh màn hình màu khá đơn giản. Nó bao gồm các bước sau :

- 1) Đặt camera trong phòng tối, cách màn hình khoảng 10 feet (1 feet=0,3048m)
- 2) Mở ống kính để phẳng mặt cong màn hình, do vậy ảnh sẽ dàn đều hơn
- 3) Tắt phím sáng tối (Brightness) và phím tương phản (Contrast) của màn hình để tạo độ rõ cho ảnh. Các màu chói, cường độ cao trên ảnh sẽ giảm đi.
- 4) Đặt tốc độ ống kính từ 1/8 đến 1/2 giây.

### 2.5.2 Kỹ thuật in ảnh

Người ta dùng kỹ thuật nửa cường độ để thể hiện ảnh trên sách báo, tạp chí. Theo kỹ thuật này, một ảnh tạo nên bởi một chuỗi các điểm in trên giấy. Thực chất, mỗi điểm ảnh thường gồm một hình vuông trắng bao quanh một chấm đen. Do vậy, nếu chấm đen càng lớn ảnh sẽ càng xảm màu. Màu xám có thể coi như chấm đen chiếm nửa vùng trắng. Vùng trắng là vùng gồm một chùm các điểm ảnh có rất ít hoặc không có chấm đen.

Từ đặc điểm cảm nhận của mắt người, sự thay đổi cường độ chấm đen trong các phần tử ảnh trắng tạo nên mô phỏng của một ảnh liên tục. Như vậy, mắt người cảm nhận từ một ảnh mà màu biến đổi từ đen qua xám rồi đến trắng. Tổng số cường độ duy nhất hiện diện sẽ xác định các kích thước khác nhau của chấm đen. Thông thường, báo ảnh tạo ảnh nửa cường độ với độ phân giải từ 60 đến 80 dpi (dot per inches : số điểm ảnh trên một inch), sách có thể in đến 150 dpi.

Tuy nhiên, các máy in ghép nối với máy tính không có khả năng sắp xếp các chấm đen có kích thước khác nhau của ảnh. Do đó, người ta dùng một số kỹ thuật biến đổi như: phân ngưỡng, chọn mẫu, dithering (dithering sẽ định nghĩa dưới đây).

#### a) Phân ngưỡng:

Kỹ thuật này đặt ngưỡng để hiển thị các tông màu liên tục. Các điểm trong ảnh được so sánh với ngưỡng định trước. Giá trị của ngưỡng sẽ quyết định điểm có được hiển thị hay không. Do vậy ảnh kết quả sẽ mất đi một số chi tiết. Có nhiều kỹ thuật chọn ngưỡng áp dụng cho các đối tượng khác nhau :

- *Hiện thị 2 màu*: chỉ dùng ảnh đen trắng có 256 mức xám. Bản chất của phương pháp này là chọn ngưỡng dựa trên lược đồ mức xám của ảnh. Để đơn giản có thể lấy ngưỡng với giá trị là 127. Như vậy :

$$u(m, n) = \begin{cases} 1 & \text{khi } h(m, n) < 127 \\ 0 & \text{khác} \end{cases}$$

trong đó  $u(m, n)$  là mức xám tại tọa độ  $(m, n)$ .

Nhìn chung kỹ thuật này khó chấp nhận vì ảnh mất khá nhiều chi tiết.

- *Hiện thị 4 màu*: hiện 4 màu để khắc phục nhược điểm của kỹ thuật hiện thị 2 màu. Một ví dụ của Bảng mã 4 màu được cho ở Bảng 1.1.

**Bảng 1.1 Bảng mã 4 màu**

Mã màu	Màn hình monochrome (đơn sắc)	Màn hình màu
0	Đen	Đen
1	Xám đậm	Đỏ
2	Xám nhạt	Xanh
3	Trắng	Vàng

### b) Kỹ thuật chọn theo mẫu

Kỹ thuật này sử dụng một nhóm các phần tử trên thiết bị ra (máy in chẳng hạn) để biểu diễn một pixel trên ảnh nguồn. Các phần tử của nhóm quyết định độ sáng tối của cả nhóm. Các phần tử này mô phỏng các chấm đen trong kỹ thuật nửa cường độ. Nhóm thường được chọn có dạng ma trận vuông. Nhóm  $n \times n$  phần tử sẽ tạo nên  $n^2 + 1$  mức sáng. Ma trận mẫu thường được chọn là ma trận Rylander. Ma trận Rylander cấp 4 có dạng như Bảng 1.2.

**Bảng 1.2. Ma trận Rylander cấp 4**

0	8	2	10
4	12	6	14
3	11	1	9
7	15	5	13

Việc chọn kích thước của nhóm như vậy sẽ làm giảm độ mịn của ảnh. Vì vậy kỹ thuật này chỉ áp dụng trong trường hợp mà độ phân giải của thiết bị ra lớn hơn độ phân giải của ảnh nguồn. Thí dụ: thiết bị ra có độ phân giải 640x480 khi sử dụng nhóm có kích thước 4x4 sẽ chỉ còn 160x120.

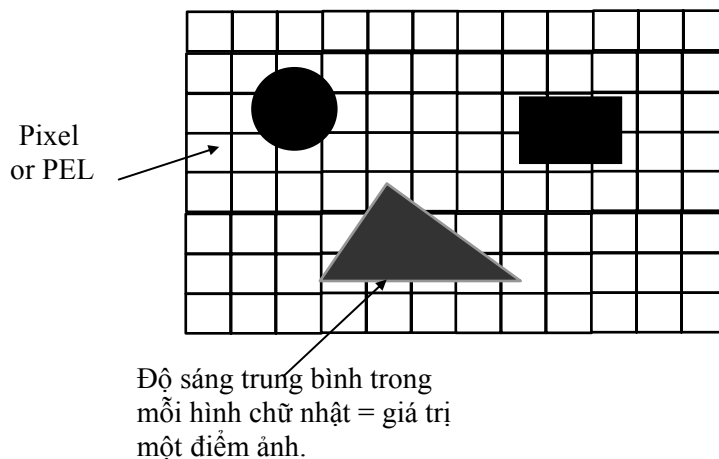
### c) Kỹ thuật Dithering

Dithering là việc biến đổi một ảnh đa cấp xám (nhiều mức sáng tối) sang ảnh nhị phân (hai mức sáng tối). Kỹ thuật Dithering được áp dụng để tạo ra ảnh đa cấp sáng khi độ phân giải nguồn và đích là như nhau. Kỹ thuật này sử dụng một ma trận mẫu gọi là ma trận Dither. Ma trận này gần giống như ma trận Rylander.

Để tạo ảnh, mỗi phần tử của ảnh gốc sẽ được so sánh với phần tử tương ứng của ma trận Dither. Nếu lớn hơn, phần tử ở đầu ra sẽ sáng và ngược lại. Học viên có thể đọc thêm ở [1] để hiểu chi tiết hơn về kỹ thuật này.

## 2.6 KHÁI NIỆM ẢNH ĐEN TRẮNG VÀ ẢNH MÀU

Ảnh có thể biểu diễn dưới dạng tín hiệu tương tự hoặc tín hiệu số. Trong biểu diễn số của các ảnh đa mức xám, một ảnh được biểu diễn dưới dạng một ma trận hai chiều. Mỗi phần tử của ma trận biểu diễn cho mức xám hay cường độ của ảnh tại vị trí đó.



*Hình 2.3 Biểu diễn mức xám của ảnh số.*

Trong Hình 2.3, một lưới chia ô vuông tương tượng được đặt lên ảnh. Độ lớn mỗi ô vuông của lưới xác định kích thước của một điểm ảnh. Mức xám của một điểm được tính bằng cường độ sáng trung bình tại mỗi ô vuông này. Mất lưới càng nhỏ thì chất lượng ảnh càng cao. Trong kỹ thuật truyền hình tiên tiến, (mục đích là cung cấp cho người xem), hình ảnh cần chất lượng cao với độ phân giải gấp hai lần so với các chuẩn hiện nay.

Trong kỹ thuật tương tự, một bức ảnh thường được biểu diễn dưới dạng các dòng nằm ngang kế tiếp nhau. Mỗi dòng là một tín hiệu tương tự mang theo các thông tin về cường độ sáng dọc theo một đường nằm ngang trong ảnh gốc. Ảnh trên một chiếc TV được hiện lên qua các dòng quét này. Mặc dù thuật ngữ "tương tự" được dùng để mô tả cho các ảnh quét liên tiếp nhưng thực tế ảnh chỉ tương tự dọc theo hướng nằm ngang. Nó là rời rạc khi xét theo hướng dọc và chính vì vậy mà tín hiệu ảnh là tín hiệu lai nửa tương tự, nửa số.

Một máy truyền hình được thiết kế để thu tín hiệu truyền hình mã hoá theo tiêu chuẩn NTSC của Mỹ có khả năng hiển thị xấp xỉ 525 dòng. Công nghệ truyền hình tiên bộ nỗ lực để cung cấp cho chúng ta số lượng các dòng gấp hai lần, cho độ phân giải tốt hơn là TV màn ảnh rộng. Một TV có màn ảnh lớn hơn 28 inch được coi là một TV có màn ảnh rộng. Một điều cần chú ý là TV có khả năng hiện một số dòng như nhau cho dù nó là 5 inch hay là 50 inch. Màn ảnh lớn nhất của loại TV dòng quét xen kẽ mà mắt người có khả năng phân biệt được từ khoảng cách thông thường vào khoảng 3 mét.

### 2.6.1 Ảnh đen trắng

Ảnh đen trắng chỉ bao gồm 2 màu: màu đen và màu trắng. Người ta phân mức đen trắng đó thành  $L$  mức. Nếu sử dụng số bit  $B=8$  bit để mã hóa mức đen trắng (hay mức xám) thì  $L$  được xác định :

$$L=2^B \text{ (trong ví dụ của ta } L=2^8=256 \text{ mức)}$$

Nếu  $L$  bằng 2,  $B=1$ , nghĩa là chỉ có 2 mức: mức 0 và mức 1, còn gọi là *ảnh nhị phân*. Mức 1 ứng với màu sáng, còn mức 0 ứng với màu tối. Nếu  $L$  lớn hơn 2 ta có ảnh đa cấp xám.

Nói cách khác, với ảnh nhị phân mỗi điểm ảnh được mã hóa trên 1 bit, còn với ảnh 256 mức, mỗi điểm ảnh được mã hóa trên 8 bit. Như vậy, với ảnh đen trắng: nếu dùng 8 bit (1 byte) để biểu diễn mức xám, số các mức xám có thể biểu diễn được là 256. Mỗi mức xám được biểu diễn dưới dạng là một số nguyên nằm trong khoảng từ 0 đến 255, với mức 0 biểu diễn cho mức cường độ đen nhất và 255 biểu diễn cho mức cường độ sáng nhất.

Ảnh nhị phân khá đơn giản, các phần tử ảnh có thể coi như các phần tử logic. Ứng dụng chính của nó được dùng theo tính logic để phân biệt đối tượng ảnh với nền hay để phân biệt điểm biên với điểm khác.

### 2.6.2 Ảnh màu

Ảnh màu theo lý thuyết của Thomas là ảnh tổ hợp từ 3 màu cơ bản: đỏ (R), lục (G), lơ (B) và thường thu nhận trên các dải băng tần khác nhau. Với ảnh màu, cách biểu diễn cũng tương tự như với ảnh đen trắng, chỉ khác là các số tại mỗi phần tử của ma trận biểu diễn cho ba màu riêng rẽ gồm: đỏ (red), lục (green) và lam (blue). Để biểu diễn cho một điểm ảnh màu cần 24 bit. 24 bit này được chia thành ba khoảng 8 bit. Mỗi màu cũng phân thành  $L$  cấp màu khác nhau (thường  $L=256$ ). Mỗi khoảng này biểu diễn cho cường độ sáng của một trong các màu chính.

Do đó, để lưu trữ ảnh màu người ta có thể lưu trữ từng màu riêng biệt, mỗi màu lưu trữ như một ảnh đa cấp xám. Do đó, không gian nhớ dành cho một ảnh màu lớn gấp 3 lần một ảnh đa cấp xám cùng kích cỡ.

## CÂU HỎI ÔN TẬP

1. Trình bày các thiết bị thu nhận ảnh .
2. Thế nào là lấy mẫu và lượng tử hóa ảnh?
3. Trình bày phương pháp mã hóa loạt dài.
4. Trình bày các phương pháp mã hóa mã xích, mã tứ phân.
5. Trình bày các phương pháp mã hóa tứ phân.
6. Nêu các định dạng ảnh cơ bản.
7. Trình bày khái niệm ảnh đen trắng và ảnh màu.
8. Mức xám là gì? Kỹ thuật thể hiện mức xám với ảnh đa cấp xám và ảnh màu ?
9. Kỹ thuật tái hiện ảnh là gì? Trình bày các kỹ thuật tái hiện ảnh



## CHƯƠNG 3 : XỬ LÝ NÂNG CAO CHẤT LƯỢNG ẢNH

**Học xong phần này sinh viên có thể nắm được:**

1. Những vấn đề về nâng cao chất lượng ảnh. Các công cụ, các toán tử được sử dụng
2. Phương pháp cải thiện ảnh sử dụng các toán tử điểm
3. Mục đích vai trò của biên ảnh. Phương pháp phát hiện biên cục bộ
4. Ý nghĩa và các công cụ đặc biệt các công cụ toán học dùng trong khôi phục ảnh.
5. Phân biệt sự khác biệt giữa khôi phục và nâng cao chất lượng ảnh

### 3.1 CẢI THIỆN ẢNH SỬ DỤNG CÁC TOÁN TỬ ĐIỂM

Nâng cao chất lượng là bước cần thiết trong xử lý ảnh nhằm hoàn thiện một số đặc tính của ảnh. Nâng cao chất lượng ảnh gồm hai công đoạn khác nhau: tăng cường ảnh và khôi phục ảnh. Tăng cường ảnh nhằm hoàn thiện các đặc tính của ảnh như :

- Lọc nhiễu, hay làm trơn ảnh,
- Tăng độ tương phản, điều chỉnh mức xám của ảnh,
- Làm nổi biên ảnh.

Các thuật toán triển khai việc nâng cao chất lượng ảnh hầu hết dựa trên các kỹ thuật trong miền điểm, không gian và tần số. Toán tử điểm là phép biến đổi đối với từng điểm ảnh đang xét, không liên quan đến các điểm lân cận khác, trong khi đó, toán tử không gian sử dụng các điểm lân cận để quy chiếu tới điểm ảnh đang xét. Một số phép biến đổi có tính toán phức tạp được chuyển sang miền tần số để thực hiện, kết quả cuối cùng được chuyển trở lại miền không gian nhờ các biến đổi ngược.

#### **Khái niệm về toán tử điểm:**

Xử lý điểm ảnh thực chất là biến đổi giá trị một điểm ảnh dựa vào giá trị của chính nó mà không hề dựa vào các điểm ảnh khác. Có hai cách tiếp cận với phương pháp này. Cách thứ nhất dùng một hàm biến đổi thích hợp với mục đích hoặc yêu cầu đặt ra để biến đổi giá trị mức xám của điểm ảnh sang một giá trị mức xám khác. Cách thứ hai là dùng lược đồ mức xám (Gray Histogram). Về mặt toán học, toán tử điểm là một ánh xạ từ giá trị cường độ ánh sáng  $u(m, n)$  tại tọa độ  $(m, n)$  sang giá trị cường độ ánh sáng khác  $v(m, n)$  thông qua hàm  $f(\cdot)$ , tức là:

$$v(m,n) = f(u(m,n)) \quad (3-1)$$

Nói một cách khác, toán tử điểm là toán tử không bộ nhớ, ở đó một mức xám  $u \in [0, N]$  được ánh xạ sang một mức xám  $v \in [0, N] : v = f(u)$ . Ứng dụng chính của các toán tử điểm là biến đổi độ tương phản của ảnh. Ánh xạ  $f$  khác nhau tùy theo các ứng dụng. Các dạng toán tử điểm được giới thiệu cụ thể như sau:

- 1) *Tăng độ tương phản.*

$$f(u) = \begin{cases} \alpha u & \alpha \leq u < a \\ \beta(u - a) + v_a & a \leq u < b \\ \gamma(u - b) + v_b & \beta \leq u < L \end{cases} \quad (3-2)$$

Các cấp độ  $\alpha, \beta, \gamma$  xác định độ tương phản tương đối.  $L$  là số mức xám cực đại.

2) Tách nhiễu và phân ngưỡng

$$f(u) = \begin{cases} 0 & 0 \leq u < a \\ \alpha u & a \leq u < b \\ L & u \geq b \end{cases} \quad (3-3)$$

Trong đó  $a = b = t$  gọi là phân ngưỡng.

3) Biến đổi âm bản.

$$f(u) = L - u \quad \text{tạo âm bản} \quad (3-4)$$

4) Cắt theo mức

$$f(u) = \begin{cases} L & a \leq u \leq b \\ 0 & \neq \end{cases}$$

5) Trích chọn bit.

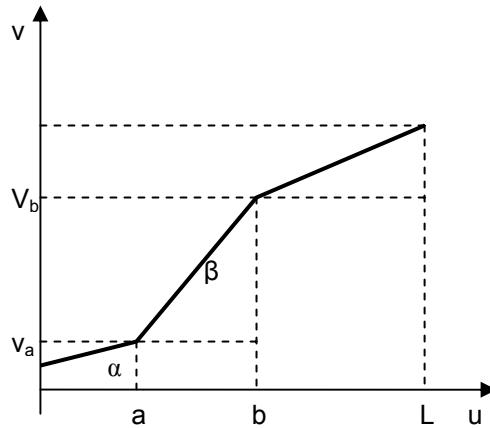
$$f(u) = (i_n - 2i_{n-1})L \quad \text{với } i_n = \text{Int}[it / 2^{n-1}], n=1, 2, \dots, B \quad (3-5)$$

### 3.1.1 Tăng độ tương phản (Stretching Contrast)

Trước tiên cần làm rõ khái niệm độ tương phản. Ảnh số là tập hợp các điểm, mỗi điểm có giá trị độ sáng khác nhau. Ở đây, độ sáng để mắt người dễ cảm nhận ảnh song không phải là quyết định. Thực tế chỉ ra rằng hai đối tượng có cùng độ sáng nhưng đặt trên hai nền khác nhau sẽ cho cảm nhận sáng khác nhau. Như vậy, độ tương phản biểu diễn sự thay đổi độ sáng của đối tượng so với nền. Nói một cách khác, *độ tương phản là độ nổi của điểm ảnh hay vùng ảnh so với nền*. Như vậy, nếu ảnh có độ tương phản kém, ta có thể thay đổi tùy ý theo ý muốn.

Ảnh với độ tương phản thấp có thể do điều kiện sáng không đủ hay không đều, hoặc do tính không tuyến tính hay biến động nhỏ của bộ cảm nhận ảnh. Để điều chỉnh lại độ tương phản của ảnh, cần điều chỉnh lại biên độ trên toàn dải hay trên dải có giới hạn bằng cách biến đổi tuyến tính biên độ đầu vào (dùng hàm biến đổi là hàm tuyến tính) hay phi tuyến (hàm mũ hay hàm lôgarit). Khi dùng hàm tuyến tính các độ dốc  $\alpha, \beta, \gamma$  phải chọn *lớn hơn một trong miền cần dẫn*. Các tham số  $a$  và  $b$  (các cận) có thể chọn khi xem xét lược đồ xám của ảnh. Chú ý, nếu dẫn độ tương phản bằng hàm tuyến tính ta có:

$\alpha = \beta = \gamma = 1$	<i>ảnh kết quả trùng với ảnh gốc</i>
$\alpha, \beta, \gamma > 1$	<i>dãn độ tương phản.</i>
$\alpha, \beta, \gamma < 1$	<i>co độ tương phản</i>



Hình 3.1. Dẫn độ tương phản.

Hàm mũ thường được dùng để dẫn độ tương phản. Hàm có dạng:

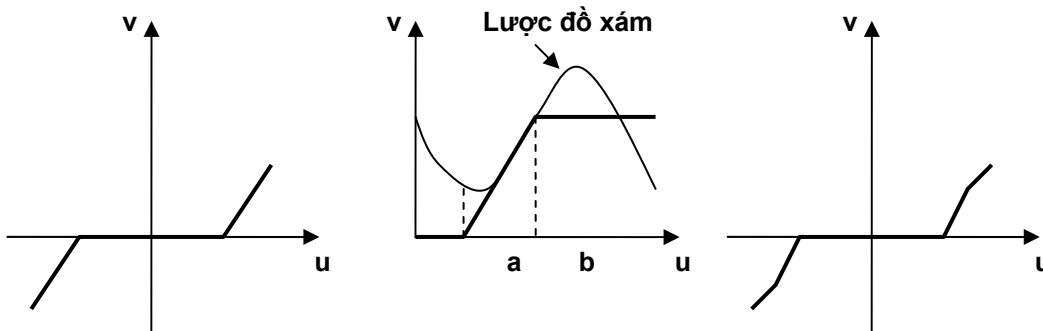
$$f(u) = (X[m, n])^p \quad (3-6)$$

với  $p$  là bậc thay đổi, thường chọn bằng 2.

### 3.1.2 Tách nhiễu và phân ngưỡng

Tách nhiễu là trường hợp đặc biệt của dẫn độ tương phản khi hệ số góc  $\alpha = \gamma = 0$ . Tách nhiễu được ứng dụng có hiệu quả để giảm nhiễu khi biết tín hiệu vào trên khoảng  $[a, b]$ .

Phân ngưỡng là trường hợp đặc biệt của tách nhiễu khi  $a = b = \text{const}$ . Trong trường hợp này, ảnh đầu vào là ảnh nhị phân (có 2 mức). Phân ngưỡng thường dùng trong kỹ thuật in ảnh 2 màu vì ảnh gần nhị phân không cho ảnh nhị phân khi quét ảnh do có nhiễu từ bộ cảm biến và biến đổi của nền ví dụ trường hợp lọc nhiễu của ảnh vân tay.



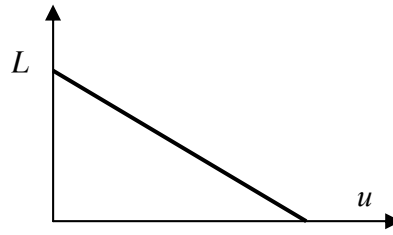
Hình 3.2. Tách nhiễu và phân ngưỡng.

### 3.1.2 Biến đổi âm bản (Digital Negative)

Âm bản nhận được bằng phép biến đổi âm. Phép biến đổi rất có nhiều hữu ích trong các phim ảnh dùng trong các ảnh y học.

$$f(u) = L - u$$

$v$



Hình 3.3. Biến đổi âm bản.

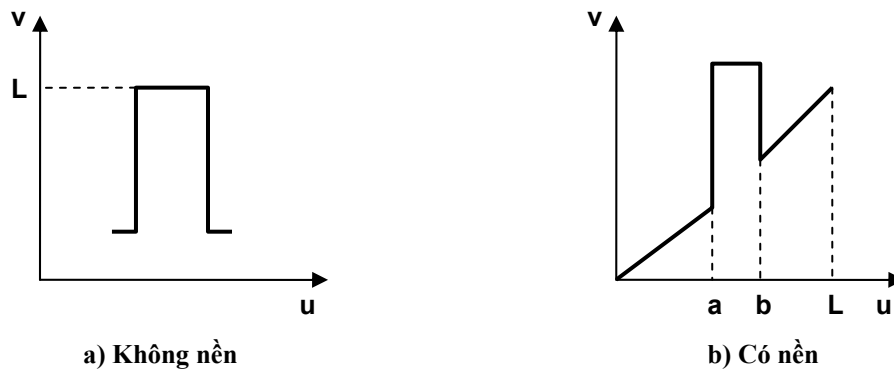
### 3.1.4 Cắt theo mức (Intensity Level Slicing)

Kỹ thuật này dùng 2 phép ánh xạ khác nhau cho trường hợp có nền và không nền

(3.7)

• Có nền: 
$$f(u) = \begin{cases} L & \text{khi } a \leq u \leq b \\ u & \neq \end{cases}$$

• Không nền: 
$$f(u) = \begin{cases} L & \text{khi } a \leq u \leq b \\ 0 & \neq \end{cases} \quad (3-8)$$



Hình 3.4. Kỹ thuật cắt theo mức

Biến đổi này cho phép phân đoạn một số mức xám từ phần còn lại của ảnh. Nó có tác dụng khi nhiều đặc tính khác nhau của ảnh nằm trên nhiều miền mức xám khác nhau.

### 3.1.5 Trích chọn bit (Bit Extraction)

Như đã trình bày trên, mỗi điểm ảnh thường được mã hóa trên  $B$  bit. Nếu  $B=8$  ta có ảnh  $2^8=256$  mức xám (ảnh nhị phân ứng với  $B=1$ ). Trong các bit mã hóa này, người ta chia làm 2 loại: *bit bậc thấp* và *bit bậc cao*. Với bit bậc cao, độ bảo toàn thông tin cao hơn so với bit bậc thấp. Các bit bậc thấp thường biểu diễn nhiễu hay nền. Trong kỹ thuật này, ta có:

$$u = k_1 2^{B-1} + k_2 2^{B-2} + \dots + k_{B-1} 2 + k_B \quad (3-9)$$

Để trích chọn bit có nghĩa nhất: bit thứ  $n$  và hiện thị chúng, dùng biến đổi sau:

$$f(u) = \begin{cases} L & \text{khi } k_n = 1 \\ 0 & \neq \end{cases} \quad (3-10)$$

Dễ thấy:  $k_n = i_n - 2i_{n-1}$ .

### 3.1.6 Trừ ảnh

Trừ ảnh được dùng để tách nhiễu khỏi nền. Người ta quan sát ảnh ở 2 thời điểm khác nhau, so sánh chúng để tìm ra sự khác nhau. Người ta dóng thẳng 2 ảnh rồi trừ đi và thu được ảnh mới. Ảnh mới này chính là sự khác nhau. Kỹ thuật này hay được dùng trong dự báo thời tiết, trong y học.

### 3.1.7 Nén dải độ sáng.

Đôi khi do dải động của ảnh lớn, việc quan sát ảnh không thuận tiện. Cần phải thu nhỏ dải độ sáng lại mà ta gọi là nén dải độ sáng. Người ta dùng phép biến đổi logarit sau:

$$v(m,n) = c \log_{10}(\delta + u(m,n)) \quad (3-11)$$

với  $c$  là hằng số tỉ lệ.  $\delta$  được coi là nhỏ so với  $u(m, n)$ . Thường  $\delta$  được chọn trong khoảng 3-10.

### 3.1.8 Mô hình hóa và biến đổi lược đồ xám

Về ý nghĩa của lược đồ xám và một số phép biến đổi lược đồ đã được trình bày trong tài liệu này. Xét đến một số biến đổi hay dùng sau:

$$f(u) = \sum_{x_1=0}^u P_u(x_i) \quad (3-12)$$

$$\text{với } P_u(x_i) = \frac{h(x_i)}{\sum_{i=0}^{L-1} h(x_i)} \quad (3-13)$$

trong đó,  $h(x_i)$  là lược đồ mức xám  $x_i$ .

Ngoài các biến đổi trên, người ta còn dùng một số biến đổi khác. Trong các biến đổi này, mức xám đầu vào  $u$ , trước tiên được biến đổi phi tuyến bởi một trong các hàm sau:

$$f(u) = \frac{\sum_{x_1=0}^u P_u^{1/n}(x_1)}{\sum_{x_1=0}^{L-1} P_u^{1/n}(x_1)} \quad \text{với } n = 2, 3, \dots \quad (3-14)$$

$$f(u) = \log(1+u); \quad u \geq 0 \quad (3-15)$$

$$f(u) = u^{1/n}; \quad u \geq 0, \quad n = 2, 3, \dots \quad (3-16)$$

Sau các biến đổi đó, đầu ra được lượng hoá đều. Ba phép biến đổi này được dùng trong lượng hóa ảnh. Nói chung, các biến đổi lược đồ nhằm biến đổi lược đồ từ một đường không thuận nhất sang đồng nhất để tiện cho phân tích ảnh.

## 3.2 CẢI THIỆN ẢNH DÙNG TOÁN TỬ KHÔNG GIAN

Cải thiện ảnh là làm cho ảnh có chất lượng tốt hơn theo ý đồ sử dụng. Thường là ảnh thu nhận có nhiều cần phải loại bỏ nhiễu hay ảnh không sắc nét bị mờ hoặc cần làm rõ các chi tiết như đường biên ảnh. Các toán tử không gian dùng trong kỹ thuật tăng cường ảnh được phân nhóm theo công dụng: làm trơn nhiễu, nổi biên. Để làm trơn nhiễu hay tách nhiễu, người ta sử dụng các

bộ lọc tuyến tính (lọc trung bình, thông thấp) hay lọc phi tuyến (trung vị, giả trung vị, lọc đồng hình). Từ bản chất của nhiễu (thường tương ứng với tần số cao) và từ cơ sở lý thuyết lọc là: bộ lọc chỉ cho tín hiệu có tần số nào đó thông qua do đó, để lọc nhiễu người ta thường dùng lọc thông thấp (theo quan điểm tần số không gian) hay lấy tổ hợp tuyến tính để san bằng (lọc trung bình). Để làm nổi cạnh (ứng với tần số cao), người ta dùng các bộ lọc thông cao, lọc Laplace.

Trước khi xem xét chi tiết các kỹ thuật áp dụng, cần phân biệt các loại nhiễu hay can thiệp trong quá trình xử lý ảnh. Trên thực tế tồn tại nhiều loại nhiễu; tuy nhiên người ta thường xem xét 3 loại nhiễu chính: nhiễu cộng, nhiễu nhân và nhiễu xung:

- **Nhiễu cộng**

Nhiễu cộng thường phân bố khắp ảnh. Nếu gọi ảnh quan sát (ảnh thu được) là  $X_{qs}$ , ảnh gốc là  $X_{gốc}$ , nhiễu là  $\eta$ , ảnh thu được có thể biểu diễn bởi:

$$X_{qs} = X_{gốc} + \eta \quad (3-17)$$

- **Nhiễu nhân**

Nhiễu nhân thường phân bố khắp ảnh và ảnh thu được sẽ biểu diễn với công thức:

$$X_{qs} = X_{gốc} * \eta \quad (3-17)$$

- **Nhiễu xung**

Nhiễu xung thường gây đột biến tại một số điểm ảnh.

### 3.2.1 Làm trơn nhiễu bằng lọc tuyến tính: lọc trung bình và lọc dải thông thấp

Do có nhiều loại nhiễu can thiệp vào quá trình xử lý ảnh nên cần có nhiều bộ lọc thích hợp. Với nhiễu cộng và nhiễu nhân ta dùng các bộ lọc thông thấp, trung bình và lọc đồng hình (Homomorphic); với nhiễu xung ta dùng lọc trung bị, giả trung vị, lọc ngoài (Outlier).

#### a. Lọc trung bình không gian

Với lọc trung bình, mỗi điểm ảnh được thay thế bằng trung bình trọng số của các điểm lân cận và được định nghĩa như sau:

$$v(m, n) = \sum_{(k,l) \in W} a(k, l) y(m - k, n - l) \quad (3-18)$$

Nếu trong kỹ thuật lọc trên, ta dùng các trọng số như nhau, phương trình trên sẽ trở thành:

$$v(m, n) = \frac{1}{N} \sum_{(k,l) \in W} y(m - k, n - l) \quad (3-19)$$

với :  $y(m, n)$ : ảnh đầu vào,

$v(m, n)$ : ảnh đầu ra,

$a(k, l)$  : là cửa sổ lọc.

với  $a_{k,l} = \frac{1}{N_w}$  và  $N_w$  là số điểm ảnh trong cửa sổ lọc  $W$ .

Lọc trung bình có trọng số chính là thực hiện chập ảnh đầu vào với nhân chập  $H$ . Nhân chập  $H$  trong trường hợp này có dạng:

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Trong lọc trung bình, thường người ta ưu tiên cho các hướng để bảo vệ biên của ảnh khỏi bị mờ khi làm trơn ảnh. Các kiểu mặt nạ được sử dụng tùy theo các trường hợp khác nhau. Các bộ lọc trên là bộ lọc tuyến tính theo nghĩa là điểm ảnh ở tâm của số sẽ được thay bởi tổ hợp các điểm lân cận chập với mặt nạ.

Giả sử đầu vào biểu diễn bởi ma trận  $I$ :

$$I = \begin{bmatrix} 4 & 7 & 3 & 7 & 1 \\ 5 & 7 & 1 & 7 & 1 \\ 6 & 6 & 1 & 8 & 3 \\ 5 & 7 & 5 & 7 & 1 \\ 5 & 7 & 6 & 1 & 2 \end{bmatrix}$$

Ảnh số thu được bởi lọc trung bình  $Y=H \otimes I$  có dạng:

$$Y = \frac{1}{9} \begin{bmatrix} 23 & 26 & 31 & 19 & 16 \\ 35 & 39 & 46 & 31 & 27 \\ 36 & 43 & 49 & 34 & 27 \\ 36 & 48 & 48 & 34 & 22 \\ 24 & 35 & 33 & 22 & 11 \end{bmatrix}$$

Một bộ lọc trung bình không gian khác cũng hay được sử dụng. Phương trình của bộ lọc đó có dạng:

$$Y[m, n] = \frac{1}{2} \left[ X[m, n] + \frac{1}{4} \{ X[m, n] + X[m-1, n] + X[m+1, n] + X[m, n-1] + X[m, n+1] \} \right] \quad (3-20)$$

Ở đây, nhân chập  $H$  có kích thước  $2 \times 2$  và mỗi điểm ảnh kết quả có giá trị bằng trung bình cộng của nó với trung bình cộng của 4 lân cận gần nhất.

Lọc trung bình trọng số là một trường hợp riêng của lọc thông thấp.

#### *b. Lọc thông thấp*

Lọc thông thấp thường được sử dụng để làm trơn nhiều. Về nguyên lý của bộ lọc thông thấp giống như đã trình bày trên. Trong kỹ thuật này người ta hay dùng một số nhân chập có dạng sau:

$$H_{11} = \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$H_b = \frac{1}{(b+2)^2} \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}$$

Ta dễ dàng nhận thấy khi  $b=1$ ,  $H_b$  chính là nhân chập  $H_{11}$  (lọc trung bình). Để hiểu rõ hơn bản chất khử nhiễu cộng của các bộ lọc này, ta viết lại phương trình thu nhận ảnh dưới dạng:

$$X_{qs}[m,n] = X_{góc}[m,n] + \eta[m,n] \quad (3-21)$$

Trong đó  $\eta[m, n]$  là nhiễu cộng có phương sai  $\sigma_n^2$ . Như vậy, theo cách tính của lọc trung bình ta có:

$$Y[m,n] = \frac{1}{N_w} \sum_{k,l \in W} X_{qs}(m-k, n-l) + \eta[m,n] \quad (3-22)$$

$$\text{hay: } Y[m,n] = \frac{1}{N_w} \sum_{k,l \in W} X_{qs}(m-k, n-l) + \frac{\sigma_n^2}{N_w} \quad (3-23)$$

Như vậy, nhiễu cộng trong ảnh đã giảm đi  $N_w$  lần.

### c. Lọc đồng hình hình (Homomorphie Filter)

Kỹ thuật lọc này hiệu quả với ảnh có nhiễu nhân. Thực tế, ảnh quan sát được gồm ảnh gốc nhân với một hệ số nhiễu. Gọi  $\bar{X}(m,n)$  là ảnh thu được,  $X(m, n)$  là ảnh gốc và  $\eta(m, n)$  là nhiễu, như vậy:

$$X(m, n) = \bar{X}(m, n) * \eta(m, n). \quad (3-24)$$

Lọc đồng hình thực hiện lấy logarit của ảnh quan sát. Do vậy ta có kết quả sau:

$$\text{Log}(X(m, n)) = \text{log}(\bar{X}(m, n)) + \text{log}(\eta(m, n)) \quad (3-25)$$

Rõ ràng, nhiễu nhân có trong ảnh sẽ bị giảm. Sau quá trình lọc tuyến tính, ta chuyển về ảnh cũ bằng phép biến đổi hàm e mũ.

## 3.2.2 Làm trơn nhiễu bằng lọc phi tuyến

Các bộ lọc phi tuyến cũng hay được dùng trong kỹ thuật tăng cường ảnh. Trong kỹ thuật này, người ta dùng bộ lọc trung vị, giả trung vị, lọc ngoài. Với lọc trung vị, điểm ảnh đầu vào sẽ được thay thế bởi trung vị các điểm ảnh còn lọc giả trung vị sẽ dùng trung bình cộng của 2 giá trị “trung vị” (trung bình cộng của max và min).

### a. Lọc trung vị

Trung vị được viết với công thức:

$$v(m, n) = \text{Trungvi}(y(m-k, n-l)) \quad \text{với } \{k, l\} \in W \quad (3-26)$$

Kỹ thuật này đòi hỏi giá trị các điểm ảnh trong cửa sổ phải xếp theo thứ tự tăng hay giảm dần so với giá trị trung vị. Kích thước cửa sổ thường được chọn sao cho số điểm ảnh trong cửa sổ là lẻ. Các cửa sổ hay dùng là cửa sổ có kích thước  $3 \times 3$ , hay  $5 \times 5$  hay  $7 \times 7$ . Thí dụ:

Nếu  $y(m) = \{2, 3, 8, 4, 2\}$  và cửa sổ  $W = (-1, 0, 1)$ , ảnh thu được sau lọc trung vị sẽ là:

$$v(m) = (2, 3, 4, 4, 2).$$

do đó:

$$\begin{aligned} v[0] &= 2 \text{ < giá trị biên >;} & v[1] &= \text{Trungvi}(2, 3, 8) = 3; & v[2] &= \text{Trungvi}(3, 4, 8) = 4; \\ v[3] &= \text{Trungvi}(8, 4, 2) = 4; & v[4] &= 2 \text{ < giá trị biên >.} \end{aligned}$$

### Tính chất của lọc trung vị:

- Lọc trung vị là loại lọc phi tuyến. Điều này dễ nhận thấy từ:



$$\text{Trungvi}(x(m)+y(m)) \neq \text{Trungvi}(x(m)) + \text{Trungvi}(y(m)).$$

- Có lợi cho việc loại bỏ các điểm ảnh hay các hàng mà vẫn bảo toàn độ phân giải.
- Hiệu quả giảm khi số điểm trong cửa sổ lớn hay bằng một nửa số điểm trong cửa sổ. Điều này dễ giải thích vì trung vị là  $(N_w+1)/2$  giá trị lớn nhất nếu  $N_w$  lẻ. Lọc trung vị cho trường hợp 2 chiều coi như lọc trung vị tách được theo từng chiều.

*b. Lọc ngoài (Outlier Filter)*

Giả thiết có ngưỡng nào đó cho các mức nhiễu (có thể dựa vào lược đồ xám). Tiến hành so sánh giá trị độ xám của một điểm ảnh với trung bình số học 8 lân cận của nó. Nếu sai lệch lớn hơn ngưỡng, điểm ảnh này được coi như nhiễu. Trong trường hợp đó, thay thế giá trị của điểm ảnh bằng giá trị trung bình 8 lân cận vừa tính được. Bộ lọc ngoài có thể diễn tả bằng công thức sau:

$$Y(m, n) = \begin{cases} \alpha(w) & \text{khi } |u(m, n) - \alpha(w)| > \delta \\ u(m, n) & \text{=} \end{cases} \quad (3-27)$$

với  $\alpha(w)$  là trung bình cộng các điểm trong lân cận  $w$ ;  $\delta$  là ngưỡng ngoài

Các cửa sổ tính toán thường chọn là 3x3. Tuy nhiên, cửa sổ có thể mở rộng đến 5x5 hay 7x7 để đảm bảo tính tương quan giữa các điểm ảnh. Vấn đề quan trọng là xác định ngưỡng để loại nhiễu mà vẫn không làm mất thông tin của ảnh.

**3.2.3 Mặt nạ gờ sai phân và làm nhẵn**

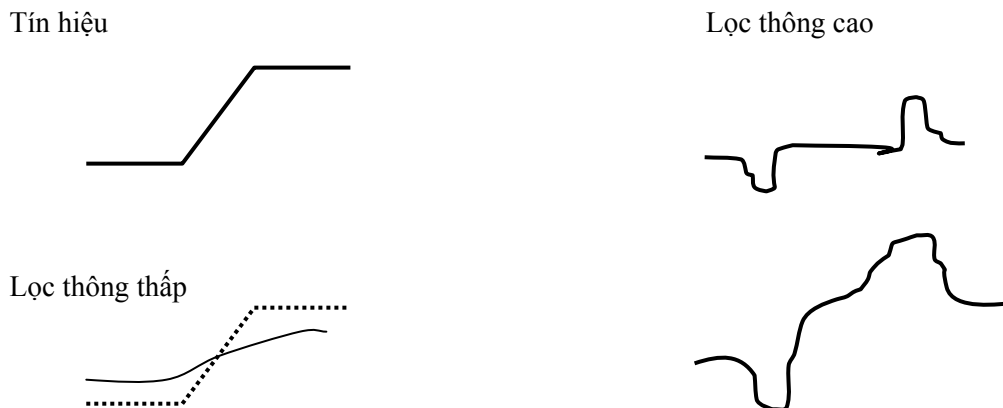
Mặt nạ gờ sai phân dùng khá phổ biến trong công nghệ in ảnh để làm đẹp ảnh. Với kỹ thuật này, tín hiệu đầu ra thu được bằng tín hiệu ra của bộ lọc gradient hay lọc dải cao bổ sung thêm đầu vào:

$$v(m, n) = u(m, n) + \lambda g(m, n) \quad (3-28)$$

với  $\lambda > 0$ ,  $g(m, n)$  là gradient tại điểm  $(m, n)$ . Hàm gradient dùng là hàm Laplace.

$$G(m, n) = u(m, n) - \{u(m-1, n) + u(m+1, n) + u(m, n+1)\} / 2 \quad (3-29)$$

Đây chính là mặt nạ chữ thập.



**Hình 3.5. Các toán tử gờ sai phân.**

### 3.2.4. Lọc thông thấp, thông cao và lọc dải thông

Toán tử trung bình không gian là lọc thông thấp. Nếu  $h_{LP}(m, n)$  biểu diễn bộ lọc thông thấp FIR (Finite Impulse Response) thì bộ lọc thông cao  $h_{HP}(m, n)$  có thể được định nghĩa:

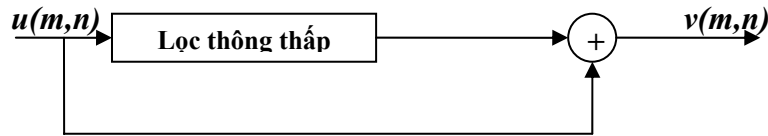
$$h_{HP}(m, n) = \delta(m, n) - h_{LP}(m, n) \quad (3-30)$$

Như vậy, bộ lọc thông cao có thể cài đặt một cách đơn giản như trên hình 3.6.

Bộ lọc dải thông có thể định nghĩa như sau:

$$H_{HP}(m, n) = h_{L1}(m, n) - h_{L2}(m, n) \quad (3-31)$$

với  $h_{L1}$  và  $h_{L2}$  là các bộ lọc thông thấp.



**Hình 3.6. Sơ đồ bộ lọc thông cao.**

Bộ lọc thông thấp thường dùng làm trơn nhiễu và nội suy ảnh. Bộ lọc thông cao dùng nhiều trong trích chọn biên và làm trơn ảnh, còn bộ lọc dải thông có hiệu quả làm nổi cạnh. Về biên sẽ được trình bày kỹ trong các phần sau. Tuy nhiên, dễ nhận thấy, biên là điểm có độ biến thiên nhanh về giá trị mức xám theo quan điểm về tần số tín hiệu. Như vậy, các điểm biên ứng với các thành phần tần số cao. Từ đó, có thể dùng bộ lọc thông cao để cải thiện nhiễu: nghĩa là có thể lọc các thành phần tần số thấp và giữ lại các thành phần tần số cao. Vì thế, lọc thông cao thường được dùng làm trơn biên trước khi tiến hành các thao tác với biên ảnh. Dưới đây là một số mặt nạ dùng trong lọc thông cao:

$$(1) \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (2) \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad (3) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

**Hình 3.7. Một số nhân chập trong lọc thông cao.**

Các nhân chập thông cao có đặc tính chung là tổng các hệ số của bộ lọc bằng 1. Nguyên nhân chính là ngăn cản sự tăng quá giới hạn của các giá trị mức xám (các giá trị điểm ảnh vẫn giữ được giá trị của nó một cách gần đúng không thay đổi quá nhiều với giá trị thực).

### 3.2.5. Khuếch đại và nội suy ảnh

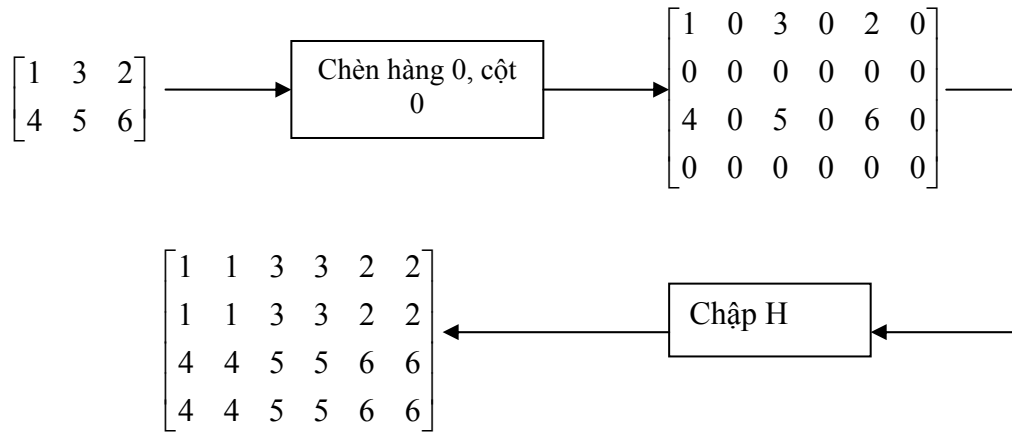
Có nhiều ứng dụng cần thiết phải phóng đại một vùng của ảnh. Có nghĩa là lấy một vùng của ảnh đã cho và cho hiện lên như một ảnh lớn. Có 2 phương pháp được dùng là lặp (Replication) và nội suy tuyến tính (Linear Interpolation).

#### Phương pháp lặp

Người ta lấy một vùng của ảnh kích thước  $M \times N$  và quét theo hàng. Mỗi điểm ảnh nằm trên đường quét sẽ được lặp lại 1 lần và hàng quét cũng được lặp lại 1 lần nữa. Như vậy, ta thu được ảnh với kích thước  $2N \times 2N$ . Điều này tương đương với việc chèn thêm một hàng 0 và 1 cột 0 rồi chập với mặt nạ  $H$ . Mặt nạ  $H$

$$H = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Kết quả thu được :  $v(m,n) = u(k,l)$  với  $k=\lfloor m/2 \rfloor$  và  $l=\lfloor n/2 \rfloor$



**Hình 3.8 Khuyếch đại bởi lặp 2x2.**

Hình 3.8 minh họa nội suy theo phương pháp lặp. Ở đây phép toán nêu trên là phép toán lấy phần nguyên của một số [1].

**Phương pháp nội suy tuyến tính**

Giả sử có một ma trận điểm ảnh. Theo phương pháp nội suy tuyến tính, trước tiên, hàng được đặt vào giữa các điểm ảnh theo hàng. Tiếp sau, mỗi điểm ảnh dọc theo cột được nội suy theo đường thẳng. Thí dụ, với mức độ khuyếch đại 2x2, nội suy tuyến tính theo hàng sẽ được tính theo công thức:

$$v_l(m,n) = u(m,n) \tag{3-32}$$

$$v_l(m,2n+1) = u(m,n) + u(m,n+1) \tag{3-33}$$

và nội suy tuyến tính của kết quả trên theo cột:

$$v_l(2m,n) = v_l(m,n) \tag{3-34}$$

$$v_l(2m+1,n) = v_l(m,n) + v_l(m+1,n) \tag{3-35}$$

với  $0 \leq m \leq M-1, 0 \leq n \leq N-1,$

Nếu dùng mặt nạ dạng:

$$H = \begin{pmatrix} 1/4 & 1/4 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}$$

ta cũng thu được kết quả trên.

Nội suy với bậc cao hơn cũng có thể áp dụng cách trên. Thí dụ, nội suy với bậc  $p$  ( $p$  nguyên), ta chèn  $p$  hàng với các số 0, rồi  $p$  cột với các số 0. Cuối cùng, tiến hành nhân chập  $p$  lần ảnh với mặt nạ  $H$  ở trên.

**3.2.6. Một số kỹ thuật cải thiện ảnh nhị phân**

Với ảnh nhị phân, mức xám chỉ có 2 giá trị là 0 hay 1, Do vậy, ta xét một phần tử ảnh như một phần tử logic và có thể áp dụng các toán tử hình học dựa trên khái niệm biến đổi hình học của một ảnh bởi một phần tử cấu trúc.

Phần tử cấu trúc là một mặt nạ dạng bất kỳ mà các phần tử của nó tạo nên một mô-típ. Người ta tiến hành rê mặt nạ đi khắp ảnh và tính giá trị điểm ảnh bởi các điểm lân cận với mô-típ của mặt nạ theo cách lấy hội hay lấy tuyến.

Dựa vào nguyên tắc trên, người ta sử dụng 2 kỹ thuật: dẫn ảnh (Dilatation) và co ảnh (Erosion).

#### a) Dẫn ảnh

Dẫn ảnh nhằm loại bỏ điểm đen bị vây bởi các điểm trắng. Trong kỹ thuật này, một cửa sổ  $(N+1) \times (N+1)$  được rê đi khắp ảnh và thực hiện đối sánh một pexel của ảnh với  $(N+1) \times 2 - 1$  điểm lân cận (không tính điểm ở tâm). Phép đối sánh ở đây thực hiện bởi phép tuyến logic. Thuật toán biến đổi được tóm tắt như sau:

```
For all pixel  $I(x,y)$  do
Begin
    tính  $F_{OR}(x, y)$       {tính OR logic}
    if  $F_{OR}(x, y)$       then  $ImaOut(x,y) < -1$ 
    Else  $ImaOut(x,y) < ImaIn(x,y)$ 
end
```

#### b) Co ảnh

Co ảnh là thao tác đối ngẫu của dẫn ảnh nhằm loại bỏ điểm trắng bị vây bởi các điểm đen. Trong kỹ thuật này, một cửa sổ  $(N+1) \times 2$  được rê đi khắp ảnh và thực hiện so sánh một pixel của ảnh với  $(N+1) \times 2 - 1$  điểm lân cận. Việc so sánh ở đây thực hiện bởi phép hội logic.

Áp dụng: người ta thường vận dụng kỹ thuật này cho các ảnh nhị phân như ảnh vân tay, chữ viết. Để không ảnh hưởng đến kích thước của đối tượng trong ảnh, người ta tiến hành  $n$  lần dẫn và  $n$  lần co.

### 3.3 KHÔI PHỤC ẢNH

**Khái niệm:** Khôi phục ảnh là phục hồi lại ảnh gốc so với ảnh ghi được đã bị biến dạng. Nói cách khác, khôi phục ảnh là các kỹ thuật cải thiện chất lượng những ảnh ghi đảm bảo gần được như ảnh thật khi ảnh bị méo.

Để khôi phục được ảnh có kết quả, điều cần thiết là phải biết được các nguyên nhân, các hàm (hay dạng) gây ra biến dạng ảnh. Các nguyên nhân biến dạng thường do:

- Do camera, đầu thu ảnh chất lượng kém.
- Do môi trường, ánh sáng, hiện trường (scene), khí quyển, nhiễu xung.
- Do chất lượng.

**Định nghĩa:** Kỹ thuật khôi phục ảnh có thể được xác định như việc ước lượng lại ảnh gốc hay ảnh lý tưởng từ ảnh quan sát được bằng cách đo ngược lại những hiện tượng gây biến dạng, qua đó ảnh được chụp. Như vậy, kỹ thuật khôi phục ảnh đòi hỏi kiến thức về các hiện tượng gây biến dạng ảnh.

**Mô hình chung:** Hầu hết các mô hình xác định ảnh gốc (mô hình tuyến tính, phi tuyến, khả biến, bất biến trong không gian) đều dựa trên hàm đáp ứng xung hai chiều  $h(m, n)$  (hay còn gọi là hàm trải điểm PSF (Point-Spread Function) như sau:

$$v(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} u(m, n)h(m - k, n - l) + \eta(m, n) \quad (3-36)$$

trong đó:

$u(m, n)$  là ảnh gốc;  $m \in [0, M - 1]; n \in [0, N - 1]$

$v(m, n)$  là ảnh ghi được;

$h(m - k, n - l)$  hàm đáp ứng xung hai chiều;  $k \in [0, N - 1], l \in [0, M - 1]$

Viết (3-36) ở dạng véctor ma trận ta có:

$$V = Hu + \eta \quad (3-37)$$

Ở đây  $u, v, \eta$  là các véctor  $M \times N$  chiều,  $H$  là ma trận khối vòng  $MN \times MN$  chiều; hàm  $h$  hoặc ma trận  $H$  mô tả quá trình biến dạng, nhưng trong quá trình tạo ( hay quá trình hình thành: formation) ảnh nên còn gọi là ma trận biến dạng trong quá trình khôi phục.

#### Các nguồn biến dạng:

Về mặt phương pháp, các nguồn tạo biến dạng có thể nhóm lại theo các xử lý như sau:

- Biến dạng điểm.
- Biến dạng không gian
- Biến dạng thường (Temporal)
- Biến dạng màu sắc (Chromatic)

Do quá trình tạo ảnh ghi được liên quan đến điểm ảnh xung quanh, chúng ta tập trung xét các biến dạng trong không gian. Một số ví dụ điển hình của biến dạng không gian được xem xét như sau.

- Nhiễu loạn của khí quyển (thiên văn) giữa các ống kính thu và đối tượng trong quá trình chụp ảnh. Do sai số hệ thống (hệ phi tuyến).
- Sai lệch hệ thống có thể biểu diễn bằng sai lệch hàm truyền (ví dụ: sự dịch pha hàm truyền cohenent trong quang học...)

#### Các kỹ thuật khôi phục ảnh:

- Mô hình khôi phục ảnh có: mô hình tạo ảnh, mô hình gây nhiễu, mô hình quan sát.
- Lọc tuyến tính có: lọc ngược, đáp ứng xung, lọc hữu hạn FIR.
- Các kỹ thuật khác: Entropy cực đại, mô hình Bayes, giải chập.

### 3.3.1. Các mô hình quan sát và tạo ảnh

#### Mô hình quan sát ảnh.

Đầu tiên, cần xem xét ảnh được hình thành như thế nào; sau đó biến đổi ngược (thực hiện lọc ngược) khử nhiễu để thu lại ảnh nguyên thủy.

Từ phương trình biến đổi tín hiệu ảnh có nhiễu, chúng ta có thể viết:

$$v(m, n) = g[w(m, n)] + \eta(m, n) \quad (3-38)$$

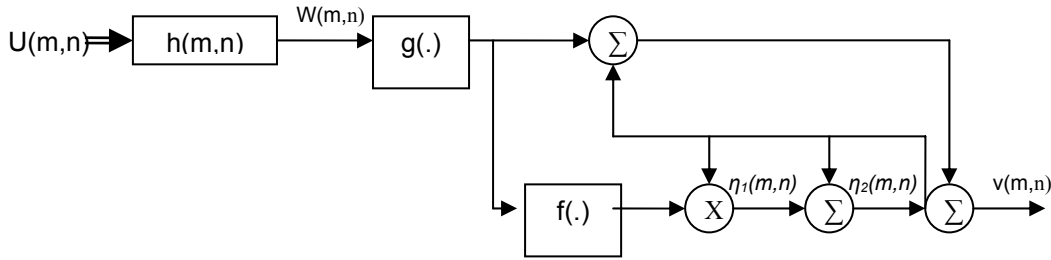
Với  $w(m, n)$  là đầu ra của hệ thống tuyến tính với đáp ứng hai chiều  $h(m, n)$  ta có:

$$w(m, n) = \sum \sum u(m, n)h(m - k, n - l) \quad (3-39)$$

Nhiều  $\eta(m, n)$  có thể gồm hai phần: nhiễu tích  $\eta_1(m, n)$ , nhiễu cộng  $\eta_2(m, n)$  đều được giả thiết là nhiễu trắng Gauss độc lập tương hỗ với trung bình 0:

$$\eta(m, n) = f[g(w(m, n))] \eta_1(m, n) + \eta_2(m, n) \quad (3-40)$$

trong đó: các hàm  $g(\cdot), f(\cdot)$  là các biến đổi (nói chung là phi tuyến) đặc trưng cho quá trình phát hiện và lưu trữ ảnh. Từ (3-40) mô hình đó được biểu diễn theo hình khối như sau:



**Hình 3.9. Quá trình phát hiện và lưu trữ ảnh**

**a) Mô hình nhiễu.**

Mô hình nhiễu là mô hình tổng quát. Trong hệ thống cụ thể như quang điện, mô hình nhiễu gây biến dạng được biểu diễn cụ thể như sau:

$$\eta(m, n) = \sqrt{g(m, n)} \eta_1(m, n) + \eta_2(m, n) \quad (3-41)$$

Trong đó  $\eta_1(m, n)$  là nhiễu phụ thuộc thiết bị, ở đó xảy ra việc truyền điện tử ngẫu nhiên. Việc truyền điện tử ngẫu nhiên thường được biểu diễn bằng phân bố Poisson (một số trường hợp phân bố nhiễu này tiệm cận đến phân bố Gauss). Do phân bố Poisson có trung bình và sai lệch bằng nhau nên nếu  $\eta_1$  có sai lệch là đơn vị thì thành phần phụ thuộc gắn với nhiễu  $\eta_1$  là  $\sqrt{\sigma}$ . Thành phần  $\eta_2(m, n)$  biểu diễn nhiễu gây ra do nhiệt và thường được mô hình hóa theo nhiễu trắng.

- Một số mô hình không gian ảnh hưởng nhiệt một cách đáng kể do đó như hệ thống phim.

$$\eta(m, n) = \sqrt{g(m, n)} \eta_1(m, n) \quad (3-42)$$

- Mô hình phim khác:

$$\eta(m, n) = e[g(m, n)]^2 \eta_1(m, n) \quad (3-43)$$

Trong đó  $e \in [1/3; 1/2]$ : hệ số chuẩn hóa.

*Tính toán gần đúng:* các thành phần bị nhiễu  $\eta_1(m, n)$  tác động gây khó khăn cho việc khôi phục ảnh. Để giải quyết theo phương pháp tương đương (hay gần đúng) người ta dùng giá trị trung bình không gian  $\mu_w$  thay cho  $w$  tức là:

$$\mu_w = W(m, n) \quad (3-44)$$

khi đó:

$$\eta(m, n) = f[g(\mu_w)]\eta_1(m, n) + \eta_2(m, n) \quad (3-45)$$

và  $\eta(m, n)$  trở thành mô hình nhiễu trắng Gauss.

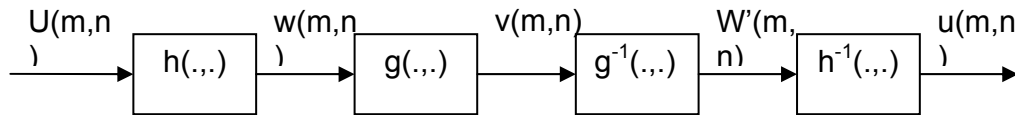
**Nhiều đốm:** ngoài 1 số mô hình trên, ảnh còn có thể bị biến dạng bởi nhiễu đốm (Speckle Noise). Nhiễu đốm xảy ra nếu bề mặt đối tượng có độ lồi lõm dạng bước sóng và tăng gấp bội nếu đối tượng đó có độ phân giải thấp. Mô hình nhiễu đốm có thể được mô tả như sau:

$$v(m, n) = u(m, n)s(m, n) + \eta(m, n) \quad (3-46)$$

Trong phạm vi tài liệu này không mô tả chi tiết về dạng nhiễu này. Đề nghị tham khảo các tài liệu [1, 4, 5].

### 3.3.2 Các bộ lọc

#### a) Kỹ thuật lọc ngược (Inverse Filter)



**Hình 3.10. Kỹ thuật lọc ngược**

$$g^T(x) = g^{-1}[g(x)] \quad \text{với } g^{-1}(x) = x \quad (3-47)$$

$$h^T(x, y, k, l) = h^{-1}(x, y, k, l) \quad (3-48)$$

$$FT \left[ \sum_{k, l=-\infty}^{+\infty} h^T(x, y, l)h(k', l'; k, l) \right] = \delta(x - k, y - l) \quad (3-49)$$

Việc thiết kế bộ lọc ngược khá khó khăn, do vậy chuyển sang biến đổi Fourier 2 vế (3-49) Chú ý rằng :

$$H(w_1, w_2)H^T(w_1, w_2) = 1$$

do đó:

$$H^T(w_1, w_2) = 1/H(w_1, w_2) \quad (3-50)$$

trong đó : biến đổi ngược Fourier của  $H(w_1, w_2)$  là  $h(x, y)$

Như vậy đã tìm được  $H^T(w_1, w_2)$ . Tương tự cũng xác định được  $G^T(w_1, w_2)$ .

#### b) Khôi phục ảnh bằng phương pháp bình phương tối thiểu.

Mô hình nhiễu có thể viết lại theo dạng véc tơ và ma trận như sau:

$$v = Hu + \eta \quad \text{hay } \eta = v - Hu \quad (3-51)$$

trong đó:  $u, v, \eta$  là các véc tơ ảnh thật, véc tơ ảnh ghi được, véc tơ nhiễu tương ứng;  $H$  là ma trận hàm truyền của hệ thống.

Nếu không nhiễu ( $\eta = 0$ ), cần chọn  $u$  sao cho  $Hu$  xấp xỉ  $v$  theo nghĩa bình phương tối thiểu. Với cách lập luận như vậy chúng ta có thể xây dựng hàm mục tiêu:

$$y(\hat{u}) = \|\eta\|^2 = \|v - H\hat{u}\|^2 \rightarrow \min \quad (3-52)$$

trong đó, các giá trị trên được định nghĩa như sau:

$$\|\eta\|^2 = \eta^T \eta; \|v - h\hat{u}\|^2 = (v' - H\hat{u})^T (v - H\hat{u}) \quad (3-53)$$

$$\frac{\partial y(\hat{u})}{\partial (\hat{u})} = -2HT(v - H\hat{u}) = 0 \quad (3-54)$$

Giả sử ta có:  $\hat{u} = (H^T H)^{-1} H^T v \quad (3-55)$

Chọn  $M=N$ , khi đó  $H$  là ma trận vuông. Giả thiết  $H^{-1}$  tồn tại, khi đó ta có

$$\hat{u} = H^{-1} (H^T)^{-1} H^T v \quad (3-56)$$

Dễ thấy,  $(H^T)^{-1} H^T$  là ma trận đơn vị  $I$ ; do đó  $\hat{u}$  trong công thức (3-56) có thể tính một cách đơn giản. Cần chú ý rằng biểu thức này xác định được với một số giả thiết đã nêu. Trong thực tế, nhiều điều kiện ràng buộc không chắc chắn đảm bảo do đó độ chính xác của giá trị ảnh ước lượng không hoàn toàn giống ảnh mong muốn nhưng sẽ cho kết quả tốt hơn, giảm độ méo của ảnh.

### CÂU HỎI ÔN TẬP CHƯƠNG 3

Cho ảnh số và các nhân chập sau:

$$I = \begin{pmatrix} 4 & 7 & 2 & 7 & 1 \\ 5 & 7 & 1 & 7 & 13 \\ 6 & 6 & 30 & 8 & 3 \\ 5 & 7 & 6 & 1 & 2 \\ 5 & 7 & 6 & 1 & 2 \end{pmatrix}; \quad H_{tt} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}; \quad H_{tc} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

1. Ảnh trên có nhiễu không? Đó là loại nhiễu gì?
2. Minh họa khử nhiễu trên bằng bộ lọc thông thấp  $H_{tt}$ .
3. Hãy tính kết quả của nhân chập ảnh với nhân chập  $H_{tc}$ .
4. Hãy biến đổi ảnh sau khi khử nhiễu về ảnh nhị phân (dùng kỹ thuật phân ngưỡng hay dựa vào lược đồ xám).
5. Viết thủ tục dùng kỹ thuật lọc trung vị sử dụng bộ lọc chữ thập kích thước 3x3 và 5x5. Việc sắp xếp các điểm theo thuật toán tùy chọn (chọn đơn giản, chèn tuyến tính hay đổi chỗ).
6. Viết thủ tục cải thiện ảnh dùng kỹ thuật lọc theo mô hình Gauss.



## CHƯƠNG 4: CÁC PHƯƠNG PHÁP PHÁT HIỆN BIÊN

Học xong phần này sinh viên có thể nắm được:

1. Khái quát về biên. Các kỹ thuật dò biên
2. Phương pháp phát hiện biên cục bộ
3. Dò biên theo quy hoạch động
4. Dò biên theo các phương pháp khác

### 4.1 GIỚI THIỆU BIÊN VÀ KỸ THUẬT PHÁT HIỆN BIÊN

#### 4.1.1 Một số khái niệm

Định nghĩa và khái niệm

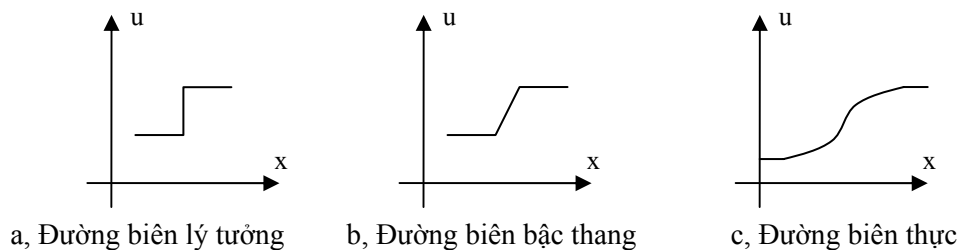
*Điểm Biên*: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng.

*Đường biên (đường bao: boundary)*: tập hợp các điểm biên liên tiếp tạo thành một đường biên hay đường bao.

*Ý nghĩa của đường biên trong xử lý*: ý nghĩa đầu tiên: đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám (màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách.

*Tầm quan trọng của biên*: để thấy rõ tầm quan trọng của biên, xét ví dụ sau: khi người họa sỹ muốn vẽ một danh nhân, họa sỹ chỉ cần vẽ vài đường nét tốc họa mà không cần vẽ một cách đầy đủ.

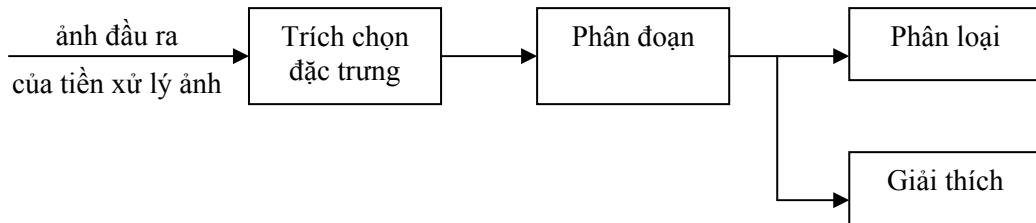
*Mô hình biểu diễn đường biên*, theo toán học: điểm ảnh có sự biến đổi mức xám  $u(x)$  một cách đột ngột theo hình dưới.



**Hình 4.1 Đường bao của ảnh**

Các khái niệm và định nghĩa tóm tắt trên là cơ sở giúp ta hiểu và dùng để hiểu cách xây dựng, thiết kế các kỹ thuật phát hiện biên ảnh.

*Chú ý:* Phát hiện biên là một phần trong phân tích ảnh, sau khi đã lọc ảnh (hay tiền xử lý ảnh). Các bước của phân tích ảnh có thể mô tả theo sơ đồ dưới đây. Việc dò và tìm biên ảnh là một trong các đặc trưng thuộc khối trích chọn đặc trưng.



**Hình 4.2** Các bước xử lý và phân tích ảnh

#### 4.1.2 Phân loại các kỹ thuật phát hiện biên

Từ định nghĩa toán học của biên người ta sử dụng hai phương pháp phát hiện biên như sau (phương pháp chính)

*a, Phương pháp phát hiện biên trực tiếp:* phương pháp này chủ yếu dựa vào sự biến thiên độ sáng của điểm ảnh để làm nổi biên bằng kỹ thuật đạo hàm.

- Nếu lấy đạo hàm bậc nhất của ảnh: ta có phương pháp Gradient
- Nếu lấy đạo hàm bậc hai của ảnh: ta có phương pháp Laplace.

Hai phương pháp này được gọi chung là phương pháp dò biên cục bộ.

Ngoài ra, người ta còn sử dụng phương pháp “đi theo đường bao” dựa vào công cụ toán học là nguyên lý quy hoạch động và được gọi là phương pháp dò biên tổng thể. Phương pháp dò biên trực tiếp có hiệu quả và ít bị tác động của nhiễu.

*b, Phương pháp phát hiện biên gián tiếp:* Nếu bằng cách nào đấy, chúng ta thu được các vùng ảnh khác nhau thì đường phân cách giữa các vùng đó chính là biên. Nói cách khác, việc xác định đường bao của ảnh được thực hiện từ ảnh đã được phân vùng. Phương pháp dò biên gián tiếp khó cài đặt nhưng áp dụng tốt khi sự biến thiên độ sáng nhỏ.

*Chú ý:* Kỹ thuật dò biên và phân vùng ảnh là hai bài toán đối ngẫu của nhau.

#### 4.1.3 Quy trình phát hiện biên.

*B1: Do ảnh ghi được thường có nhiễu, bước một là phải lọc nhiễu theo các phương pháp đã tìm hiểu ở các phần trước.*

*B2: Làm nổi biên sử dụng các toán tử phát hiện biên.*

*B3: Định vị biên. Chú ý rằng kỹ thuật nổi biên gây tác dụng phụ là gây nhiễu làm một số biên giả xuất hiện do vậy cần loại bỏ biên giả.*

*B4: Liên kết và trích chọn biên.*

## 4.2 PHƯƠNG PHÁP PHÁT HIỆN BIÊN CỤC BỘ

### 4.2.1 Phương pháp Gradient

*Định nghĩa:* Gradient là một vec tơ  $f(x, y)$  có các thành phần biểu thị tốc độ thay đổi mức xám của điểm ảnh (theo hai hướng  $x, y$  trong bối cảnh xử lý ảnh hai chiều) tức:

$$\frac{\partial f(x, y)}{\partial x} = f'_x \approx \frac{f(x + dx, y) - f(x, y)}{dx} \quad (4-1)$$

$$\frac{\partial f(x, y)}{\partial y} = f'_y \approx \frac{f(x, y + dy) - f(x, y)}{dy} \quad (4-2)$$

Trong đó  $dx, dy$  là khoảng cách giữa 2 điểm kế cận theo hướng  $x, y$  tương ứng (thực tế chọn  $dx = dy = 1$ ). Đây là phương pháp dựa theo đạo hàm riêng bậc nhất theo hướng  $x, y$ .

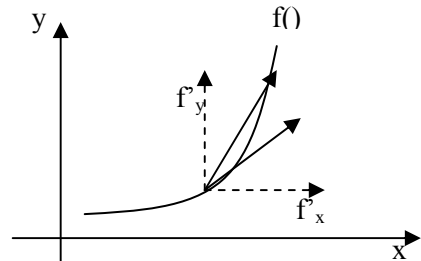
Gradient trong tọa độ góc  $(r, \theta)$ , với  $r$  là vec tơ,  $\theta$ : góc

$$\frac{df(r, \theta)}{dr} = \frac{\partial f}{\partial x} \frac{dx}{dr} + \frac{\partial f}{\partial y} \frac{dy}{dr} = f'_x \cos \theta + f'_y \sin \theta \quad (4-3)$$

$f(.)$  đạt cực đại khi  $\left(\frac{df(.)}{dr}\right) = 0$

tức:  $f'_x \cos \theta + f'_y \sin \theta = 0$  hay:

$$\begin{cases} \operatorname{tg} \theta = \frac{\sin \theta}{\cos \theta} = \frac{f'_y}{f'_x} \\ \theta = \operatorname{arctg} \left( \frac{f'_y}{f'_x} \right) \text{ và } f_{\max} = \sqrt{f'^2_x + f'^2_y} \end{cases} \quad (4-4)$$



a. *Kỹ thuật Gradient.* Theo định nghĩa về Gradient, nếu áp dụng nó vào xử lý ảnh, việc tính toán sẽ rất phức tạp. Để đơn giản mà không mất tính chất của phương pháp Gradient, người ta sử dụng kỹ thuật Gradient dùng cặp mặt nạ  $H_1, H_2$  trực giao (theo 2 hướng vuông góc). Nếu định nghĩa  $g_1, g_2$  là Gradient theo hai hướng  $x, y$  tương ứng thì biên độ  $g(m, n)$  tại điểm  $(m, n)$  được tính:

$$g(m, n) = \sqrt{g_1^2(m, n) + g_2^2(m, n)} = A_0 \quad (4-5)$$

$$\theta_r(m, n) = \operatorname{artg}(g_2(m, n)) \quad (4-6)$$

Để giảm độ phức tạp tính toán,  $A_0$  được tính gần đúng như sau:

$$A_0 = |g_1(m, n)| + |g_2(m, n)| \quad (4-7)$$

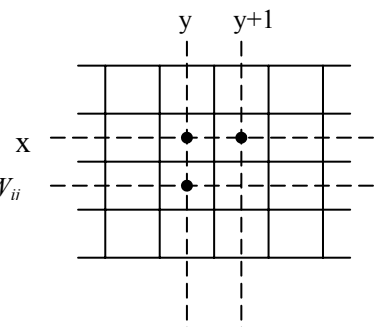
Xét một số toán tử Gradient tiêu biểu như toán tử Robert, Sobel, Prewitt, đẳng hướng (Isometric), 4-lân cận như dưới đây.

• **Toán tử Robert (1965).**

Robert áp dụng công thức tính Gradient tại điểm  $(x, y)$  như hình bên  $W_{ii}$

Với mỗi điểm ảnh  $I(x, y)$  đạo hàm theo  $x, y$  được ký hiệu tương ứng:

$g_x, g_y$ :



$$\begin{cases} g_x = I(x+1, y) - I(x, y) \\ g_y = I(x, y+1) - I(x, y) \end{cases} \quad (4-8)$$

Các công thức kể trên được cụ thể hóa bằng các mặt nạ theo chiều x và y tương ứng như sau:

$$H_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \downarrow$$

Hướng ngang (x)                      Hướng dọc (y)

**Hình 4.5. Mặt nạ Robert.**

$$A_l = |g_1(m, n) + g_2(m, n)| \quad (4-9)$$

$$A_e = \text{Max}[|g_1(m, n)|, |g_2(m, n)|] \quad (4-10)$$

• **Toán tử (mặt nạ) Sobel.**

Toán tử Sobel được Duda và Hart [5] đặt ra năm 1973 với các mặt nạ tương tự như của Robert nhưng cấu hình khác như sau:

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Hướng ngang (x)                      Hướng dọc (y)

• **Mặt nạ Prewitt**

Toán tử được Prewitt đưa ra vào năm 1970 có dạng:

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Hướng ngang (x)                      Hướng dọc (y)

• **Mặt nạ đẳng hướng:**

Một mặt nạ khác cũng được nêu như dưới đây gọi là mặt nạ đẳng hướng (Isometric).

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \downarrow$$

Hướng ngang (x)                      Hướng dọc (y)

• **Toán tử 4-lân cận (4-Neighbour Operator).**

Toán tử 4-lân cận được Chaudhuri và Chandor (1984) nêu ra, trong đó mặt nạ có kích thước 3x3 được thay cho mặt nạ 2x2 của toán tử Robert. Các mặt nạ này được cho:

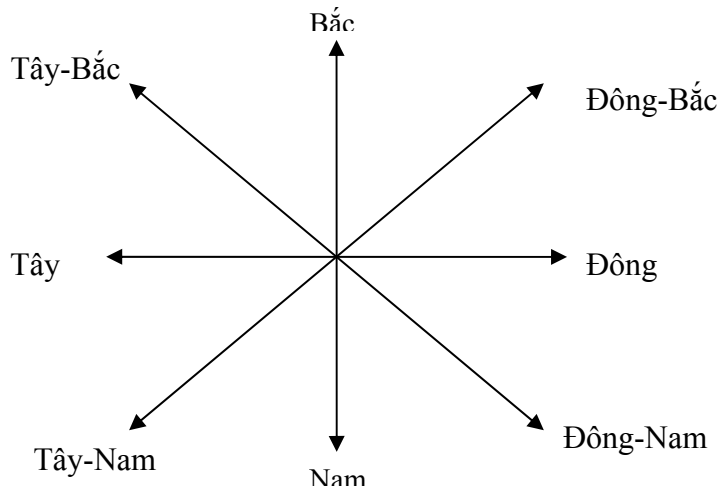
0	0	0
-1	0	1
0	0	0

Theo hướng x

0	-1	0
0	0	1
0	0	0

Theo hướng y

**Hình 4.6: Mặt nạ 4-lân cận.**



**Hình 4.8 Mặt nạ 8 hướng theo Kirsh.**

**Một vài nhận xét:**

1. Toán tử Prewitt có thể tách sườn đứng tốt hơn toán tử Sobel, trong khi đó toán tử Sobel tách các sườn trên các điểm ở đường chéo tốt hơn. Mặt khác, các toán tử Robert và toán tử 4-lân cận có nhược điểm là nhảy với nhiễu. Các toán tử Gradient và Sobel giảm nhiễu do tác dụng của lọc trung bình các điểm lân cận. Như vậy, để đạt được kết quả mong muốn các toán tử Gradient thường được dùng trước để làm sạch nhiễu.

2. Các mặt nạ của các toán tử trên có kích thước 2x2 hoặc 3x3 chiều. Các mặt nạ có số chiều lớn hơn cũng được sử dụng. Ví dụ trong kỹ thuật phát hiện biên người ta dùng mặt nạ 5x5 cho toán tử Sobel:

$$H_x = \begin{bmatrix} 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \\ 2 & 1 & 0 & 1 & -2 \end{bmatrix} \quad H_y = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

**Hình 4.7 Toán tử Sobel 5x5.**

3. Các toán tử kể trên đều sử dụng các mặt nạ theo hai chiều (x,y) tức là bốn hướng (-x, y; -y, y). Với mục đích cho kết quả tinh và chính xác hơn (khi mà tốc độ và bộ nhớ máy tính tốt).

b. Toán tử la bàn:

Kirsh đã đề xuất các mặt nạ theo 8 hướng như 8 hướng của la bàn (Compass). Hình 4.8 là mô hình 8 hướng và được đặt tên theo hướng địa lý và theo chiều kim đồng hồ: Đông, Đông-Nam, Nam, Nam-Tây, Tây, Tây-Nam, Tây-Bắc, Bắc, Đông-Bắc; mỗi hướng lệch nhau 45°.

• Toán tử la bàn Kirsh:

Có nhiều toán tử la bàn khác nhau. Trong phạm vi tài liệu này, ta xem xét toán tử la bàn Kirsh đặc trưng bởi tám mặt nạ với kích thước 3x3 như sau:

$$\begin{aligned}
 H_{\text{Bãã}} &= \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}; H_{\text{Đông-Bãã}} = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}; H_{\text{Đông}} = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}; \\
 H_{\text{Đông-Nam}} &= \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}; H_{\text{Nam}} = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}; H_{\text{Tây-Nam}} = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}; \\
 H_{\text{Tây}} &= \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}; H_{\text{Tây-Bãã}} = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}
 \end{aligned}$$

Ký hiệu  $A_i; i= 1, 2, \dots, 8$  là Gradient theo 8 hướng như 8 mặt nạ kể trên, khi đó biên độ Gradient tại điểm ảnh (x,y) được tính theo

$$A(x, y) = \text{Max} (|g_i(x,y)|) \quad i=1, 2, \dots, 8. \quad (4-11)$$

• Toán tử la bàn khác:

Ngoài toán tử la bàn Kirsh, một số toán tử la bàn khác sử dụng bộ mặt nạ 8 hướng khác như:

$$\begin{aligned}
 H_{\text{Bắc}} &= \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; H_{\text{Đông-Bắc}} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}; H_{\text{Đông}} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}; \\
 H_{\text{Đông-Nam}} &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}; H_{\text{Nam}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}; H_{\text{Tây-Nam}} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}; \\
 H_{\text{Tây}} &= \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}; H_{\text{Tây-Bãã}} = \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}
 \end{aligned}$$

hoặc:

$$\begin{aligned}
 H_{B\ddot{a}\ddot{a}} &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}; \quad H_{\text{Đông-B}\ddot{a}\ddot{a}} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}; \quad H_{\text{Đông}} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \\
 H_{\text{Đông-Nam}} &= \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}; \quad H_{\text{Nam}} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad H_{\text{Tây-Nam}} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}; \\
 H_{\text{Tây}} &= \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}; \quad H_{\text{Tây-B}\ddot{a}\ddot{a}} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}.
 \end{aligned}$$

Trường hợp tổng quát, người ta có thể mở rộng các mặt nạ với  $n$  hướng cách đều tương ứng với các mặt  $W_i; i=1, 2, \dots, n$ . Khi đó, biên độ tại hướng thứ  $i$  với mặt nạ  $W_i$  được xác định:

$$A(x, y) = \text{Max}(|W_i^T I(x, y)|) \quad \text{với } i=1, 2, \dots, n. \quad (4-12)$$

**Tóm lại:** Các kỹ thuật sử dụng phương pháp Gradient khá tốt khi độ sáng có tốc độ thay đổi nhanh, khá đơn giản trên cơ sở các mặt nạ theo các hướng. Nhược điểm của các kỹ thuật Gradient là nhạy cảm với nhiễu và tạo các biên kép làm chất lượng biên thu được không cao.

### c. Kỹ thuật Laplace

Để khắc phục hạn chế và nhược điểm của phương pháp Gradient, trong đó sử dụng đạo hàm riêng bậc nhất người ta nghĩ đến việc sử dụng đạo hàm riêng bậc hai hay toán tử Laplace. Phương pháp dò biên theo toán tử Laplace hiệu quả hơn phương pháp toán tử Gradient trong trường hợp mức xám biến đổi chậm, miền chuyển đổi mức xám có độ trải rộng.

Toán tử Laplace được định nghĩa như sau:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4-13)$$

Toán tử Laplace dùng một số kiểu mặt nạ khác nhau nhằm tính gần đúng đạo hàm riêng bậc hai. Các dạng mặt nạ theo toán tử Laplace bậc 3x3 có thể:

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}; \quad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}; \quad H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -1 \\ 1 & -2 & 1 \end{bmatrix}$$

**Ghi chú:** Mặt nạ  $H_1$  còn cải biên bằng việc lấy giá trị ở tâm bằng 8 thay vì giá trị 4. Để thấy rõ việc xấp xỉ đạo hàm riêng bậc hai trong không gian hai chiều với mặt nạ  $H_1$  làm ví dụ, ta có thể tính gần đúng như sau:

$$\frac{\partial^2 f}{\partial x^2} = 2f(x, y) - f(x-1, y) - f(x+1, y)$$

$$\frac{\partial^2 f}{\partial y^2} = 2f(x, y) - f(x, y-1) - f(x, y+1)$$

do đó:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 4f(x, y) - f(x-1, y) - f(x, y-1) - f(x+1, y) - f(x, y+1) \quad (4-14)$$

(Học viên có thể kiểm tra giá trị và dấu của công thức trên với các giá trị trong mặt nạ  $H_I$ ).

**Tóm lại:** Kỹ thuật theo toán tử Laplace tạo đường biên mảnh (có độ rộng 1 pixel). Nhược điểm của kỹ thuật này rất nhạy với nhiễu, do vậy đường biên thu được thường kém ổn định.

*d. Tách sườn ảnh theo phương pháp Canny.*

Bộ tác sườn ảnh theo Canny (1986) dựa trên cặp đạo hàm riêng bậc nhất với việc làm sạch nhiễu. Mục này được đề riêng vì đây là phương pháp tách đường biên khá phổ biến được dùng theo toán tử đạo hàm. Như đã nói, phương pháp đạo hàm chịu ảnh hưởng lớn của nhiễu. Phương pháp đạt hiệu quả cao khi xấp xỉ đạo hàm bậc nhất của Gauss.

$$\nabla f = \nabla(G \otimes I) = f_x + f_y \quad (4-15)$$

với  $f_x, f_y$  là đạo hàm riêng theo  $x$  và  $y$  của  $f$ .

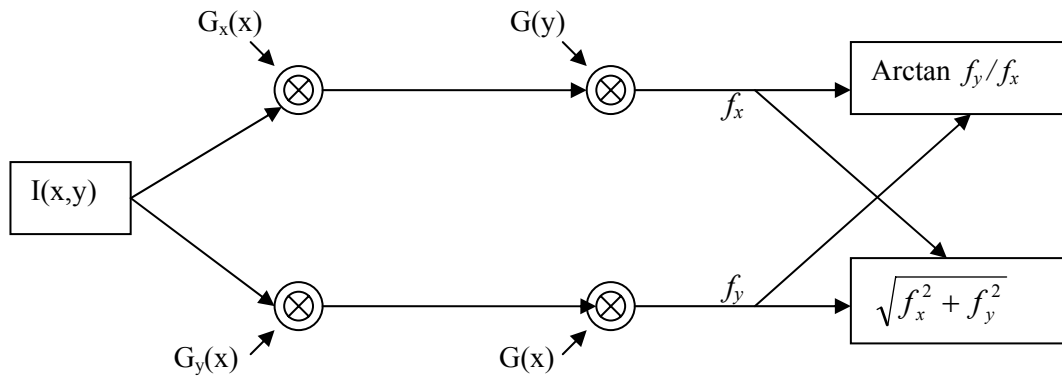
do vậy:

$$\nabla f = \nabla(G \otimes I)_x + \nabla(G \otimes I)_y = (G_x \otimes I) + (G_y \otimes I) \quad (4-16)$$

Lấy đạo hàm riêng theo  $x$  và  $y$  của  $G$  ta được:

$$G_x(x, y) = \sigma \frac{-x}{2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4-17)$$

$$G_y(x, y) = \sigma \frac{-y}{2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4-18)$$



**Hình 4.9. Mô hình tính của phương pháp Canny.**

Do bộ lọc Gauss là tách được, ta có thể thực hiện riêng biệt các tích chập theo  $x$  và  $y$ :

$$G_x(x, y) = G_x(x) \otimes G(y) \quad \text{và} \quad G_y(x, y) = G_y(y) \otimes G(x) \quad (4-19)$$

Từ đó ta có:

$$f_x(x, y) = G_x(x) \otimes G(y) \otimes I \quad \text{và} \quad f_y(x, y) = G_y(y) \otimes G(x) \otimes I \quad (4-20)$$

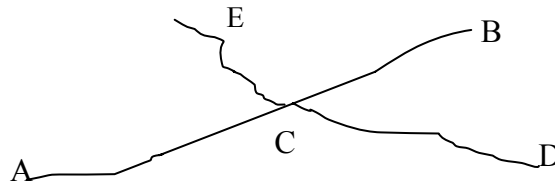
Với biên độ và hướng tính theo công thức trên, thuật toán được minh họa trên Hình 4.9.

#### 4.2.2. Dò biên theo quy hoạch động



Như trên đã nói, dò biên theo phương pháp Gradient là xác định cực trị cục bộ của Gradient theo các hướng; còn phương pháp Laplace dựa vào cắt điểm không của đạo hàm bậc hai. Phương pháp dò biên theo quy hoạch động là phương pháp tìm cực trị tổng thể theo nhiều bước. Nó dựa vào nguyên lý tối ưu của Bellman. Nguyên lý này phát biểu như sau: “Con đường tối ưu giữa 2 điểm cho trước cũng là tối ưu giữa 2 điểm bất kỳ nằm trên đường tối ưu đó”.

Thí dụ, nếu  $C$  là một điểm trên con đường tối ưu giữa  $A$  và  $B$  thì đoạn  $CB$  cũng là con đường tối ưu từ  $C$  đến  $B$  không kể đến ta đến  $C$  bằng cách nào (Hình 4.10).



**Hình 4.10. Minh họa nguyên lý Bellman.**

Trong kỹ thuật này, giả sử bản đồ biên đã được xác định và được biểu diễn dưới dạng đồ thị liên thông  $N$  chặng. Giả sử hàm đánh giá được tính theo công thức:

$$S(x_1, \dots, x_N, N) = \sum_{k=1}^N |g(x_k)| - \alpha \sum_{k=1}^N |\theta(x_k) - \theta(x_{k-1})| - \beta \sum_{k=2}^N d(x_N, x_{N-1}) \quad (4-21)$$

với:

- $x_k, k=1, \dots, N$ : biểu diễn các đỉnh của đồ thị trong chặng thứ  $k$ ;
- $d(x, y)$ : khoảng cách giữa 2 đỉnh  $x$  và  $y$  tính theo các định nghĩa tương ứng về khoảng cách;
- $|g(x_k)|$  và  $\theta(x_k)$  là Gradient biên độ và Gradient hướng ở đỉnh  $x_k$ .
- $\alpha$  và  $\beta$  là các hằng số không âm.

Đường bao tối ưu sẽ nhận được bằng cách nối các đỉnh  $\overline{x_k}, k=1, \dots, N$  nào đó sao cho  $S(x_1, \dots, x_N, N)$  đạt cực đại.

Định nghĩa hàm  $\phi$  như sau:

$$\phi(x_N, N) = \text{Max}_{x_1, \dots, x_{N-1}} \{S(x_1, \dots, x_N, N)\} \quad (4-22)$$

Viết lại công thức (4-21) một cách đệ quy ta có:

$$S(x_1, \dots, x_N, N) = S(x_1, \dots, x_{N-1}, N-1) + |g(x_N)| - \alpha |\theta(x_N) - \theta(x_{N-1})| - \beta d(x_N, x_{N-1}) \quad (4-23)$$

Đặt  $f(x_{N-1}, x_N) = |g(x_N)| - \alpha |\theta(x_N) - \theta(x_{N-1})| - \beta d(x_N, x_{N-1})$  và thay vào (4-23) ta có:

$$S(x_1, \dots, x_N, N) = S(x_1, \dots, x_{N-1}, N-1) + f(x_{N-1}, x_N) \quad (4-24)$$

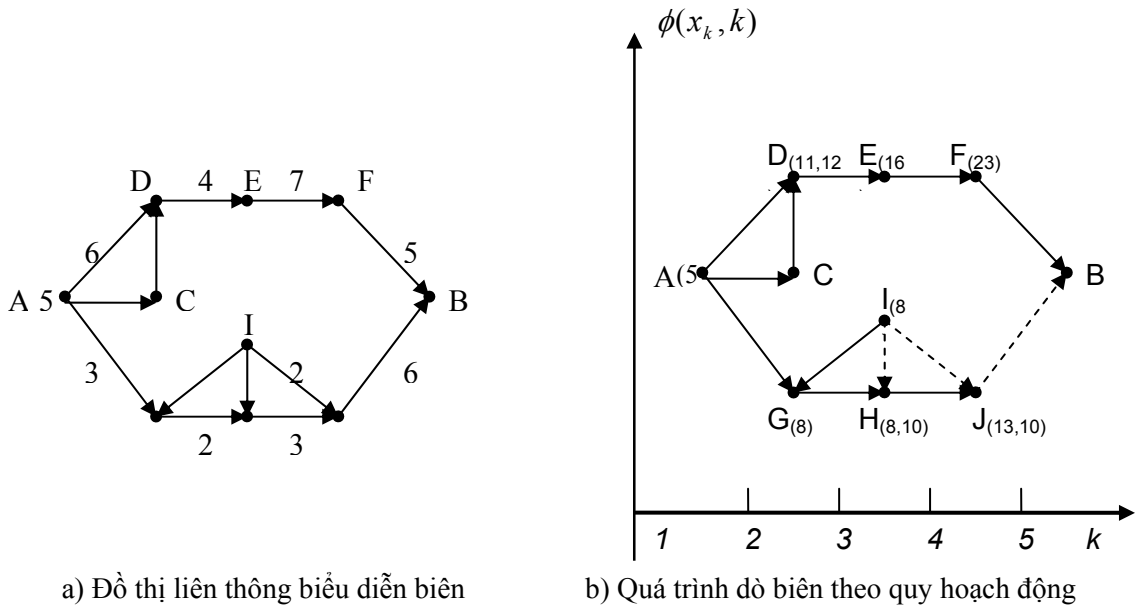
Lấy  $N = k$  theo (4-22) và (4-24) thực hiện suy diễn ta có:

$$\begin{aligned} \phi(x_k, k) &= \text{Max}_{x_1, \dots, x_{k-1}} \{S(x_1, \dots, x_k, k) + f(x_{k-1}, x_k)\} \\ &= \text{Max}_{x_1, \dots, x_{k-1}} \{\phi(x_{k-1}, k-1) + f(x_{k-1}, x_k)\} \end{aligned} \quad (4-25)$$

Như vậy,  $S(\overline{x_1}, \dots, \overline{x_N}, N) = \underset{XN}{\text{Max}} \{ \phi(x_N, N) \}$  với  $\phi(x, 1) = |g(x_1)|$

Với cách này, thay vì tìm tối ưu toàn cục phức tạp của  $S(x_1, \dots, x_N, N)$ , ta tìm tối ưu của  $N$  chặng theo tối ưu 2 biến. Trong mỗi chặng, với mỗi  $x_k$  tìm tối ưu  $\phi(x_k, k)$ . Để dễ hình dung, xét ví dụ sau:

Giả sử có bản đồ biểu diễn bởi đồ thị liên thông (Hình 4.11a). Theo phương pháp trên có  $\phi(A, 1) = 5$ , với  $k=2$  có  $\phi(D, 2) = \max(11, 12) = 12$ . Điều đó có nghĩa là đường từ  $A$  đến  $D$  đi qua  $C$  và  $ACD$  là biên được chọn với  $k=2$ . Tương tự, với  $k=4$ , có hai đường được chọn là  $ACDEF$  và  $AGHJ$ . Tuy nhiên, với  $k=5$  thì đoạn  $JB$  bị loại và chỉ tồn tại đường duy nhất với cực đại là 28. Như vậy, biên được xác định là  $ADEFB$ .



Hình 4.11. Dò biên theo phương pháp quy hoạch động.

Trên hình 4.11b, những đường nét đứt đoạn biểu thị cung bị loại; đường nét liền có mũi tên biểu thị đường đi hay biên của ảnh.

#### 4.2.3 Một số phương pháp khác

Ngoài các phương pháp trên, người ta cũng áp dụng một số phương pháp khác cải tiến như tiếp cận bởi mô hình mặt, cách tiếp cận tối ưu hóa.

Cách tiếp cận theo mô hình mặt dựa vào việc thực hiện xấp xỉ đa thức trên ảnh gốc hay ảnh đã thực hiện phép lọc Laplace. Cách tiếp cận tối ưu nhằm xác định một hàm (một bộ lọc), làm giảm phương sai  $\sigma^2$  hoặc giảm một số điểm cực trị cục bộ. Dưới đây sẽ trình bày một cách tóm tắt các phương pháp đó.

##### a. Tiếp cận theo mô hình mặt

Tư tưởng của phương pháp này là tại lân cận điểm cắt không (điểm biên), ảnh sau khi lọc Laplace có thể được xấp xỉ bởi một đa thức bậc 3 theo hàng và cột. Đa thức thường được dùng là đa thức Trebuchep với kích thước  $3 \times 3$ . Các đa thức này được định nghĩa như sau:

$$P_0(x,y)=1; P_1(x,y)=x; P_2(x,y)=y; P_3(x,y)=x^2 - \frac{2}{3}; P_4(x,y)=xy; P_5(x,y)=y^2 - \frac{2}{3};$$

$$P_6(x,y)=xP_5(x,y); P_7(x,y)=yP_3(x,y); P_8(x,y)=P_5(x,y)P_3(x,y)$$

Với mỗi điểm cắt không phát hiện tại  $P(x, y)$  trong ảnh đã được lọc bởi toán tử Laplace – Gauss, Huertas và Medioni đã cho được tính theo công thức tính xấp xỉ:

$$I_{(L-G)}(x, y) = \sum_{n=0}^{N-1} a_n P_n(x, y) \quad (4-26)$$

Vấn đề là xác định các hệ số  $a_i, i=1, 2, \dots, N-1$ . Nếu  $W$  là cửa sổ lọc tại điểm cắt không và  $x, y, i, j$  trong cửa sổ; các hệ số  $a_i$  có thể được tính toán như một tổ hợp tuyến tính:

$$a_n = \frac{\sum_x \sum_y P_n(x, y) I_{L-G}(x, y)}{\sum_i \sum_j P_n^2(i, j)} \quad (4-27)$$

ở đây,  $I_{L-G}(x, y)$  là ký hiệu ảnh đã được lọc bởi toán tử Laplace–Gauss. Các hệ số này có thể nhận được bởi chập ảnh  $I_{L-G}(x, y)$  với các nhân chập như trung bình có trọng số hay một số nhân chập khác.

Các bước cài đặt phương pháp này có thể mô tả như sau:

1. Chập ảnh gốc kích thước  $N \times M$  với toán tử Laplac –Gauss kích thước  $M^2$ , ảnh thu được gọi là  $I_{L-G}$ .
2. Trích chọn các điểm cắt không của ảnh  $I_{L-G}$ , ảnh kết quả ký hiệu là  $I_{ZCR}$ .
3. Với mỗi điểm cắt không trong  $I_{ZCR}$ , thực hiện một xấp xỉ với kích thước  $3 \times 3$  để suy ra các điểm cắt không theo cách giải tích.
4. Tạo một ảnh mới của các điểm cắt không kích thước  $n \times n \times Y$  mà các đường bao được xác định với độ phân giải  $n$  nào đó.

*b. Tiếp cận tối ưu hóa*

Ý tưởng của cách tiếp cận này là định vị đúng vị trí bằng cách cực tiểu hóa phương sai  $\sigma^2$  vị trí các điểm cắt không hoặc hạn chế số điểm cực trị cục bộ để chỉ tạo ra một đường bao. Canny đã đề xuất 3 ràng buộc ứng với 3 điều kiện:

$$\sum = \frac{A \int_{-\infty}^0 h(x) dx}{\sqrt[n_0]{\int_{-\infty}^{\infty} h^2(x) dx}} \quad (4-28)$$

$$\Lambda = \frac{A|h'(0)|}{\sqrt[n_0]{\int_{-\infty}^{\infty} h^2(x) dx}} \quad (4-29)$$

$$x_{\max} = 2\pi \sqrt{\frac{\int_{-\infty}^{\infty} h'^2(x) dx}{\int_{-\infty}^{\infty} h''^2(x) dx}} \quad (4-30)$$

Ràng buộc đầu tiên ( $\Sigma$ ) nhằm tìm hàm  $h(x)$  phản đối xứng sao cho tỉ số giữa tín hiệu và nhiễu là cực đại. Ràng buộc thứ hai ( $\Lambda$ ) nhằm cực tiểu hóa phương sai. Ràng buộc thứ ba nhằm hạn chế điểm cực trị cục bộ với mục đích cung cấp chỉ một đường bao.

## CÂU HỎI VÀ BÀI ÔN TẬP

- 1, Hãy viết biểu thức tính đạo hàm bậc hai một theo mặt nạ.
- 2, Cho ảnh số I:

$$I = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 \\ 1 & 5 & 5 & 7 & 3 \\ 11 & 5 & 5 & 7 & 3 \\ 1 & 5 & 7 & 7 & 6 \\ 1 & 8 & 7 & 6 & 8 \end{bmatrix}$$

Hãy tính  $G = |G_x| + |G_y|$  với  $G_x = H_x \otimes I$  và  $G_y = H_y \otimes I$ ,  $H_x, H_y$  là nhân chập Prewitt.

- 3, Xây dựng thuật toán phát hiện biên ảnh dùng toán tử Robert.
- 4, Trình bày phương pháp phát hiện biên ảnh dùng toán tử Sobel.
- 5, Trình bày phương pháp phát hiện biên ảnh dùng toán tử Prewitt.
- 6, Trình bày phương pháp tính Laplace của ảnh đã cho với kiểu mặt nạ tự chọn
- 7, Trình bày phương pháp dò biên theo Canny.
- 8, Xây dựng thủ tục dò biên theo quy hoạch động dựa vào thuật toán đã cho.

## CHƯƠNG 5: PHÂN VÙNG ẢNH

**Học xong phần này sinh viên có thể nắm được:**

1. Mục đích, tầm quan trọng của phân vùng ảnh trong xử lý và phân tích ảnh số. Các phương pháp phân vùng ảnh.
2. Phương pháp phân vùng ảnh theo ngưỡng biên độ
3. Phương pháp phân vùng ảnh theo miền đồng nhất
4. Phương pháp phân vùng theo kết cấu bề mặt và tổng quát về một vài phương pháp khác.

### 5.1 GIỚI THIỆU

Phân vùng ảnh là bước then chốt trong xử lý ảnh. Giai đoạn này nhằm phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng mức xám, cùng màu hay cùng độ nhám... Trước hết cần làm rõ khái niệm "vùng ảnh" (Segment) và đặc điểm vật lý của vùng.

Vùng ảnh là một chi tiết, một thực thể trong toàn cảnh. Nó là một tập hợp các điểm có cùng hoặc gần cùng một tính chất nào đó : mức xám, mức màu, độ nhám... Vùng ảnh là một trong hai thuộc tính của ảnh. Nói đến vùng ảnh là nói đến tính chất bề mặt. Đường bao quanh một vùng ảnh (Boundary) là biên ảnh. Các điểm trong một vùng ảnh có độ biến thiên giá trị mức xám tương đối đồng đều hay tính kết cấu tương đồng.

Dựa vào đặc tính vật lý của ảnh, người ta có nhiều kỹ thuật phân vùng : phân vùng dựa theo miền liên thông gọi là phân vùng dựa theo miền đồng nhất hay miền kề ; phân vùng dựa vào biên gọi là phân vùng biên. Ngoài ra còn có các kỹ thuật phân vùng khác dựa vào biên độ, phân vùng dựa theo kết cấu.

### 5.2 PHÂN VÙNG ẢNH THEO NGƯỠNG BIÊN ĐỘ

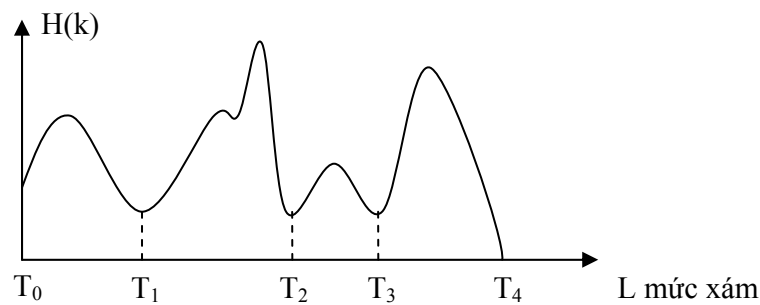
Các đặc tính đơn giản, cần thiết nhất của ảnh là biên độ và các tính chất vật lý như : độ tương phản, độ truyền sáng, màu sắc hoặc đáp ứng phổ.

Như vậy, có thể dùng ngưỡng biên độ để phân vùng khi biên độ đủ lớn đặc trưng cho ảnh. Thí dụ, biên độ trong bộ cảm biến ảnh hồng ngoại có thể phản ánh vùng có nhiệt độ thấp hay vùng có nhiệt độ cao. Kỹ thuật phân ngưỡng theo biên độ rất có lợi đối với ảnh nhị phân như văn bản in, đồ họa, ảnh màu hay ảnh X-quang.

Việc chọn ngưỡng rất quan trọng. Nó bao gồm các bước :

- Xem xét lược đồ xám của ảnh để xác định các đỉnh và các khe. Nếu ảnh có dạng rần lộn (nhiều đỉnh và khe), các khe có thể dùng để chọn ngưỡng.
- Chọn ngưỡng  $t$  sao cho một phần xác định trước  $\eta$  của toàn bộ số mẫu là thấp hơn  $t$ .
- Điều chỉnh ngưỡng dựa trên lược đồ xám của các điểm lân cận.

- Chọn ngưỡng theo lược đồ xám của những điểm thỏa mãn tiêu chuẩn chọn. Thí dụ, với ảnh có độ tương phản thấp, lược đồ của những điểm có biên độ Laplace  $g(m,n)$  lớn hơn giá trị  $t$  định trước (sao cho từ 5% đến 10% số điểm ảnh với Gradient lớn nhất sẽ coi như biên) sẽ cho phép xác định các đặc tính ảnh lưỡng cực tốt hơn ảnh gốc.
- Khi có mô hình phân lớp xác suất, việc xác định ngưỡng dựa vào tiêu chuẩn xác suất nhằm cực tiểu xác suất sai số hoặc dựa vào một số tính chất khác của luật Bayes.
- Để hiểu rõ hơn nguyên tắc phân vùng dựa vào ngưỡng biên độ, xét thí dụ sau:



**Hình 5.1** Lược đồ rần lược và cách chọn ngưỡng

Giả sử ảnh có lược đồ xám như Hình 5.1, chọn các ngưỡng như hình trên với:

$T_0 = L_{min}, \dots, T_4 = L_{max}$ . Ta có 5 ngưỡng và phân ảnh thành 4 vùng, ký hiệu  $C_k$  là vùng thứ  $k$  của ảnh,  $k=1,2,3,4$ . Cách phân vùng theo nguyên tắc :

$$P(m,n) \in C_k \text{ nếu } T_{k-1} \leq P(m,n) < T_k, k=1,2,3,4.$$

Khi phân vùng xong, nếu ảnh rõ nét thì việc phân vùng coi như kết thúc. Nếu không, cần điều chỉnh ngưỡng.

### 5.3 PHÂN VÙNG ẢNH THEO MIỀN ĐỒNG NHẤT

Kỹ thuật phân vùng ảnh thành các miền đồng nhất dựa vào các tính chất quan trọng nào đó của miền ảnh. Việc lựa chọn các tính chất của miền sẽ xác định tiêu chuẩn phân vùng. Tính đồng nhất của một miền ảnh là điểm chủ yếu xác định tính hiệu quả của việc phân vùng. Các tiêu chuẩn hay được dùng là sự thuần nhất về mức xám, màu sắc đối với ảnh màu, kết cấu sợi và chuyển động.

Các phương pháp phân vùng ảnh theo miền đồng nhất thường áp dụng là :

- Phương pháp tách cây tứ phân
- Phương pháp cục bộ
- Phương pháp tổng hợp

#### 5.3.1 Phương pháp tách cây tứ phân

Về nguyên tắc, phương pháp này kiểm tra tính đúng đắn của tiêu chuẩn đề ra một cách tổng thể trên miền lớn của ảnh. Nếu tiêu chuẩn được thỏa mãn, việc phân đoạn coi như kết thúc. Trong trường hợp ngược lại, chia miền đang xét thành 4 miền nhỏ hơn. Với mỗi miền nhỏ, áp

dùng một cách đệ quy phương pháp trên cho đến khi tất cả các miền đều thỏa mãn điều kiện. Phương pháp này có thể mô tả bằng thuật toán sau :

*Procedure PhanDoan(Mien)*

*Begin*

*If miền đang xét không thỏa Then*

*Begin*

*Chia miền đang xét thành 4 miền :  $Z_1, Z_2, Z_3, Z_4$*

*For  $i=1$  to  $4$  do PhanDoan ( $Z_i$ )*

*End*

*Else exit*

*End*

Tiêu chuẩn xét miền đồng nhất ở đây có thể dựa vào mức xám. Ngoài ra, có thể dựa vào độ lệch chuẩn hay độ chênh giữa giá trị mức xám lớn nhất và giá trị mức xám nhỏ nhất. Giả sử  $Max$  và  $Min$  là giá trị mức xám lớn nhất và nhỏ nhất trong miền đang xét. Nếu :

$$|Max - Min| < T \text{ (ngưỡng)}$$

ta coi miền đang xét là đồng nhất. Trường hợp ngược lại, miền đang xét không là miền đồng nhất và sẽ được chia làm 4 phần.

Thuật toán kiểm tra tiêu chuẩn dựa vào độ chênh lệch max, min được viết :

*Function Examin\_Criteria( $I, N_1, M_1, N_2, M_2, T$ )*

*/\* Giả thiết ảnh có tối đa 255 mức xám.*

*( $N_1, M_1$ ), ( $N_2, M_2$ ) là tọa độ điểm đầu và điểm cuối của miền;  $T$  là ngưỡng. \*/*

*Begin*

*1.  $Max=0$  ;  $Min=255$*

*2. For  $i = N_1$  to  $N_2$  do*

*If  $I[i,j] < Min$*

*Then  $Min=I[i,j]$  ;*

*If  $I[i,j] > Max$*

*Then  $Max=I[i,j]$  ;*

*3. If  $ABS(Max-Min) < T$*

*Then Examin\_Criteria=0*

*Else Examin\_Criteria=1 ;*

*End*

Nếu hàm trả về giá trị 0, có nghĩa vùng đang xét là đồng nhất, nếu không thì không đồng nhất. Trong giải thuật trên, khi miền là đồng nhất cần tính lại giá trị trung bình và cập nhật lại ảnh đầu ra. Giá trị trung bình được tính bởi :

$$\text{Tổng giá trị mức xám} / \text{tổng số điểm ảnh trong vùng}$$

Thuật toán này tạo nên một cây mà mỗi nút cha có 4 nút con ở mọi mức trừ mức ngoài cùng. Vì thế, cây này có tên là cây tứ phân. Cây cho ta hình ảnh rõ nét về cấu trúc phân cấp của các vùng tương ứng với tiêu chuẩn.

Một vùng thỏa mãn điều kiện sẽ tạo nên một nút lá; nếu không nó sẽ tạo nên một nút trong và có 4 nút con tương ứng. Tiếp tục như vậy cho đến khi phân chia xong để đạt các vùng đồng nhất..

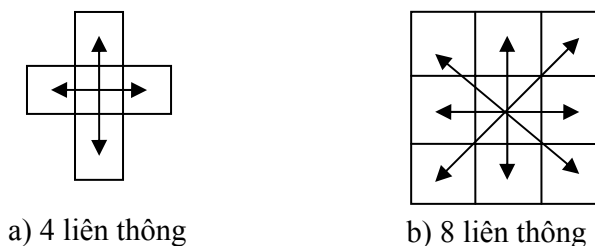
### 5.3.2 Phương pháp cục bộ

Ý tưởng của phương pháp là xét ảnh từ các miền nhỏ nhất rồi nối chúng lại nếu thỏa mãn tiêu chuẩn để được một miền đồng nhất lớn hơn. Tiếp tục với các miền thu được cho đến khi không thể nối thêm được nữa. Số miền còn lại cho ta kết quả phân đoạn. Như vậy, miền nhỏ nhất của bước xuất phát là điểm ảnh.

Phương pháp này hoàn toàn ngược với phương pháp tách. Song điều quan trọng ở đây là nguyên lý nối 2 vùng. Việc nối 2 vùng được thực hiện theo nguyên tắc sau :

- Hai vùng phải đáp ứng tiêu chuẩn, thí dụ như cùng màu hay cùng mức xám.
- Hai vùng phải kề cận nhau.

Khái niệm kề cận: trong xử lý ảnh, người ta dùng khái niệm liên thông để xác định tính chất kề cận. Có hai khái niệm về liên thông là 4 liên thông và 8 liên thông. Với 4 liên thông một điểm ảnh  $I(x,y)$  sẽ có 4 kề cận theo 2 hướng  $x$  và  $y$  ; trong khi đó với 8 liên thông, điểm  $I(x,y)$  sẽ có 4 liên thông theo 2 hướng  $x, y$  và 4 liên thông khác theo hướng chéo  $45^\circ$ .



**Hình 5.2 Khái niệm 4 liên thông và 8 liên thông**

Dựa theo nguyên lý của phương pháp nối, ta có 2 thuật toán :

- Thuật toán tô màu (Blob Coloring) : sử dụng khái niệm 4 liên thông, dùng một cửa sổ di chuyển trên ảnh để so sánh với tiêu chuẩn nối.
- Thuật toán đệ quy cục bộ: sử dụng phương pháp tìm kiếm trong một cây để làm tăng kích thước vùng.

### 5.3.3 Phương pháp tổng hợp

Hai phương pháp nối (hợp) và tách đều có nhược điểm. Phương pháp tách sẽ tạo nên một cấu trúc phân cấp và thiết lập mối quan hệ giữa các vùng. Tuy nhiên, nó thực hiện việc chia quá chi tiết. Phương pháp hợp cho phép làm giảm số miền liên thông xuống tối thiểu, nhưng cấu trúc hàng ngang dàn trải, không cho ta thấy rõ mối liên hệ giữa các miền.

Vì nhược điểm này, người ta nghĩ đến phối hợp cả 2 phương pháp. Trước tiên, dùng phương pháp tách để tạo nên cây tứ phân, phân đoạn theo hướng từ gốc đến lá. Tiếp theo, tiến hành duyệt cây theo chiều ngược lại và hợp các vùng có cùng tiêu chuẩn. Với phương pháp này ta thu được một cấu trúc ảnh với các miền liên thông có kích thước tối đa.

Giải thuật tách hợp gồm một số bước chính sau:



1. Kiểm tra tiêu chuẩn đồng nhất.

a) Nếu không thỏa mãn tiêu chuẩn đồng nhất và số điểm trong một vùng nhiều hơn 1, tách vùng ảnh làm 4 miền (trên, dưới, phải, trái) bằng cách đệ quy. Nếu kết quả tách xong và không tách được nữa chuyển sang bước 2.

b) Nếu tiêu chuẩn đồng nhất thỏa mãn thì tiến hành hợp vùng và cập nhật lại giá trị trung bình của vùng cho vùng này.

2. Hợp vùng

Kiểm tra 4 lân cận như đã nêu trên. Có thể có nhiều vùng thỏa mãn. Khi đó, chọn vùng tối ưu nhất rồi tiến hành hợp.

**5.4 PHÂN VÙNG ẢNH THEO KẾT CẤU BỀ MẶT**

Kết cấu thường được nhận biết trên bề mặt của các đối tượng như gỗ, cát, vải vóc... Kết cấu là thuật ngữ phản ánh sự lặp lại của các phần tử sợi (texel) cơ bản. Sự lặp lại này có thể ngẫu nhiên hay có tính chu kì hoặc gần chu kì. Một texel chứa rất nhiều điểm ảnh. Trong phân tích ảnh, kết cấu được chia làm hai loại chính là: loại thống kê và loại cấu trúc.

**5.4.1 Phương pháp thống kê**

Tính kết cấu ngẫu nhiên rất phù hợp với các đặc trưng thống kê. Vì vậy, người ta có thể dùng các đặc trưng ngẫu nhiên để đo nó như: Hàm tự tương quan (AutoCorrelation Function-ACF), các biến đổi mật độ gờ, ma trận tương tranh,...

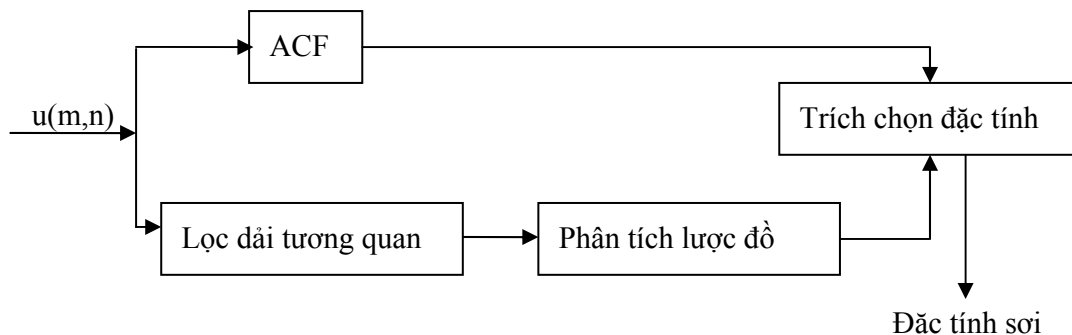
Theo cách tiếp cận bằng hàm tự tương quan, độ thô của kết cấu sợi tỉ lệ với độ rộng của ACF, được biểu diễn bởi khoảng cách  $x_0, y_0$  sao cho  $r(x_0, 0) = r(0, y_0) = 1$ . Người ta cũng dùng cách đo nhánh của ACF nhờ hàm khởi sinh moment :

$$M(k, l) = \sum_m \sum_n (m - \mu_1)^k (n - \mu_2)^l r(m, n) \tag{5.1}$$

với :

$$\mu_1 = \sum_m \sum_n mr(m, n) \text{ và } \mu_2 = \sum_m \sum_n nr(m, n)$$

Các đặc trưng của kết cấu sợi như độ thô, độ mịn hay hướng có thể ước lượng nhờ các biến đổi ảnh bằng kỹ thuật lọc tuyến tính. Một mô hình đơn giản trong trường hợp ngẫu nhiên cho việc phân tích tính kết cấu được mô tả trong hình dưới đây :



**Hình 5.4 Phân tích kết cấu sợi bằng dải tương quan**

Trong mô hình này, trường kết cấu sợi trước tiên được giải chấp bởi bộ lọc lấy từ đầu ra của ACF. Như vậy, nếu  $r(m,n)$  là ACF thì :

$$u(m,n) \otimes a(m,n) = \varepsilon(m,n) \quad (5.2)$$

là trường ngẫu nhiên không tương quan.

Lưu ý rằng, bộ lọc là không duy nhất, có thể là nhân quả, bán nhân quả hay không nhân quả. Các ACF hay dùng như  $M(0,2)$ ,  $M(2,0)$ ,  $M(1,1)$ ,  $M(2,2)$ . Các đặc trưng của lược đồ bậc một của  $\varepsilon(m,n)$  chẳng hạn như trung bình  $m_1$ , độ phân tán  $\sqrt{\mu_2}$  cũng hay được sử dụng.

Ngoài các đặc trưng trên, có thể đưa thêm một số khái niệm và định nghĩa các đại lượng dựa trên đó như: lược đồ mức xám (Histogram Grey Level Difference), ma trận xuất hiện mức xám (Grey Level Occurrence Matrices).

#### Lược đồ hiệu mức xám:

Lược đồ hiệu mức xám dùng để mô tả các thông tin mang tính không gian và được định nghĩa như sau. Cho  $d=(d_1, d_2)$  là vecto dịch chuyển giữa 2 điểm ảnh và  $g(d)$  là hiệu mức xám với khoảng cách  $d$  :

$$g(d) = |f(k,l) - f(k+d_1, l+d_2)| \quad (5.3)$$

với hàm  $f(k, l)$  cho giá trị mức xám tại tọa độ  $(k, l)$ . Gọi  $h_g(g, d)$  là lược đồ của hiệu mức xám khoảng cách  $d$ . Với mỗi khoảng cách  $d$  ta có một lược đồ mức xám riêng.

Với một miền ảnh có kết cấu thô, lược đồ  $h_g(g, d)$  có khuynh hướng tập trung xung quanh  $g=0$  với khoảng cách  $d$  nhỏ. Trái lại, với một miền ảnh có kết cấu mịn,  $h_g(g, d)$  sẽ phân nhánh dù với vecto dịch chuyển  $d$  khá nhỏ. Dựa trên lược đồ này, người ta định nghĩa lại một số đại lượng :

$$\text{- Trung bình :} \quad \mu_d = \sum_{k=1}^N g_k h_g(g_k, d) \quad (5.4)$$

$$\text{- Phương sai :} \quad \sigma_d^2 = \sum_{k=1}^N (g_k - \mu_d)^2 h_g(g_k, d) \quad (5.5)$$

$$\text{- Độ tương phản :} \quad c_d = \sum_{k=1}^N g_k^2 h_g(g_k, d) \quad (5.6)$$

Phương sai đo độ tản mát của hiệu mức xám tại một khoảng cách  $d$  nào đấy. Kết cấu tất định thường có phương sai  $\sigma_d$  tương đối nhỏ. Độ tương phản  $c_d$  chính là mômen của lược đồ  $h_g(g,d)$  xung quanh  $g=0$  và đo độ tương phản của hiệu mức xám.

Người ta sử dụng entropy để đo độ đồng nhất của lược đồ  $h_g$  :

$$H_g = -\sum_{k=1}^N h_g(g_k, d) \ln(h_g(g_k, d)) \quad (5.7)$$

Ưu điểm cơ bản của lược đồ hiệu mức xám là tính toán đơn giản. Ngoài ra còn có khả năng cho ta tổ chức kết cấu không gian.

#### Ma trận xuất hiện liên hiệp mức xám

Giả sử  $P(k,l,d)$  là xác suất liên hiệp của hai điểm ảnh  $f_k$  và  $f_l$  với các mức xám  $k,l$  tương ứng cách nhau một khoảng  $d$ . Xác suất này dễ dàng tính được nhờ việc tính số lần xuất hiện  $n_{k,l}$  của cặp điểm ảnh  $(f_k, f_l)$  có mức xám  $k$  và  $l$  với khoảng cách  $d$ . Gọi  $n$  là tổng số cặp liên hiệp có

thể với khoảng cách  $d$  trong ảnh. Các phần tử  $c_{k,l}$  của ma trận xuất hiện liên hiệp mức xám  $c_d$  được tính như sau :  $c_d = (c_{k,l})$

$$\text{Và} \quad c_{k,l} = P(k,l,d) = \frac{n_{k,l}}{n} \quad (5.8)$$

Ma trận xuất hiện liên hiệp mức xám  $C_d$  là ma trận vuông  $N \times N$  phần tử ( $N$  là số mức xám của ảnh). Ma trận này chứa các thông tin hữu ích về tổ chức kết cấu không gian. Nếu kết cấu tương đối thô thì các phần tử của ma trận tập trung xung quanh đường chéo chính. Ngược lại, nếu kết cấu bề mặt mịn, giá trị các phần tử của  $c_d$  sẽ phân rải tương đối rõ.

Dựa trên khái niệm này người ta định nghĩa về một số độ đo :

- Xác suất cực đại : 
$$p_d = \max_{(k,l)} C_{k,l} \quad (5.9)$$

Entropy :

$$H_d = - \sum_{k=1}^N \sum_{l=1}^N C_{k,l} \ln(C_{k,l}) \quad (5.10)$$

Để dàng thấy được entropy cực đại khi xác suất liên hiệp  $P(k,l,d)$  có phân phối đều.

Mô men bậc  $m$  :

$$I_d = \sum_{k=1}^N \sum_{l=1}^N |k-l|^m C_{k,l} \quad (5.11)$$

$I_d$  cực tiểu khi các phần tử của ma trận  $C$  tập trung trên đường chéo chính vì khoảng cách  $|k-l|^m$  rất nhỏ,  $I_d$  nhỏ có nghĩa là kết cấu khá thô. Người ta cũng còn đưa vào một số độ đo khác như hàm tự tương quan, phổ năng lượng.

Để áp dụng cách tiếp cận này, cần cài đặt các giải thuật tính các đại lượng đo trên.

### 5.4.2 Phương pháp cấu trúc

Kết cấu sợi có cấu trúc thuần nhất là những texels xác định, mà sự xuất hiện lặp đi lặp lại tuân theo một luật tất định hay ngẫu nhiên nào đấy. Một texel về thực tế là một nhóm các điểm ảnh có cùng một số tính chất bất biến lặp trên ảnh. Một texel cũng có định nghĩa theo mức xám, theo bề mặt hay tính đồng nhất đối với một số các tính chất như kích thước, hướng, lược đồ bậc hai (ma trận tương tranh).

Với các texel được phân bố ngẫu nhiên, tính kết cấu sợi tương ứng của nó được coi là yếu (Weak) ngược với qui luật phân bố tất định gọi là khỏe (Strong). Khi tính kết cấu sợi là yếu, luật phân bố có thể đo bởi:

- Mật độ gờ
- Các loạt dài của các texel liên thông tối đa
- Mật độ cực trị tương đối; số pixel trên một đơn vị diện tích có mức xám cực trị địa phương so với các lân cận.

Ngoài hai cách tiếp cận trên, người ta còn dùng cách tiếp cận khác bằng cách lấy tổ hợp 2 cách trên và gọi là kỹ thuật *mosaic*. Mô hình này biểu diễn các quá trình học ngẫu nhiên, thí dụ như khảm ngẫu nhiên hay đều của một mặt phẳng vào các đường cong lồi sẽ làm nổi lên tính kết cấu tế bào.

### 5.4.3 Tiếp cận theo tính kết cấu

Khi đối tượng xuất hiện trên một nền có tính kết cấu cao, việc phân đoạn dựa vào tính kết cấu trở nên quan trọng. Nguyên nhân là kết cấu sợi thường chứa mật độ cao các gờ (edge) làm cho phân đoạn theo biên kém hiệu quả, trừ phi ta loại tính kết cấu. Việc phân đoạn dựa vào miền đồng nhất cũng có thể áp dụng cho các đặc trưng kết cấu và có thể dùng để phân đoạn các miền có tính kết cấu.

Nhìn chung, việc phân loại và phân vùng dựa vào kết cấu là một vấn đề phức tạp. Ở đây, tài liệu chỉ mang tính chất giới thiệu. Có thể giải quyết vấn đề này trong thực tế nếu ta biết trước các loại kết cấu (dựa vào quy luật hay các phân bố của nó).

## CÂU HỎI ÔN TẬP CHƯƠNG 5

1. Thế nào là vùng ảnh ? Mục đích của phân vùng ảnh là gì ?
2. Thế nào là phân vùng ảnh theo ngưỡng biên độ ? Cho ví dụ ?
3. Thế nào là phân vùng ảnh theo miền đồng nhất ? Cho ví dụ ?
4. Trình bày phương pháp tách cây tứ phân để phân vùng ảnh ?
5. Trình bày phương pháp hợp để phân vùng ảnh ?
6. Trình bày phương pháp tổng hợp (tách và hợp) để phân vùng ảnh ?
7. Thế nào là phân vùng ảnh dựa vào kết cấu?

## CHƯƠNG 6: NHẬN DẠNG ẢNH

**Học xong phần này sinh viên có thể nắm được:**

1. Một số khái niệm, thuật ngữ, các phương pháp nhận dạng ảnh
2. Nhận dạng dựa theo miền không gian
3. Nhận dạng dựa theo cấu trúc
4. Nhận dạng dựa theo mạng Noron

### 6.1 GIỚI THIỆU

Nhận dạng ảnh là giai đoạn cuối của các hệ thống xử lý ảnh. Nhận dạng ảnh dựa trên lý thuyết nhận dạng (Pattern Recognition) đã được đề cập trong nhiều sách về nhận dạng. Trong lý thuyết về nhận dạng nói chung và nhận dạng ảnh nói riêng có ba cách tiếp cận khác nhau:

- Nhận dạng dựa vào phân hoạch không gian.
- Nhận dạng dựa vào cấu trúc.
- Nhận dạng dựa vào kỹ thuật mạng noron.

Hai cách tiếp cận đầu là cách tiếp cận kinh điển. Các đối tượng ảnh quan sát và thu nhận được phải trải qua giai đoạn tiền xử lý nhằm tăng cường chất lượng, làm nổi các chi tiết, tiếp theo là trích chọn và biểu diễn các đặc trưng, cuối cùng mới là giai đoạn nhận dạng. Cách tiếp cận thứ ba hoàn toàn khác. Nó dựa vào cơ chế đoán nhận, lưu trữ và phân biệt đối tượng mô phỏng theo hoạt động của hệ thần kinh con người. Do cơ chế đặc biệt, các đối tượng thu nhận bởi thị giác người không cần qua giai đoạn cải thiện mà chuyển ngay sang giai đoạn tổng hợp, đối sánh với các mẫu đã lưu trữ để nhận dạng. Đây là cách tiếp cận đầy hứa hẹn được trình bày cụ thể trong các phần dưới đây.

#### **Khái niệm nhận dạng.**

Nhận dạng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó và gán chúng một tên (gán cho đối tượng một tên gọi, tức là một dạng) dựa theo những quy luật và mẫu chuẩn. Quá trình nhận dạng dựa vào những mẫu học biết trước gọi là *nhận dạng có thầy* hay học có thầy, trong những trường hợp ngược lại gọi là *học không có thầy*. Chúng ta sẽ lần lượt giới thiệu các khái niệm này.

#### **6.1.1 Không gian biểu diễn đối tượng, không gian diễn dịch**

##### **a) Không gian biểu diễn đối tượng**

Các đối tượng khi quan sát hay thu thập được thường được biểu diễn bởi tập các đặc trưng hay đặc tính. Giả sử đối tượng ảnh  $X$  (ảnh, chữ viết, dấu vân tay,...) được biểu diễn bởi  $n$  thành phần ( $n$  đặc trưng):  $X = \{x_1, x_2, \dots, x_n\}$ ; mỗi  $x_i$  biểu diễn một đặc tính. Không gian biểu diễn thường được gọi tắt là không gian đối tượng  $\mathcal{X}$  được định nghĩa:

$$\mathcal{X} = \{X_1, X_2, \dots, X_m\}$$

Trong đó mỗi  $X_i$  biểu diễn một đối tượng. Không gian này có thể là vô hạn. Để tiện xem xét chúng ta chỉ xét tập  $\mathcal{X}$  là hữu hạn.

### b) Không gian diễn dịch

Không gian diễn dịch là tập các tên gọi của đối tượng. Kết thúc quá trình nhận dạng ta xác định được tên gọi cho các đối tượng. Một cách hình thức gọi  $\Omega$  là tập tên đối tượng:

$\Omega = \{w_1, w_2, \dots, w_k\}$  với  $w_i, i=1, 2, \dots, k$  là tên các đối tượng.

Quá trình nhận dạng đối tượng  $f$  là một ánh xạ  $f: \mathcal{X} \rightarrow \Omega$  với  $f$  là tập các quy luật để định một phần tử trong  $\mathcal{X}$  ứng với một phần tử trong  $\Omega$ . Nếu tập các quy luật và tập tên các đối tượng là biết trước như trong nhận dạng chữ viết (có 26 lớp từ A đến Z), người ta gọi là nhận dạng có thầy. Trường hợp thứ hai là nhận dạng không có thầy.

## 6.1.2 Mô hình và bản chất của quá trình nhận dạng

### a) Mô hình

Trong nhận dạng người ta chia thành hai họ lớn:

- Họ mô tả theo tham số.
- Họ mô tả theo cấu trúc.

Cách mô tả được lựa chọn sẽ xác định mô hình của đối tượng. Như vậy, chúng ta sẽ có hai loại mô hình: *mô hình tham số* và *mô hình cấu trúc*.

**Mô hình tham số:** sử dụng một vectơ để đặc tả đối tượng. Mỗi phần tử của vectơ mô tả một đặc tính của đối tượng. Thí dụ như trong các đặc trưng chức năng, người ta sử dụng các hàm cơ sở trực giao để biểu diễn.

Việc lựa chọn phương pháp biểu diễn sẽ làm đơn giản cách xây dựng. Tuy nhiên việc lựa chọn đặc trưng nào là hoàn toàn phụ thuộc vào ứng dụng. Thí dụ, trong nhận dạng chữ, các tham số là các dấu hiệu:

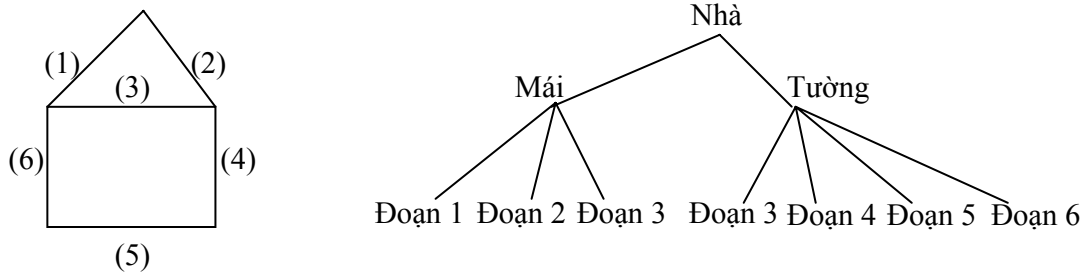
- Số điểm chạc ba, chạc tư.
- Số điểm chu trình.
- Số điểm ngoặt.
- Số điểm kết thúc.

**Mô hình cấu trúc:** Cách tiếp cận trong mô hình này dựa vào việc mô tả đối tượng nhờ một số khái niệm biểu thị các đối tượng cơ sở trong ngôn ngữ tự nhiên. Để mô tả đối tượng người ta dùng một số dạng nguyên thủy như đoạn thẳng, cung... Chẳng hạn một hình chữ nhật được định nghĩa gồm 4 đoạn thẳng vuông góc với nhau từng đôi một. Trong mô hình này người ta sử dụng một bộ ký hiệu kết thúc  $V_t$ , một bộ ký hiệu không kết thúc gọi là  $V_n$ . Ngoài ra có dùng một tập các luật sản xuất để mô tả cách xây dựng các đối tượng phù hợp dựa trên các đối tượng đơn giản hơn hoặc đối tượng nguyên thủy (tập  $V_t$ ). Trong cách tiếp cận này, ta chấp nhận khẳng định là: cấu trúc một dạng là kết quả của việc áp dụng luật sản xuất theo những nguyên tắc xác định bắt đầu từ một dạng gốc ban đầu. Một cách hình thức, ta có thể coi mô hình này tương đương một văn phạm  $G=(V, V_n, P, S)$  với:

- $V_t$  là bộ ký hiệu kết thúc;
- $V_n$  là bộ ký hiệu không kết thúc;

- $P$  là luật sản xuất;
- $S$  là dạng (ký hiệu bắt đầu).

Thí dụ, đối tượng nhà gồm mái và tường, mái là một tam giác gồm 3 cạnh là 3 đoạn thẳng, tường là một hình chữ nhật gồm 4 cạnh vuông góc với nhau từng đôi một sẽ được mô tả thông qua cấu trúc mô tả dựa vào văn phạm sinh như chỉ trong hình dưới đây:



**Hình 6.1** Mô hình cấu trúc của đối tượng nhà

**b) Bản chất**

Quá trình nhận dạng gồm 3 giai đoạn chính:

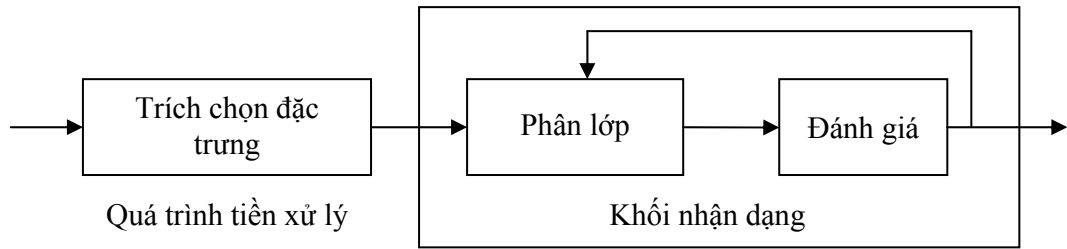
- Chọn mô hình biểu diễn đối tượng.
- Chọn luật ra quyết định (phương pháp nhận dạng) và suy diễn.
- Học trong nhận dạng.

Trong việc lựa chọn để biểu diễn đối tượng, đối tượng có thể được xác định theo cách định lượng (mô hình tham số) hay định tính (mô hình cấu trúc). Khi đối tượng đã được xác định, quá trình nhận dạng chuyển sang giai đoạn thứ hai-giai đoạn học (Learning). Học là giai đoạn cung cấp tri thức cho hệ thống. Mục đích học nhằm cải thiện, điều chỉnh việc phân loại tập đối tượng thành các lớp. Nhận dạng là tìm ra quy luật và các thuật toán để có thể gắn đối tượng vào một lớp hay nói một cách khác gán cho đối tượng một tên.

**Học có thầy:** kỹ thuật phân loại nhờ kiến thức biết trước gọi là học có thầy. Đặc điểm cơ bản của kỹ thuật này là người ta có một thư viện các mẫu chuẩn. Mẫu cần nhận dạng sẽ được đem so sánh với mẫu chuẩn để xem nó thuộc loại nào. Vấn đề chủ yếu là thiết kế một hệ thống để có thể đối sánh đối tượng trong ảnh với mẫu chuẩn và quyết định gán cho chúng vào một lớp. Việc đối sánh nhờ vào các thủ tục ra quyết định dựa trên một công cụ gọi là hàm phân lớp hay hàm ra quyết định.

**Học không có thầy:** kỹ thuật này phải tự định ra các lớp khác nhau và xác định các tham số đặc trưng cho từng lớp. Học không có thầy đương nhiên là gặp khó khăn hơn. Một mặt, do số lớp không được biết trước, mặt khác những đặc trưng của lớp cũng không được biết trước. Kỹ thuật này nhằm tiến hành mọi cách gộp nhóm có thể và chọn lựa cách tốt nhất. Bắt đầu từ tập dữ liệu, nhiều thủ tục xử lý khác nhau nhằm phân lớp và nâng cấp dần để đạt được một phương án phân loại.

Nhìn chung, dù là mô hình nào và kỹ thuật nhận dạng ra sao, một hệ thống nhận dạng có thể tóm tắt theo sơ đồ sau:



Hình 6.2 Sơ đồ tổng quát hệ thống nhận dạng ảnh

## 6.2 NHẬN DẠNG DỰA THEO MIỀN KHÔNG GIAN

Trong kỹ thuật này, các đối tượng nhận dạng là các đối tượng định lượng. Mỗi đối tượng được biểu diễn bởi một vectơ nhiều chiều. Trước tiên, ta xem xét một số khái niệm như: phân hoạch không gian, hàm phân biệt sau đó sẽ đi vào một số kỹ thuật cụ thể.

### 6.2.1 Phân hoạch không gian

Giả sử không gian đối tượng  $\mathcal{X}$  được định nghĩa:  $\mathcal{X} = \{X_i, i=1, 2, \dots, m\}$ , với  $X_i$  là một vectơ. Người ta nói  $\mathcal{D}$  là một phân hoạch của không gian  $\mathcal{X}$  thành các lớp  $C_i, C_i \subset \mathcal{X}$  nếu:

$$C_i \cap C_j = \Phi \text{ với } i \neq j \text{ và } \cup C_i = \mathcal{X} \quad (6-1)$$

Đây là trường hợp lý tưởng khi tập  $\mathcal{X}$  tách được hoàn toàn. Trong thực tế, thường gặp không gian biểu diễn tách được từng phần. Như vậy, phân loại là dựa vào việc xây dựng một ánh xạ  $f: \mathcal{X} \rightarrow \mathcal{D}$ . Công cụ xây dựng ánh xạ này là các hàm phân biệt (Discriminant Functions).

### 6.2.2 Hàm phân lớp hay hàm ra quyết định

Để chia đối tượng thành các lớp, cần xác định số lớp và ranh giới giữa các lớp đó. Gọi  $\{g_i\}$  là lớp các hàm phân lớp hay hàm tách biệt. Lớp hàm này được định nghĩa như sau:

Nếu  $\forall i \neq k, g_k(X) > g_i(X)$  thì ta quyết định  $X \in$  lớp  $k$ .

Như vậy để phân biệt  $k$  lớp ta cần  $k-1$  hàm phân biệt. Hàm phân biệt  $g(\cdot)$  của một lớp nào đó thường được dùng trong thực tế do tính đơn giản, dễ xử lý là hàm tuyến tính. Hàm tuyến tính có dạng:

$$g(X) = W_0 + W_1 X_1 + W_2 X_2 + \dots + W_k X_k \quad (6-2)$$

trong đó:

- $W_i$  là trọng số gán cho các thành phần  $X_i$ ,
- $W_0$  là trọng số hằng.

Trong trường hợp hàm  $g(\cdot)$  là tuyến tính, người ta nói việc phân lớp là tuyến tính (trong trường hợp một hay hai chiều) hay siêu phẳng (trong trường hợp nhiều chiều). Các hàm phân biệt thường được xây dựng dựa trên khái niệm khoảng cách hay dựa vào xác suất có điều kiện.

*Phân lớp dựa theo khoảng cách (Distance)* là một công cụ tốt để xác định đối tượng có “gần nhau” về một đặc trưng nào đó hay không. Nếu khoảng cách nhỏ hơn một ngưỡng  $\tau$  nào đấy thì ta coi hai đối tượng là giống nhau. Nếu chúng giống nhau ta gộp gộp chúng, nếu chúng khác nhau và ta tách thành hai hoặc nhiều lớp phân biệt.



*Phân lớp dựa theo xác suất có điều kiện (Conditional Probability)*. Trong một số trường hợp, người ta dựa vào xác suất có điều kiện để phân lớp cho đối tượng. Lý thuyết xác suất có điều kiện được Bayes nghiên cứu khá kỹ lưỡng và được dùng để phân biệt đối tượng.

### 6.2.3 Nhận dạng theo phương pháp thống kê

Nếu các đối tượng nhận dạng tuân theo luật phân bố Gauss, với hàm mật độ xác suất:

$$f(x) = \frac{1}{2\pi\sigma^2} \exp - \frac{(x-m)^2}{2\pi\sigma^2} \quad (6-3)$$

trong đó  $m$  là kỳ vọng,  $\sigma$  là độ lệch chuẩn.

Người ta có dùng phương pháp ra quyết định dựa vào lý thuyết Bayes. Lý thuyết Bayes thuộc loại lý thuyết thống kê nên phương pháp nhận dạng dựa trên lý thuyết Bayes có tên là phương pháp thống kê.

#### Quy tắc Bayes:

Giả sử cho: - không gian đối tượng  $\mathcal{X} = \{X_l, l=1,2,\dots,L\}$ ,

- không gian diễn dịch  $\Omega = \{C_1, C_2 \dots C_r\}$ , với  $r$  là số lớp

Giả sử tồn tại một sai số  $\varepsilon$  trong kết quả nhận dạng, khi đó quy tắc Bayes được phát biểu:

$\varepsilon: \mathcal{X} \rightarrow \Omega$  sao cho  $X \in C_k$  nếu  $P(C_k/X) > P(C_l/X) \quad \forall l \neq k, l=1,2,\dots,r$ .

ở đây:  $P(C_k/X)$  là xác suất của  $C_k$  trong điều kiện  $X$  xảy ra. Tương tự đối với  $P(C_l/X)$ .

Trường hợp lý tưởng là nhận dạng đúng (không có sai số). Thực tế, luôn tồn tại sai số  $\varepsilon$  trong quá trình nhận dạng. Vấn đề chính ở đây là xây dựng quy tắc nhận dạng với sai số  $\varepsilon$  là nhỏ nhất.

#### Phương pháp ra quyết định với $\varepsilon$ tối thiểu:

Cần xác định  $X \in C_k$  nhờ xác suất  $P(C_k/X)$ . Nếu có sai số sẽ được tính bởi  $1-P(C_k/X)$ . Để đánh giá sai số trung bình, người ta xây dựng một ma trận  $L(r,r)$  với giả thiết có  $n$  lớp.

Ma trận  $L$  được định nghĩa như sau:

$$L_{k,j} = \begin{cases} l_{k,j} > 0 & \text{khi } k \neq j \\ l_{k,j} \leq 0 & \text{khi } k = j \end{cases} \quad (6-4)$$

Như vậy, sai số trung bình của sự phân lớp sẽ là:

$$r_k(X) = \sum_{j=1}^r l_{k,j} P(C_j / X) \quad (6-5)$$

Để sai số là nhỏ nhất ta cần có  $r_k$  là nhỏ nhất (*min*). Từ lý thuyết xác suất ta có công thức tính xác suất có điều kiện (Công thức Bayes):

$$P(C_j / X) = \frac{P(X / C_j)P(C_j)}{P(X)} \quad (6-6)$$

Từ công thức (6-5) và (6-6) suy ra:

$$r_k(X) = \sum_{j=1}^r l_{k,j} P(X / C_j)P(C_j) \quad (6-7)$$

Vậy, quy tắc ra quyết định dựa trên lý thuyết Bayes có tính đến sai số được phát biểu như sau:

$$X \in C_k \text{ nếu } p_k < p_p \text{ với } p < k, p=1,2\dots r \quad (6-8)$$

với  $p_k$  là  $r_k(X)$  được xác định theo (6.7). Rõ ràng, từ điều kiện  $p_k < p_p$  ta hoàn toàn xác định đối tượng  $X$  thuộc lớp  $C_k$  nào. Đây chính là nội dung tư tưởng của phương pháp thống kê.

### 6.2.4 Thuật toán nhận dạng dựa vào khoảng cách

Có nhiều thuật toán nhận dạng học không có thầy. Ở đây, chúng ta xét thuật toán học (Learning Algorithm) căn cứ vào khoảng cách lớn nhất.

#### a. Nguyên tắc

Giả sử có tập gồm  $m$  đối tượng. Xác định khoảng cách giữa các đối tượng và khoảng cách lớn nhất ứng với phần tử xa nhất tạo nên lớp đối tượng mới. Việc phân lớp được tạo nên dần dần dựa vào thủ tục xác định khoảng cách giữa các đối tượng và các lớp. Điều này có thể minh họa bằng thuật toán sau:

#### b. Thuật toán

*Bước 1:*

- Chọn hạt nhân ban đầu. Giả sử  $X_1 \in C_1$  gọi là lớp  $g_1$ .  
Gọi  $Z_1$  là phần tử trung tâm của  $g_1$ .
- Tính tất cả các khoảng cách  $D_{j1}=D(X_j, Z_1)$  với  $j=1,2\dots m$
- Tìm  $D_{k1}=\max_j D_{jk}$ , trong đó  $X_k$  là phần tử xa nhất của nhóm  $g_1$ .  
Như vậy,  $X_k$  là phần tử trung tâm của lớp mới  $g_2$ . Kí hiệu  $Z_2$ .
- Tính  $d_1=D_{12}=D(Z_1, Z_2)$ .

*Bước 2:*

Tính các khoảng cách  $D_{j1}, D_{j2}$  với  
 $D_{j1}=D(X_j, Z_1); D_{j2}=D(X_j, Z_2)$ . Đặt  $D_k^{(2)} = \max_j D_j$

*Nguyên tắc chọn:*

- Nếu  $D_k^{(2)} < \theta d_k$ , với  $\theta$  là ngưỡng cho trước.  
Kết thúc thuật toán. Việc phân lớp kết thúc;
- Nếu không, tạo nhóm thứ ba. Gọi  $X_3$  là phần tử trung tâm của  $g_3$ , kí hiệu  $Z_3$ ;
- Tính  $D_3=(D_{12}+D_{13}+D_{23});$   
 $D_{13}=D(Z_1, Z_3);$   
 $D_{23}=D(Z_2, Z_3).$
- Quá trình lặp lại cho đến khi phân xong.  
Kết quả thu được các lớp đại diện  $Z_1, Z_2, \dots, Z_m$ .

## 6.3 NHẬN DẠNG DỰA THEO CẤU TRÚC

### 6.3.1 Biểu diễn định tính

Ngoài cách biểu diễn định lượng (theo tham số) như đã mô tả ở trên, tồn tại nhiều kiểu đối tượng mạng tính định tính (theo cấu trúc). Trong cách biểu diễn này, người ta quan tâm đến các dạng và mối quan hệ giữa chúng. Giả thiết rằng, mỗi đối tượng được biểu diễn bởi một dãy ký tự, các đặc tính biểu diễn bởi cùng một số ký tự. Phương pháp nhận dạng ở đây là nhận dạng logic, dựa vào hàm phân biệt là hàm Bool. Cách nhận dạng là nhận dạng các từ có cùng độ dài.

Giả sử hàm phân biệt cho mọi ký hiệu là  $g_a(X)$ ,  $g_b(X)$ , ... tương ứng với các ký hiệu  $a$ ,  $b$ , ... Để dễ dàng hình dung, ta giả sử có từ 'abcd' được biểu diễn bởi một dãy ký tự  $X = \{x_1, x_2, x_3, x_4\}$ , khi đó hàm phân biệt tương ứng nhận được là:

$$g_a(x_1) + g_b(x_2) + g_c(x_3) + g_d(x_4)$$

Các phép cộng ở đây có thể áp dụng toán tử OR. Trên cơ sở tính giá trị cực đại của hàm phân biệt, việc xác định (quyết định)  $X$  có thuộc lớp các từ "abcd" hay không. Trong cách tiếp cận này, đối tượng của ta có thể xem là tương đương với một câu hay một mệnh đề.

Thủ tục phân loại và nhận dạng ở đây gồm hai giai đoạn:

- *Giai đoạn 1*: xác định các quy tắc xây dựng, tương đương với việc nghiên cứu một văn phạm trong một ngôn ngữ chính thống
- *Giai đoạn 2*: xem xét tập các dạng trong không gian mẫu có được sinh ra hoàn toàn từ các dạng cơ bản đó không. Nếu nó thuộc tập đó thì coi như đã phân loại xong.

Tuy nhiên, ở phương pháp này, văn phạm là một vấn đề lớn khá phức tạp và khó có thể tìm được loại phù hợp một cách hoàn hảo với mọi đối tượng. Vì vậy, trong nhận dạng dự theo cấu trúc, ta chỉ sử dụng được một phần rất nhỏ.

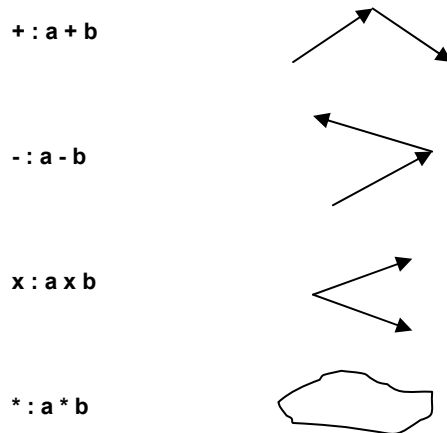
Như đã trình bày trong phần các mô hình biểu diễn mẫu. Mô hình cấu trúc tương đương với một văn phạm  $G = \{V1, Vn, P, S\}$ . Ngoài ra còn có rất nhiều văn phạm khác nhau từ chính tắc đến phi ngữ cảnh. Một văn phạm sẽ được sử dụng trong nhận dạng bởi một ngôn ngữ hình thức, trong đó có một ngôn ngữ điển hình cho nhận dạng cấu trúc là PLD (Picture Language Description).

Trong ngôn ngữ PLD, các từ vựng là các vạch có hướng. Có bốn từ vựng cơ bản:



**Hình 6.3 Các từ vựng cơ bản của ngôn ngữ hình thức PLD**

Các phép toán cho các từ vựng trên được định nghĩa như sau:



**Hình 6.4. Các phép toán trong ngôn ngữ LCD**

Văn phạm sinh ra các mô tả trong ngôn ngữ PLD được định nghĩa như sau:

$$GA = \{Vn, VT, P, S\}$$

Với  $Vn = \{A, B, C, D, E\}$  và  $VT = \{a, b, c, d\}$ .  $S$  là ký hiệu bắt đầu và  $P$  là tập luật sản xuất.

### 6.3.2 Các bước nhận dạng

Các đối tượng cần được nhận dạng theo phương pháp này được biểu diễn bởi một câu trong ngôn ngữ, gọi là  $L(G)$ . Khi đó thao tác phân lớp chính là xem xét một đối tượng có thuộc văn phạm  $L(G)$  không. Nói cách khác, nó có được sinh ra bởi các luật của văn phạm  $G$  hay không. Như vậy các bước cần phải thực hiện là:

- Xác định tập  $V_l$  chung cho tất cả mọi đối tượng
- Xác định các quy tắc  $P$  để sản sinh ra một câu và chúng khác nhau đối với mỗi lớp
- Thực hiện quá trình học với các câu biểu diễn các đối tượng mẫu  $l$  nhằm xác định văn phạm  $G$ .
- Ra quyết định: xác định một đối tượng  $X$  được biểu diễn bởi một câu  $l_x$ . Nếu  $l_x$  nhận biết bởi  $L(G_k)$  thì ta nói rằng  $X$  là một đối tượng thuộc loại  $C_k$ .

Nói cách khác, việc ra quyết định phân lớp dựa vào phân tích câu  $G_k$  biểu diễn lớp  $C_k$ . Việc nhận dạng dựa theo cấu trúc vẫn còn là một ý tưởng và còn nhiều điều cần nghiên cứu.

## 6.4 NHẬN DẠNG DỰA THEO MẠNG NƠN

### 6.4.1 Giới thiệu mạng nơon

Nghiên cứu về nơon thần kinh từ lâu đã trở thành đề tài được nhiều nhà khoa học quan tâm. Nhưng kỷ nguyên của mạng nơon chính thức được bắt đầu với báo cáo khoa học của McCulloch và Pitts năm 1943 miêu tả một phép tính logic của mạng nơon. Báo cáo này được công chúng đón nhận cho đến năm 1949 học thuyết về mạng nơon chính thức của McCulloch và Pitts được mô tả chủ yếu trong bài giảng thứ hai trong bốn bài giảng mà Von Neumann đã phát biểu tại trường đại học Illinois. Sự phát triển tiếp theo của quá trình nghiên cứu mạng nơon được đánh dấu vào năm 1949 với việc xuất bản cuốn sách “*The Organization of Behavior: A neuropsychological Theory*” của Donald Olding Hebb. Đến năm 1952, cuốn sách của Ashby “*Design for a Brain*” đã mô tả những điều kiện cần và đủ đối với một hệ thống hoạt động giống như bộ não “đó là phải học để tồn tại trong môi trường luôn thay đổi và nhận được những cái nó cần”. Tiếp đó, năm 1954, Minsky đã viết luận án tiến sĩ mang tên “*Theory of Neural-analog Reinforcement Systems and Application to Brain-Model Problem*” tại trường Đại học Princeton, sau đó là bài báo của ông “*Steps Toward Artificial Interligence*” năm 1961 về việc học củng cố trong mạng nơon hiện nay. Một chủ đề khác được đánh giá cao là phát kiến về bộ nhớ liên kết của Taylor vào năm 1956, mở đầu một loạt các kết quả phát triển to lớn về sau. Các kết quả có thể kể đến là sự ra đời của mạng Perceptron được Frank Rosenblatt công bố vào năm 1957 và được coi là “mạng nơon truyền thẳng đơn giản nhất”. Tiếp đó năm 1960, mạng nơon khác được Bernard Widrow và Marcian Hoff giới thiệu là ADALINE (ADaptive LINEar Element). Với mạng ADALINE lần đầu tiên kiểu hội tụ các mạch con chứa trọng số trước node tổng được sử dụng để phân lớp các mẫu. Năm 1969, Minsky và Papert xuất bản cuốn “*Perceptron, An*

*Introduction to Computational Geometry*” chỉ ra những giới hạn trong mạng Perceptron một lớp đơn và đề nghị khắc phục trong mạng Perceptron nhiều lớp. Những năm 1970 nổi bật với các sự kiện: các mạng liên kết của Kohonen và Anderson (1972), Cognitron – mạng tự tổ chức nhiều lớp đầu tiên do Kunihiko Fukushima người Nhật giới thiệu năm 1975. Thời kì phát triển vượt bậc của quá trình nghiên cứu mạng nơron là những năm 80 với sự ra đời của một loạt các mạng nơron có giá trị: mạng Hopfield của John Hopfield (1982), SOM (Self-Organization Map) của Kohonen, máy Boltzmann của Ackley, Hinton và Sejnowski. Tiếp theo là sự ra đời của mạng Back-Propagation năm 1986 do D. Rumelhart, G. Hilton và R. Williams giới thiệu, sau đó là mạng ART (Adaptive Resonance Networks) vào năm 1987 của Gail Carpenter và Stephen Grossberg tại đại học Boston. Trong những năm 1990 việc ứng dụng các mô hình mạng nơron vào phục vụ các lĩnh vực trong cuộc sống được phát triển mạnh mẽ.

## **NƠRON SINH HỌC VÀ NƠRON NHÂN TẠO**

Mạng nơron nhân tạo được thiết kế dựa trên mô hình mạng nơron thần kinh với các phần tử nơron nhân tạo của nó là sự mô phỏng nơron sinh học. Các mức tổ chức bộ não và cấu trúc mạng nơron sinh vật có thể được tham khảo trong tài liệu. Trong phần này chúng ta chỉ tập trung tìm hiểu cấu tạo của nơron sinh học và nơron nhân tạo để thấy được sự tương quan giữa chúng.

### **Cấu trúc nơron sinh học**

Nơron sinh vật có nhiều dạng khác nhau như dạng hình tháp, dạng tổ ong, dạng rễ cây. Tuy khác nhau về hình dạng, chúng có cấu trúc và nguyên lý hoạt động chung. Một tế bào nơron gồm bốn phần cơ bản:

*Các nhánh và rễ:* Các nhánh và rễ là các bộ phận nhận thông tin, các đầu nhạy hoặc các đầu ra của các nơron khác bám vào rễ hoặc nhánh của một nơron. Khi các đầu vào từ ngoài này có sự chênh lệch về nồng độ  $K^+$ ,  $Na^+$  hay  $Cl^-$  so với nồng độ bên trong của nó thì xảy ra hiện tượng thẩm thấu từ ngoài vào trong thông qua một cơ chế màng thẩm đặc biệt. Hiện tượng thẩm thấu như vậy tạo nên một cơ chế truyền đạt thông tin với hàng ngàn hàng vạn lối vào trên một nơron sinh vật, ứng với hàng nghìn hàng vạn liên kết khác nhau. Mức độ thẩm thấu được đặc trưng bởi cơ chế màng tương đương bằng một tỷ lệ. Tỷ lệ đó được gọi là tỷ trọng hay đơn giản gọi là trọng (Weight).

*Thân thần kinh (Soma):* Thân thần kinh chứa các nhân và cơ quan tổng hợp protein. Các ion vào được tổng hợp và biến đổi. Khi nồng độ các ion đạt đến một giá trị nhất định, xảy ra quá trình phát xung (hay kích thích). Xung đó được phát ở các đầu ra của nơron. Dây dẫn đầu ra xung được gọi là dây thần kinh.

*Dây thần kinh (Axon):* Dây thần kinh là đầu ra. Đó là phương tiện truyền dẫn tín hiệu. Dây thần kinh được cấu tạo gồm các đốt và có thể dài từ micro mét đến vài mét tùy từng kết cấu cụ thể. Đầu ra này có thể truyền tín hiệu đến các nơron khác.

Khớp thần kinh là bộ phận tiếp xúc của đầu ra nơron với rễ, nhánh của các nơron khác. Chúng có cấu trúc màng đặc biệt để tiếp nhận các tín hiệu khi có sự chênh lệch về nồng độ ion giữa bên trong và bên ngoài. Nếu độ lệch về nồng độ càng lớn thì việc truyền các ion càng nhiều và ngược lại. Mức độ thẩm thấu của các ion có thể coi là một đại lượng thay đổi tùy thuộc vào nồng độ như một giá trị đo thay đổi được gọi là trọng.

Trong não người có khoảng 15 tỷ nơron, mỗi nơron được nối với nhiều nơron khác bằng những khớp thần kinh (synapses), một nơron có thể có đến hơn nghìn synapses, và số synapses tổng cộng lại được ước lượng khoảng 1 triệu tỷ. Tín hiệu được truyền qua thân nơron tới các synapses, và tùy theo trạng thái của chúng mà một hay nhiều nơron khác sẽ được kích thích để tiếp tục truyền tín hiệu. Đó là mức thấp nhất, nhưng từ đó tổ chức lên các tầng trên như thế nào, có thể có được một bản đồ chi tiết của một bộ não tới từng nơron không?

Mạng nơron sinh vật tổ chức thành từng lớp (layer). Ta có:

+ Mạng một lớp: là tập hợp các phần tử nơron có đầu vào và đầu ra trên mỗi một phần tử. Nếu mạng nối đầu ra của các phần tử này với đầu vào của phần tử kia gọi là mạng tự liên kết (autoassociative).

+ Mạng hai lớp: gồm một lớp đầu vào và một lớp đầu ra riêng biệt.

+ Mạng nhiều lớp: gồm một lớp đầu vào và một lớp đầu ra riêng biệt. Các lớp nằm giữa lớp đầu vào và lớp đầu ra gọi là lớp ẩn (hidden layers).

+ Mạng truyền thẳng: là mạng hai hay nhiều lớp mà quá trình truyền tín hiệu từ đầu ra lớp này đến đầu vào lớp kia theo một hướng.

+ Mạng truyền ngược: là mạng mà trong đó một hoặc nhiều đầu ra của các phần tử lớp sau truyền ngược tới đầu vào của lớp trước.

+ Mạng tự tổ chức: là mạng có khả năng sử dụng những kinh nghiệm của quá khứ để thích ứng với những biến đổi của môi trường (không dự báo trước). Loại mạng này thuộc nhóm hệ học, thích nghi không cần có tín hiệu chỉ đạo từ bên ngoài.

Bảng sau đây đưa ra những con số cụ thể để có thể so sánh những ưu việt của bộ não con người so với một máy tính cá nhân mức trung bình hiện nay.

*Bảng 6.1: So sánh khả năng làm việc của bộ não và máy tính*

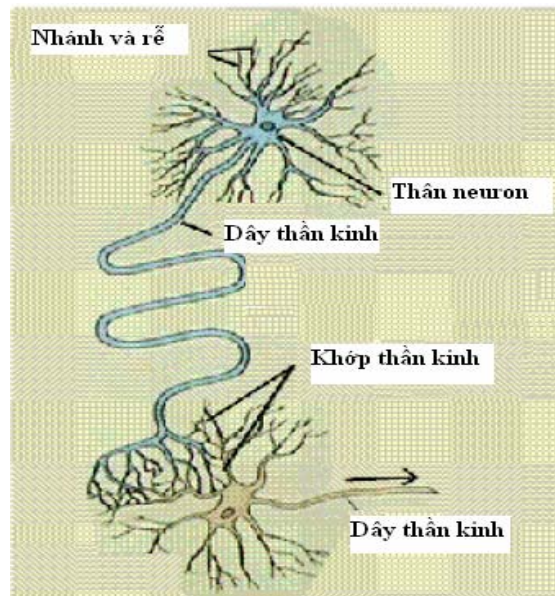
	Máy tính	Bộ não người
Đơn vị tính toán	Bộ xử lý trung tâm với 105 mạch logic	Mạng 1011 nơron
Bộ nhớ	109 bit RAM	1011 nơron
Bộ nhớ ngoài	Trên 1010 bit bộ nhớ ngoài	Với 1014 khớp nối thần kinh
Thời gian xử lý	10-8 giây	10-3 giây
Thông lượng	109 bit/giây	1014 bit/giây
Cập nhật thông tin	105 bit/giây	1014 nơron/giây

Những nghiên cứu trên đây là cơ sở mở đường cho con người nghiên cứu phương pháp mô phỏng lại mạng nơron sinh vật, đó chính là mạng nơron nhân tạo

Nơron sinh học có nhiều dạng khác nhau như dạng hình tháp, dạng tổ ong, dạng rễ cây tuy nhiên chúng có chung một cấu trúc và nguyên lý hoạt động. Hình sau đây mô phỏng một nơron sinh học:

Một tế bào nơron gồm các phần cơ bản:

*Nhánh và rễ:* Đây là bộ phận tiếp nhận thông tin của tế bào nơron. Các đầu nhạy hoặc các đầu ra của các nơron khác bám vào rễ hoặc nhánh của một nơron. Khi các đầu vào từ ngoài này có sự chênh lệch về nồng độ  $K^+$ ,  $Na^+$  hay  $Cl^-$  so với nồng độ bên trong của nó thì xảy ra hiện tượng thẩm thấu từ ngoài vào trong thông qua cơ chế màng thẩm đặc biệt. Hiện tượng thẩm thấu như vậy tạo nên một cơ chế truyền đạt thông tin với hàng ngàn hàng vạn lối vào trên một nơron sinh học ứng với hàng nghìn hàng vạn liên kết khác nhau. Mức độ thẩm thấu của các ion có thể coi là một đại lượng thay đổi tùy thuộc vào nồng độ gọi là tỷ trọng hay đơn giản là trọng (weight).



**Hình 6.1. Hình minh họa nơron sinh học.**

*Thân thần kinh:* Thân thần kinh chứa các nhân và cơ quan tổng hợp protein. Trong quá trình các ion vào được tổng hợp và biến đổi khi nồng độ các ion đạt đến một giá trị nhất định, xảy ra quá trình phát xung (hay kích thích). Xung đó được phát ở đầu ra của nơron. Dây dẫn đầu ra xung được gọi là dây thần kinh.

*Dây thần kinh:* Đây là phương tiện truyền dẫn tín hiệu của giữa các nơron. Dây thần kinh được cấu tạo gồm các đốt và có thể dài từ vài micro mét đến vài mét tùy từng kết cấu cụ thể.

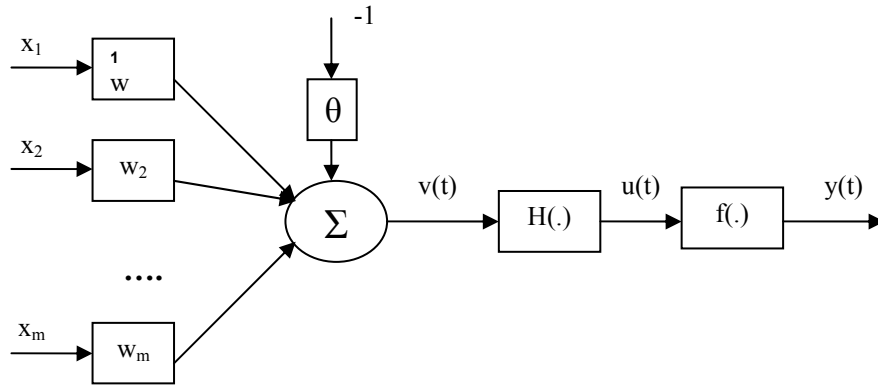
*Khớp thần kinh:* Là bộ phận tiếp xúc của đầu ra nơron với nhánh, rễ của các nơron khác. Chúng có cấu trúc màng đặc biệt để tiếp nhận các tín hiệu khi có sự chênh lệch về nồng độ ion giữa bên trong và bên ngoài. Nếu độ lệch càng lớn thì việc truyền các ion càng nhiều và ngược lại.

### **Cấu trúc nơron nhân tạo**

Trên cơ sở cấu trúc của nơron sinh học tổng quát người ta đề xuất mô hình nơron nhân tạo gồm 3 phần chính: Bộ tổng liên kết đầu vào, bộ động học tuyến tính và bộ phi tuyến, được minh họa cụ thể qua hình vẽ dưới đây: Cụ thể là:

**Bộ tổng liên kết đầu vào:** Là bộ tổng hợp các liên kết tại đầu vào của một phần tử neuron, mô tả như sau:

$$v(t) = \sum_{k=1}^m w_k x_k(t) - \theta \quad (1.1)$$



**Hình 1.2. Mô hình neuron nhân tạo.**

trong đó:

$v(t)$ : Tổng tất cả các đầu vào mô tả toàn bộ thế năng tác động ở thân neuron .

$x_k(t)$ : Các đầu vào ngoài; với k là chỉ số chạy,  $k=1,2,\dots,m$ .

$m$ : Số lượng đầu vào mô tả tín hiệu vào từ các đầu nhạy thần kinh hoặc các neuron khác.

$w_k$ : Trọng liên kết ngoài giữa các đầu vào k tới neuron hiện tại.

$y(t)$ : Tín hiệu đầu ra neuron .

$\theta$ : Ngưỡng (là hằng số), xác định ngưỡng kích thích hay ức chế.

**Bộ động học tuyến tính:** Đầu vào của phần động học tuyến tính là  $v(t)$ , đầu ra của nó là  $u(t)$  gọi là đầu ra tương tự. Hàm truyền tương ứng của phần động học tuyến tính có thể mô tả dưới dạng:

$$H(s) = \frac{U(s)}{V(s)}$$

hay

$$U(s) = H(s)V(s) \quad (1.2)$$

Các hàm  $H(s)$  thường dùng được liệt kê trong bảng dưới đây:

**Bảng 1.1. Các hàm truyền  $H(s)$  thường dùng**

H(s)	1	$\frac{1}{s}$	$\frac{1}{1-sT}$	$Exp(-sT)$
Quan hệ vào ra	$u(t) = v(t)$	$\frac{du(t)}{dt} = v(t)$	$T \frac{du(t)}{dt} + u(t) = v(t)$	$u(t) = v(t - T)$

**Bộ phi tuyến:** Phần này sử dụng hàm quan hệ phi tuyến  $f(.)$  cho đầu ra  $y$ , để chặn tín hiệu ở đầu ra. Các hàm đầu ra thông thường là các hàm có giới hạn ngưỡng phù hợp với đặc điểm đầu

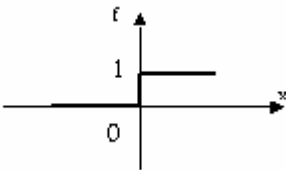
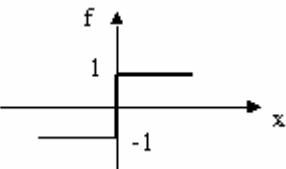
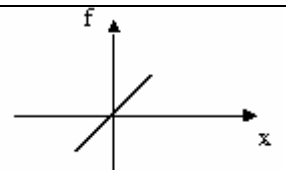
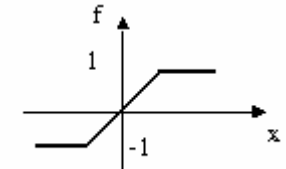
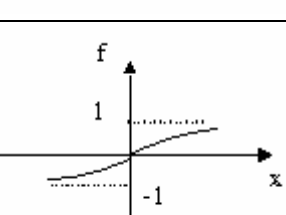


ra của nơron sinh học. Hàm dạng này thường được gọi là hàm kích hoạt (activation) thể hiện đặc điểm kích hoạt hay ức chế của một nơron.

$$y = f(u(t)) = f\left(\sum_{i=1}^m x_i(t)w_i - \theta\right) \quad (1.3)$$

Hàm phi tuyến ở đây có thể chia thành hai nhóm: Nhóm hàm bước nhảy và nhóm hàm liên tục. Đặc điểm chung của các hàm này thường tiến tới một giá trị cố định với các giá trị được gán  $[0,1]$  hoặc  $[-1,1]$ . Bảng dưới đây liệt kê một số hàm phi tuyến thường được dùng trong các mô hình nơron:

Bảng 1.2. Một số hàm phi tuyến thường dùng trong mô hình nơron

Tên hàm	Công thức	Đặc tính
Bước nhảy đơn vị	$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$	
Hàm dấu (sgn)	$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$	
Hàm tuyến tính	$f(x) = x$	
Hàm tuyến tính bão hòa đối xứng	$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } -1 \leq x \leq 1 \\ -1 & \text{if } x < -1 \end{cases}$	
Hàm Sigmoid lưỡng cực	$f(x) = \frac{2}{1 + e^{-2x}} - 1$	

Ngoài ra còn một số hàm khác cũng được sử dụng như dạng hàm Gauss, hàm logarit, hàm mũ, hàm arctg.

## MẠNG NƠN NHÂN TẠO VÀ LUẬT HỌC

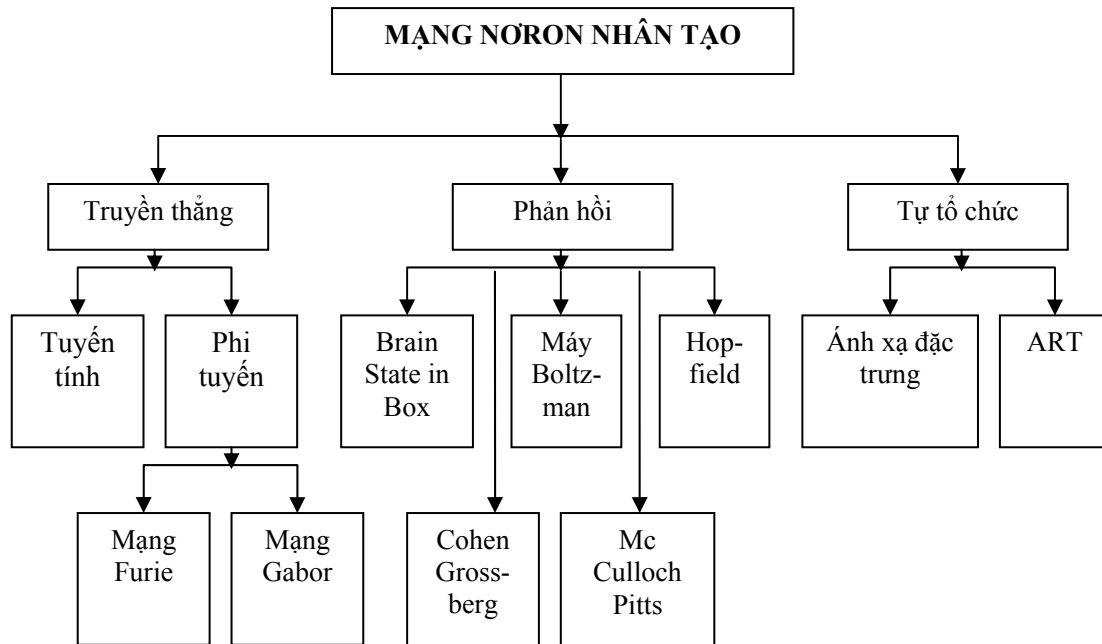
### Khái niệm mạng nơron nhân tạo

Các nơon sinh học liên kết với nhau thành lớp tạo thành mạng nơon sinh học. Nơon nhân tạo là sự mô phỏng nơon sinh học. Có nhiều cách kết hợp các nơon nhân tạo thành mạng, mỗi cách kết hợp tạo thành một lớp mạng nơon nhân tạo khác nhau.

**Phân loại mạng nơon nhân tạo**

Có nhiều cách để phân loại mạng nơon nhân tạo, tùy theo tiêu chí đề ra:

- Dựa vào số lượng lớp có trong mạng, ta phân thành:
  - + Mạng một lớp
  - + Mạng nhiều lớp



Hình 1.3. Một cách phân loại mạng

- Dựa vào đường truyền tín hiệu trong mạng, ta phân thành:
  - + Mạng truyền thẳng
  - + Mạng phản hồi
  - + Mạng tự tổ chức

Trong mỗi lớp mạng lại có nhiều mạng với các tên gọi và đặc trưng khác nhau, hình vẽ trên đây (Hình 1.3. Phân loại mạng) thể hiện một kiểu phân loại điển hình các mạng nơon.

**Khái quát luật học trong mạng nơon**

Khái niệm học trong mạng nơon được hiểu theo hai nghĩa: Học về cấu trúc và học về tham số.

**Học tham số (Parameter Learning)**

Mục tiêu của việc học tham số là thay đổi, cập nhật các trọng liên kết. Hầu hết các luật học tồn tại thuộc kiểu học tham số. Các kiểu học điển hình mà chúng ta sẽ nghiên cứu trong phần 1.4 cũng thuộc dạng học tham số. Thông thường, luật học tham số được chia thành ba dạng chính, đó là: Học giám sát, học không giám sát và học củng cố.

*Học có thầy (Học giám sát: Supervised Learning):* Với kiểu học này, tại mỗi thời điểm có đầu vào mạng nơron thì đầu ra mong muốn của hệ sẽ được cho sẵn. Có thể cụ thể hóa như sau:

Mạng được cung cấp một tập các mẫu  $(x^{(1)}, d^{(1)}), (x^{(2)}, d^{(2)}), \dots (x^{(n)}, d^{(n)})$  là các cặp đầu vào – đầu ra mong muốn. Khi một đầu vào mạng  $x^{(k)}$  được đưa vào, đầu ra mong muốn  $d^{(k)}$  cũng được đưa vào mạng. Sai khác giữa giá trị đầu ra thực sự  $y^{(k)}$  và đầu ra mong muốn  $d^{(k)}$  sẽ là cơ sở tạo tín hiệu lỗi để mạng sửa lỗi trọng sao cho đầu ra thực sự gần với đầu ra mong muốn hơn.

*Học không có thầy (Học không giám sát: Unsupervised Learning):* Với kiểu học này, không có bất kì một thông tin phản hồi từ môi trường nào. Mạng phải tự tìm ra các mẫu, đặc tính, tính quy tắc, sự tương quan trong dữ liệu đầu vào và tập hợp lại để tạo đầu ra. Khi tự tìm ra các đặc điểm này, mạng đã trải qua các thay đổi về tham số của nó. Quá trình này được gọi là tự tổ chức.

Các luật học có thể được khái quát thành dạng chung với lượng điều chỉnh trọng như sau:

$$\Delta w_{ij} = \alpha x_j(t) \quad (1.4)$$

Trong đó:  $i$  là nơron thứ  $i$ ;  $j$  là nơron thứ  $j$ ;  $\alpha$  - là hằng số học (dương) xác định tốc độ học;  $r$  là tín hiệu học. Tín hiệu học tổng quát là một hàm của  $w$ ,  $x$  và  $d$  tức là  $r=f(w,x,d)$ .

Đối với các trọng biến đổi liên tục ta có thể sử dụng dạng sau:

$$\frac{dw_i(t)}{dt} = \alpha r x_j(t) \quad (1.5)$$

*Học củng cố (reinforcement learning):* Đôi khi trong một số trường hợp, ta không biết hết được các giá trị đầu ra chính xác để có thể học giám sát mà chỉ biết được một số thông tin đánh giá. Khi đó luật học dựa trên những thông tin đánh giá này gọi là luật học củng cố và thông tin phản hồi được gọi là thông tin củng cố. Luật học này chỉ khác luật học giám sát ở chỗ, thông tin phản hồi từ môi trường chỉ mang tính chất đánh giá chứ không mang tính chất dạy, có nghĩa là chỉ đánh giá đầu ra thực sự là tốt hay xấu mà không đưa ra được đầu ra mong muốn là gì. Tín hiệu củng cố được mạng sử dụng với hi vọng là đầu ra thực sự có đánh giá phản hồi tốt hơn trong lần học tiếp theo.

### ***Học cấu trúc (Structure Learning)***

Mục tiêu của học cấu trúc là thay đổi số nơron, kiểu liên kết để làm cho cấu trúc mạng thay đổi. Đối với học tham số, chúng ta giả sử cấu trúc mạng đã có sau đó đưa ra các thuật học để thay đổi các trọng liên kết nhằm làm cho đầu ra thực sự gần với đầu ra mong muốn. Còn việc học mức cấu trúc có thể sử dụng các kỹ thuật liên quan đến thuật toán gen (Genetic Algorithm) và lập trình tiến hóa (Evolutionary Programming). Các cách tìm kiếm trong thuật toán gen và lập trình tiến hóa khá tiêu tốn thời gian ngay cả đối với mạng có kích thước trung bình. Do đó, còn có thể sử dụng các kỹ thuật sửa đổi hay xây dựng mạng dần dần từ một cấu trúc ban đầu. Kỹ thuật thường được sử dụng trong trường hợp này là kỹ thuật xén bớt mạng nơron, phát triển mạng nơron và kết hợp cả hai: Xén bớt và phát triển mạng nơron.

## **MỘT SỐ CẤU TRÚC, LUẬT HỌC MẠNG NƠN ĐIỆN HÌNH**

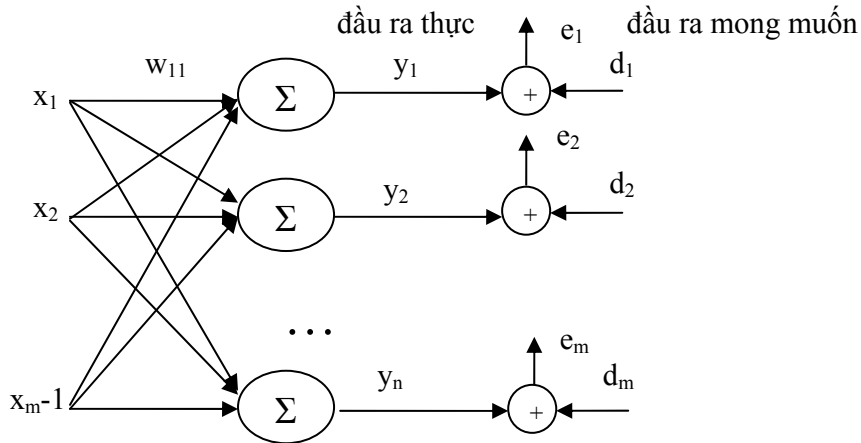
Trong phần này chúng ta sẽ giới thiệu cấu trúc và luật học của ba mạng nơron điển hình: Mạng Perceptron một lớp đơn, mạng Hopfield rời rạc và mạng nhiều lớp lan truyền ngược Back Propagation. Một điểm đặc biệt đó là trong số ba mạng được giới thiệu thì mạng Back

Propagation là mạng sẽ được sử dụng để giải quyết bài toán nhận dạng mã vạch EAN-8 mà đồ án đặt ra.

**Mạng Perceptron một lớp đơn**

**Cấu trúc**

Cấu trúc của mạng Perceptron một lớp đơn đặc trưng cho loại mạng truyền thẳng được mô tả trong hình vẽ dưới đây:



**Hình 1.4. Mạng Perceptron một lớp đơn**

Đầu vào của mạng có thể được mô tả là vector  $X=[x_1, x_2, \dots, x_m]^T$ , trong đó  $m$  là số lượng đầu vào. Giá trị ngưỡng của các nơron là các trọng liên kết với đầu vào cuối cùng  $x_m=-1$ . Với  $n$  nơron, vector đầu ra thực tế là  $Y=[y_1, y_2, \dots, y_n]$ . Mạng Perceptron sử dụng luật học có giám sát. Do đó tương ứng với mẫu đầu vào là vector  $X(k)=[x_1(k), x_2(k), \dots, x_m(k)]^T$ , mẫu đầu ra mong muốn là vector  $d(k)=[d_1(k), d_2(k), \dots, d_n(k)]^T$ . Với  $k=1, 2 \dots p$ ;  $p$  – là số cặp mẫu vào/ra;  $m$  – là số đầu vào;  $n$  – là số đầu ra;  $T$ : là kí hiệu chuyển vị. Chúng ta muốn đầu ra thực sự  $y(k)=d(k)$  sau quá trình học và có thể được mô tả như sau:

$$y_i^{(k)} = f(W_i^T x_i^{(k)}) = f\left(\sum_{j=1}^m W_{ij} x_j^{(k)}\right) = d_i^{(k)} \quad \text{với } i=1, 2 \dots n; k=1, 2 \dots p \quad (1.6)$$

Trong mạng Perceptron sử dụng hàm phi tuyến là hàm dấu nên phương trình trên có thể viết thành:

$$y_i^{(k)} = \text{Sign}(W_i^T x_i^{(k)}) = d_i^{(k)} \quad (1.7)$$

**Luật học**

Như đã nói ở trên mạng Perceptron sử dụng luật học Perceptron là luật học có giám sát, dạng chung của lượng điều chỉnh trọng là (công thức 1.4):

$$\Delta w_{ij} = \alpha r x_j(t)$$

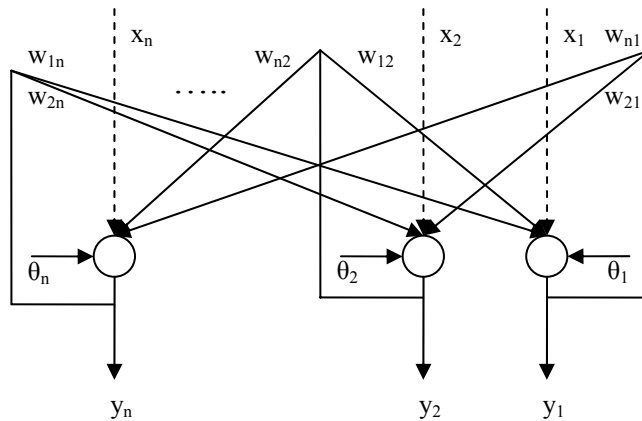
trong đó, tín hiệu học  $r = d_i - y_i$ . Do đầu ra mong muốn chỉ gồm hai giá trị 1 và -1 nên ta có:

$$\Delta W_{ij} = \alpha(d_i - y_i)x_j = \alpha(d_i - \text{Sign}(w_i^T x))x_j = \begin{cases} 2\alpha d_i x_i & \text{if } y_i \neq d_i \\ 0 & \text{else} \end{cases} \quad (1.8)$$

Như vậy, các trọng chỉ được cập nhật khi đầu ra thực sự  $y_i$  khác với  $d_i$ . Các trọng được khởi tạo với giá trị bất kì và luật học Perceptron sẽ hội tụ sau một số bước hữu hạn.

### Mạng Hopfield rời rạc (1984)

**Cấu trúc.** Mạng Hopfield mang tên một nhà vật lý người Mỹ là mạng phản hồi một lớp. Cấu trúc mạng được chỉ ra trong hình 1.5 dưới đây.



**Hình 1.5. Cấu trúc mạng Hopfield rời rạc**

Khi xử lý theo thời gian rời rạc, nó được gọi là một mạng Hopfield rời rạc hay cũng có thể được gọi là mạng hồi quy một lớp. Khi mạng hồi quy một lớp thực hiện một tiến trình cập nhật tuần tự, một mẫu đầu vào trước tiên được cung cấp cho mạng và theo đó đầu ra của mạng được khởi tạo. Sau đó, mẫu khởi tạo được xóa đi, đầu ra được cập nhật thông qua các kết nối phản hồi. Đầu vào được cập nhật lần thứ nhất sẽ có đầu ra được cập nhật lần thứ nhất, hoạt động này tái diễn, đầu vào được cập nhật lần thứ hai thông qua các liên kết phản hồi và cung cấp đầu ra được cập nhật lần thứ hai. Quá trình chuyển tiếp tiếp tục cho đến khi không có sự biến đổi, các đáp ứng cập nhật được cung cấp và mạng đã đạt được trạng thái cân bằng của nó.

Trong mạng Hopfield, mỗi node có một đầu vào bên ngoài  $x_i$  và một ngưỡng  $\theta_j$ , trong đó  $j=1,2,\dots,n$ . Một điều rất quan trọng là trong mạng Hopfield không có sự tự truyền ngược. Đầu ra node thứ  $j$  được kết nối tới đầu vào của tất cả các node còn lại sau khi nhân với trọng  $w_{ij}$ , với  $i,j=1,2,\dots,n; i \neq j$  và  $w_{ij}=w_{ji}$ .

### Luật học (Learning Rule)

Nguyên tắc tiến triển (nguyên tắc cập nhật) cho mỗi node trong mạng Hopfield là:

$$y_i^{(k+1)} = \text{sgn}\left(\sum_{j=1, j \neq i}^n w_{ij} y_j^{(k)} + x_i - \theta_i\right) \quad \text{với } i=1,2,\dots,n \quad (1.9)$$

Trong đó,  $\text{sgn}(\cdot)$  là hàm dấu,  $k$  là chỉ số của việc cập nhật đệ quy.

Nguyên tắc cập nhật ở trên được áp dụng theo kiểu không đồng bộ. Việc cập nhật tiếp theo trên một node được chọn ngẫu nhiên sử dụng các đầu ra vừa mới được cập nhật. Nói cách khác, do việc xử lý không đồng bộ của mạng, mỗi đầu ra của node được cập nhật một cách riêng rẽ, trong khi sử dụng các giá trị mới nhất mà đã được cập nhật. Phương pháp cập nhật không đồng bộ được đề nghị như một sự đệ quy ngẫu nhiên không đồng bộ của mạng Hopfield rời rạc.

### Tính ổn định

Tính ổn định của mạng được mô tả thông qua hàm năng lượng:

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1, j \neq i}^n w_{ij} y_i y_j - \sum_{i=1}^n x_i y_i + \sum_{i=1}^n \theta_i y_i \quad (1.10)$$

hoặc ngắn gọn:

$$\Delta E = -(net_i) \Delta y_i \text{ trong đó } \Delta y_i = y_i^{(k+1)} - y_i^{(k)} \quad (1.11)$$

Bản chất của công thức trên chỉ ra thực tế rằng  $y_i^{(k+1)} \neq y_i^{(k)}$  với  $j \neq i$  và  $w_{ij} = w_{ji}$  và  $w_{ii} = 0$  (thuộc tính trọng đối xứng).

Khi hàm năng lượng đạt được cực tiểu của nó (có thể là cực tiểu địa phương) thì mạng sẽ ổn định. Vì thế, bất đầu tại bất kì trạng thái khởi tạo nào, một mạng Hopfield luôn luôn hội tụ tại một trạng thái ổn định sau một số hữu hạn các bước cập nhật node, trong đó tất cả các trạng thái ổn định nằm tại một cực tiểu địa phương của hàm năng lượng  $E$ . Nguyên tắc này vận dụng lý thuyết cân bằng Lyapunov nổi tiếng mà thực tế được sử dụng để cung cấp tính cân bằng của một hệ thống động được định nghĩa từ nhiều phương trình khác nhau phối hợp chặt chẽ với nhau. Vì thế nó cung cấp một công cụ mạnh trong nghiên cứu lý thuyết của các mạng nơron.

### Mạng Lan truyền ngược

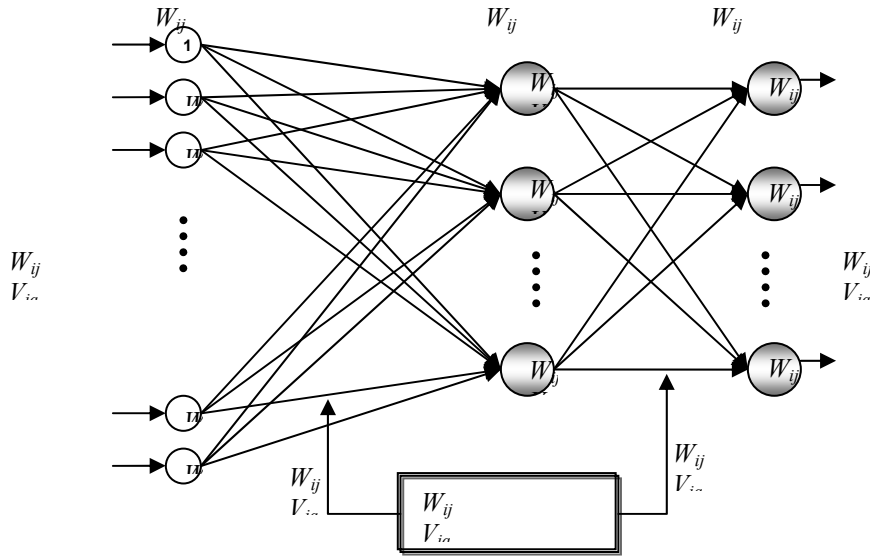
Thuật học lan truyền ngược là một trong những phát triển quan trọng trong mạng nơron. Thuật toán này được áp dụng cho các mạng nhiều lớp truyền thẳng (FeedForward) gồm các phần tử xử lý với hàm kích hoạt liên tục. Các mạng như vậy kết hợp với thuật toán học lan truyền ngược được gọi là mạng lan truyền.

Về mặt lý thuyết đã chứng minh được rằng: mạng ba lớp trở lên có thể nhận biết được mọi hàm bất kỳ. Chính vì vậy, luật học truyền ngược có ý nghĩa rất quan trọng trong việc cập nhật trọng của mạng nhiều lớp truyền thẳng. Nền tảng của thuật toán cập nhật trọng này cũng là phương pháp hạ Gradient. Thật vậy, cho cặp mẫu đầu vào - đầu ra  $(x^{(k)}, d^{(k)})$ , thuật toán lan truyền ngược thực hiện 2 pha. Đầu tiên, mẫu đầu vào  $x^{(k)}$  được truyền từ lớp vào tới lớp ra và kết quả của luồng dữ liệu tiến (forward) này là tạo đầu ra thực sự  $y^{(k)}$ . Sau đó, tín hiệu lỗi tạo từ sai khác giữa  $d^{(k)}$  và  $y^{(k)}$  sẽ được lan truyền ngược từ lớp ra quay trở lại các lớp trước đó để chúng cập nhật trọng. Để minh họa chi tiết thuật toán lan truyền ngược, xét một mạng 3 lớp: lớp vào có  $m$  nơron, lớp ẩn có  $l$  nơron và lớp ra có  $n$  nơron (hình 6.6)

- **Cấu trúc**

+ *Lớp ẩn*: với tập mẫu đầu vào  $x$ , nơron thứ  $q$  của lớp ẩn nhận tổng đầu vào  $l$

$$net_q = \sum_{j=1}^m v_{jq} x_j \quad j=1,2, \dots, m; \quad q=1,2, \dots, l \quad (6-14)$$



Hình 6.6. Mạng nhiều lớp lan truyền ngược

và tạo đầu ra của lớp ẩn:

$$z_q = f(net_q) = f\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (6-15)$$

trong đó  $f(.)$  là hàm tương tác đầu ra.

+ *Lớp ra*: giả thiết hàm tương tác đầu ra của lớp ra giống các lớp khác, tức là  $f(.)$ . Khi đó tổng đầu vào của neuron thứ  $i$  có thể xác định

$$net_i = \sum_{q=1}^l w_{iq} z_q = \sum_{q=1}^l w_{iq} f\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (6-16)$$

và tạo đầu ra:

$$y_i = f(net_i) = f\left(\sum_{q=1}^l w_{iq} z_q\right) = f\left(\sum_{q=1}^l w_{iq} f\left(\sum_{j=1}^m v_{qj} x_j\right)\right) \quad (6-17)$$

• **Luật học lan truyền ngược (Backpropagation Learning Rule)**

Cơ sở của luật học lan truyền ngược được xây dựng trên phương pháp hạ Gradient. Đầu tiên, xây dựng hàm chi phí (hay còn gọi là hàm sai số giữa đầu ra mong muốn  $d_i$  với đầu ra thực tế  $y_i$ )

$$E(w) = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^n [d_i - f(net_i)]^2 = \frac{1}{2} \sum_{i=1}^n \left[ d_i - f\left(\sum_{q=1}^l w_{iq} z_q\right) \right]^2 \quad (6-18)$$

Theo phương pháp hạ Gradient, trọng liên kết giữa lớp ẩn và lớp đầu ra được cập nhật bởi:

$$\begin{aligned}\Delta w_{iq} &= -\eta \frac{\partial E}{\partial w_{iq}} = -\eta \left( \frac{\partial E}{\partial y_i} \right) \left( \frac{\partial y_i}{\partial net_i} \right) \left( \frac{\partial net_i}{\partial w_{iq}} \right) \\ &= \eta (d_i - y_i) (f'(net_i)) z_q = \eta \delta_{oi} z_q\end{aligned}\quad (6-19)$$

với  $\delta_{oi}$  là tín hiệu lỗi tại nơron thứ i trong lớp ra ( $o = out$ )

$$\delta_{oi} = (d_i - y_i) f'(net_i) \quad (6-20)$$

Đối với trọng liên kết giữa nơron thứ j của lớp vào và nơron thứ q của lớp ẩn sẽ được cập nhật theo:

$$\begin{aligned}\Delta v_{qj} &= -\eta \frac{\partial E}{\partial v_{qj}} = -\eta \left( \frac{\partial E}{\partial net_q} \right) \left( \frac{\partial net_q}{\partial v_{qj}} \right) = -\eta \left( \frac{\partial E}{\partial z_q} \right) \left( \frac{\partial z_q}{\partial net_q} \right) \left( \frac{\partial net_q}{\partial v_{qj}} \right) \\ &= \eta \sum_{i=1}^n [(d_i - y_i) f'(net_i) w_{iq}] f'(net_q) x_j\end{aligned}\quad (6-21)$$

Từ phương trình (1-26) ta có:

$$\Delta v_{qj} = \eta \sum_{i=1}^n [\delta_{oi} w_{iq}] f'(net_q) x_j = \eta \delta_{hq} x_j \quad (6-22)$$

với  $\delta_{hq}$  là tín hiệu lỗi của nơron thứ q trong lớp ẩn ( $h = hidden$ )

$$\delta_{hq} = f'(net_q) \sum_{i=1}^n \delta_{oi} w_{iq} \quad (6-23)$$

Dựa vào công thức (1-20) ta tính ra được :

$$f'(net_q) = z_q (1 - z_q) \quad (6-24)$$

Thay (1-29) vào 1-28) ta được công thức cuối cùng cho tín hiệu lỗi của lớp ẩn:

$$\delta_{hq} = z_q (1 - z_q) \sum_{i=1}^n \delta_{oi} w_{iq} \quad (6-25)$$

Như vậy việc cập nhật trọng cho lớp ra và lớp ẩn có khác nhau một chút, công thức của lớp ra đơn giản hơn bởi nó không cần tính tổng  $\sum_{k=1}^M \delta_{pk} w_{kj}$  từ lớp kế sau nó mà chỉ đơn giản là hiệu giữa đầu ra mong muốn và đầu ra thực tại nơron hiện tại ( $d_{pj} - O_{pj}$ ). Điều này cũng dễ hiểu bởi lớp ra là lớp cuối cùng nên không có lớp kế sau.

Sau đây chúng ta nghiên cứu một vài ứng dụng đơn giản của mạng nơ ron trong nhận dạng ảnh.

#### 6.4.2 Nhận dạng ảnh các hình khối

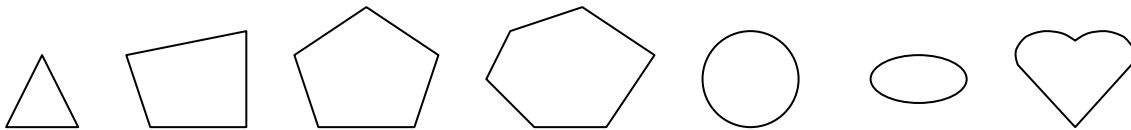
##### a) Đặt vấn đề

Bài toán đặt ra là : Sử dụng mạng Back-propagation để thử nghiệm nhận dạng các hình khối. Bài toán này có phạm vi đối tượng mẫu rất rộng, nhằm xác định cụ thể hơn cho miền hình ảnh nhận dạng của đồ án và việc phân tích bài toán được sâu hơn, giới hạn bài toán nhận dạng hình khối trong tài liệu này như sau:.



- 1) Nhận dạng các hình khối riêng lẻ không lồng nhau.
- 2) Hình ảnh không quá đặc biệt, ví dụ như hình tam giác không quá tù.
- 3) Ảnh có định dạng BMP.
- 4) Mức độ xoay của ảnh không quá cao

Các ảnh đầu vào có thể là nhiều dạng khác nhau: PCX, BMP, IMG, JPG, ..., nhưng trong hệ thống nhận dạng hình khối này chỉ thử nghiệm nhận dạng ảnh đầu vào dạng BMP. Tập dữ liệu ảnh không quá đặc biệt và khác giống nhau, các trường hợp như cùng cỡ ảnh nhưng ảnh thực trong đó quá bé so với toàn bộ ảnh cũng có thể dẫn đến kết quả không đúng. Các hình khối đầu vào của hệ nhận dạng:



**Hình 6.7. Các hình khối cơ bản.**

### **b) Giải quyết bài toán nhận dạng hình khối trên cơ sở mạng nơron nhân tạo**

Có hai cách cơ bản để nhận dạng ảnh đầu vào. Thứ nhất là xây dựng hàm biểu diễn cho các hình khối, rồi từ hàm biểu diễn đó hệ thống nhận ra đặc điểm hàm để phân loại hình. Ta đã biết, mỗi hình cơ bản đều có hàm đặc trưng riêng, ví dụ hình tròn là hàm:  $a.x^2+b.y^2=R^2$ , nếu coi 2 biên liền kề của ảnh là một hệ tọa độ thì ta hoàn toàn có thể tính toán được tọa độ các điểm trong hình, từ một loạt tọa độ của các điểm ảnh, với phương pháp nội suy gần đúng ta có thể đưa ra một hàm gần đúng của hình khối, từ đó mà nhận dạng được chúng.

Phương pháp thứ hai là tách lấy các đặc trưng về cấu trúc của ảnh như: số chu trình, số điểm nối (chạc 3, chạc tư), điểm kết thúc, điểm ngoặt,... với các đầu vào là các đặc trưng trên, sau một quá trình huấn luyện, mạng nơron sẽ nhận ra các đặc trưng riêng của mỗi đối tượng mà phân loại.

Trước hết là chọn phương pháp nhận dạng. trong bài toán này, phương pháp nhận dạng theo hàm là không hợp lý, bởi phép nội suy và hàm phù hợp với các đường cong không khép kín và đầu ra là các đa thức. Các hình khối mà chúng ta nhận dạng là các đường khép kín và không liên tục nên hàm đầu ra không thể là các đa thức mà là những hàm rất phức tạp. Do đó chúng ta sẽ chọn phương pháp trích chọn đặc trưng để giải quyết bài toán của chúng ta.

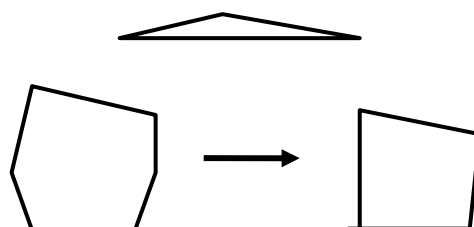
Tiếp đến là việc chọn tập các đặc trưng. Đặc trưng thông dụng và đơn giản được nhiều chương trình ứng dụng và đã thành công chính là các giá trị điểm ảnh. Ưu điểm của đặc trưng này là khâu trích chọn đặc trưng sẽ không phức tạp mà được giảm thiểu thành việc đọc các giá trị các điểm ảnh rồi mã hóa thành giá trị đầu vào cho mạng nơron. Ảnh sẽ được nén lại để giảm số nơron của mạng nhằm tăng tốc độ tính toán của mạng. Trong đồ bài toán này, kích cỡ ảnh sau khi nén được chọn là: 16x12 điểm ảnh, do đó, đầu vào của mạng là một lớp gồm 192 nơron. Lớp đầu ra có 7 nơron ứng với 7 hình phân loại. Lớp ẩn có 20 nơron.

### c) Kết quả

Với việc huấn luyện trên tập mẫu gồm 30 mẫu, mỗi mẫu 7 hình khối, số lần lặp huấn luyện là trên 2500 lần. Sau khi huấn luyện xong mạng được thử nghiệm 10 lần với kết quả như sau:

*Nhận dạng hình tam giác:* sau các thử nghiệm, kết quả cho thấy hình tam giác nhận dạng tốt ở những hình cân đối các góc không quá lệch nhau. Khi có một góc quá tù thì việc nhận dạng sẽ không hiệu quả. Số hình nhận dạng sai là 3.

*Nhận dạng hình tứ giác và hình lục giác:* hình tứ giác là một hình có nhiều dạng nên việc nhận dạng tốt thì phải trải qua quá trình huấn luyện mẫu phong phú. Số hình nhận dạng sai của cả hai loại tứ giác và lục giác là 7. Hình tứ giác và hình lục giác rất dễ nhầm với nhau khi hình lục giác có 2 góc đối diện với nhau lớn quá, dẫn đến việc nhầm thành đường thẳng:



**Hình 6.8. Một số hình khối quá đặc biệt.**

*Nhận dạng hình ngũ giác:* kết quả khá tốt, nhận dạng sai 2 hình.

*Nhận dạng hình tròn và hình elip:* khi nhận dạng hình in, kết quả cho khá tốt, nhưng nếu hình elip có hai trục không chênh lệch nhiều sẽ dễ bị nhầm với hình tròn. Nhận dạng sai 6 hình đối với cả hai loại.

*Nhận dạng hình trái tim:* hình trái tim cho kết quả nhận dạng tốt nhất trong các hình cơ bản bởi vì nó có những đặc trưng khác biệt so với các hình khác và sự biến dạng của nó là ít hơn. Số hình nhận dạng sai là 1.

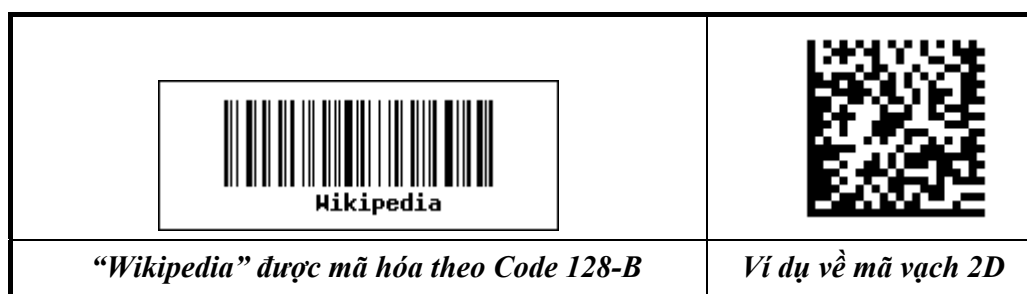
### 6.4.3 Nhận dạng ảnh mã vạch

#### a) Đặt vấn đề

**Khái niệm.** Mã vạch là sự thể hiện thông tin trong các dạng nhìn thấy trên các bề mặt mà máy móc có thể đọc được. Nói cách khác mã vạch là một công nghệ nhận dạng tự động cho phép dữ liệu được lưu trữ một cách chính xác và nhanh chóng. Trước kia, mã vạch lưu trữ dữ liệu theo bề rộng của các vạch được in song song cũng như của khoảng trống giữa chúng, nhưng ngày nay chúng còn được in theo mẫu của các điểm, theo các vòng tròn đồng tâm hay chúng ẩn trong các hình ảnh. Mã vạch có thể được đọc bởi các thiết bị quét quang học gọi là máy đọc mã vạch (máy quét mã vạch) hay được quét thành ảnh bằng các phần mềm chuyên biệt.

**Ứng dụng.** Mã vạch được sử dụng để đánh số các đồ vật với các thông tin liên quan mà máy tính có thể xử lý. Thay vì việc phải đánh một chuỗi dữ liệu vào phần nhập liệu của máy tính

thì người ta chỉ cần quét mã vạch của đồ vật đó bởi thiết bị đọc mã vạch. Chúng đặc biệt làm việc tốt trong điều kiện tự động hóa hoàn toàn, chẳng hạn như trong luân chuyển hành lý ở các sân bay.



**Hình 6.3 Ví dụ về các hình ảnh mã vạch**

Các dữ liệu chứa trong mã vạch thay đổi tùy theo ứng dụng. Trong trường hợp đơn giản nhất là một chuỗi số định danh được sử dụng như các chỉ mục (entry) trong cơ sở dữ liệu lưu toàn bộ các thông tin khác. Các mã EAN và UPC tìm thấy phổ biến trên hàng bán lẻ làm việc theo phương thức này.

Trong trường hợp khác, mã vạch chứa toàn bộ thông tin về sản phẩm mà không cần cơ sở dữ liệu ngoài. Điều này dẫn tới việc phát triển mã vạch tượng trưng có khả năng biểu diễn nhiều hơn, không phải chỉ là các số thập phân mà có thể bổ sung thêm các kí tự hoa và thường của bảng chữ cái cho đến toàn bộ bảng mã kí tự ASCII và nhiều hơn thế. Việc lưu trữ nhiều thông tin hơn đó dẫn đến việc phát triển các ma trận mã (một dạng mã vạch 2D), trong đó không chứa các vạch mà là một lưới các vùng.

- Phương pháp nhận dạng mã vạch:

Sự ra đời của mã vạch vào giữa thế kỉ XX mở ra một kỉ nguyên mới cho vấn đề lưu trữ thông tin sản phẩm trong các lĩnh vực sản xuất kinh doanh. Ngày nay ta có thể dễ dàng tìm thấy mã vạch trong các bao bì sản phẩm trong cuộc sống hàng ngày. Sự xuất hiện của mã vạch trở nên phổ biến và quen thuộc với người tiêu dùng chứng tỏ các công nghệ hỗ trợ quá trình đọc mã vạch cũng phải phát triển đủ mạnh để có thể đáp ứng được yêu cầu sử dụng. Mã vạch có thể được đọc bởi các máy quét quang học cũng có thể được quét từ hình ảnh bằng các phần mềm xử lí chuyên biệt. Hiện nay máy quét quang học (Barcode Scanner) được ứng dụng rộng rãi bởi sự phù hợp của nó với môi trường làm việc cũng như loại sản phẩm sử dụng mã vạch.

Tuy nhiên phương pháp sử dụng máy quét mã vạch cũng có những ưu nhược điểm riêng. Ưu điểm nổi bật của phương pháp này là tính tiện dụng, nhận dạng nhanh với độ chính xác cao. Nhược điểm của nó là đòi hỏi về trang thiết bị cùng các phụ kiện đi kèm thường không gọn nhẹ và không mang tính khả chuyển ở khoảng cách như các chỉ mục (entry) trong cơ sở dữ liệu lưu toàn bộ các thông tin khác. Các mã EAN và UPC tìm thấy phổ biến trên hàng bán lẻ làm việc theo phương thức này.

Trong trường hợp khác, mã vạch chứa toàn bộ thông tin về sản phẩm mà không cần cơ sở dữ liệu ngoài. Điều này dẫn tới việc phát triển mã vạch tượng trưng có khả năng biểu diễn nhiều hơn, không phải chỉ là các số thập phân mà có thể bổ sung thêm các kí tự hoa và thường của bảng chữ cái cho đến toàn bộ bảng mã kí tự ASCII và nhiều hơn thế. Việc lưu trữ nhiều thông tin hơn

đó dẫn đến việc phát triển các ma trận mã (một dạng mã vạch 2D), trong đó không chứa các vạch mà là một lưới các vùng.

- Phương pháp nhận dạng má vạch:

Sự ra đời của mã vạch vào giữa thế kỉ XX đã mở ra một kỉ nguyên mới cho vấn đề lưu trữ thông tin sản phẩm trong các lĩnh vực sản xuất kinh doanh. Ngày nay ta có thể dễ dàng tìm thấy mã vạch trên các bao bì sản phẩm trong cuộc sống hàng ngày. Sự xuất hiện của mã vạch trở nên phổ biến và quen thuộc với người tiêu dùng chứng tỏ các công nghệ hỗ trợ qui trình đọc mã vạch cũng phải phát triển đủ mạnh để có thể đáp ứng được yêu cầu sử dụng. Mã vạch có thể được đọc bởi các máy quét quang học cũng có thể được quét từ hình ảnh bằng các phần mềm xử lý chuyên biệt. Hiện nay máy quét quang học (Barcode Scanner) được ứng dụng rộng rãi bởi sự phức hợp của nó với mọi trường làm việc cũng như loại sản phẩm sử dụng má vạch.

Tuy nhiên phương pháp sử dụng máy quét mã vạch cũng có những ưu nhược điểm riêng. Ưu điểm nổi bật của phương pháp này là tính tiện dụng, nhận dạng nhanh với độ chính xác cao. Nhược điểm của nó là đòi hỏi về trang thiết bị cùng các phụ kiện đi kèm thường không gọn nhẹ và không mang tính khả chuyển ở khoảng cách lớn cũng như yêu cầu về chất lượng mã vạch phải tương đối tốt.

Một câu hỏi đặt ra là trong những trường hợp người dùng không có sẵn các thiết bị quét mã vạch trong tay, họ chỉ có thể chụp ảnh của mã vạch mà muốn biết tất cả các thông tin liên quan đến sản phẩm có mã vạch đó thì phải đáp ứng yêu cầu này như thế nào? Và liệu phương pháp nào có khả năng nhận ra mã vạch khi mã vạch đó được chụp thành ảnh? Câu trả lời được gợi mở khi chúng ta tìm hiểu về mạng nơron và thế mạnh nhận dạng ảnh của nó.

Bài toán “*Nhận dạng mã vạch EAN-8 trên cơ sở mạng nơron nhân tạo*” là một thử nghiệm để trả lời cho câu hỏi đã nêu trên.

#### **b) Giải quyết bài toán nhận dạng mã vạch EAN-8 trên cơ sở mạng nơron nhân tạo**

- **Cấu trúc mã vạch EAN-8:** EAN-8 là phiên bản EAN (hệ thống đánh số sản phẩm châu Âu – European Article Numbering) được tạo ra với mục đích sử dụng trên các loại bao bì hàng hóa nhỏ. Đúng như tên gọi của nó, EAN-8 mã hóa 8 số thập phân và đặc trưng của chuỗi 8 số này là được các tổ chức có thẩm quyền về mã vạch cung cấp trực tiếp do đó chuỗi 8 số này phải được lưu trữ trong mỗi cơ sở dữ liệu như mã các sản phẩm riêng biệt.

Trong chuỗi 8 số của EAN-8 có hai hoặc ba số là mã nước, năm hay bốn số còn lại là mã sản phẩm (chiều dài mã sản phẩm phụ thuộc vào chiều dài mã nước), số cuối cùng trong chuỗi là số kiểm tra.



**Hình 6.4 Mã vạch EAN-8 thể hiện chuỗi 12345670**

*Quy ước:* các bit có giá trị 1 được in bằng mực đen thành một đường thẳng đứng, các bit có giá trị 0 không được in (hoặc được in bằng mực trắng) thành một đường thẳng có cùng độ rộng với bit có giá trị 1.

Cấu trúc của mã vạch theo thứ tự như sau:

- + Vạch bảo vệ trái có giá trị nhị phân là 101.
- + Bốn số đầu trong chuỗi 8 số.
- + Vạch bảo vệ giữa có giá trị nhị phân là 01010.
- + Bốn số sau trong chuỗi 8 số.
- + Vạch bảo vệ phải có giá trị nhị phân là 101.

- Cấu trúc mạng Back-Propagation:

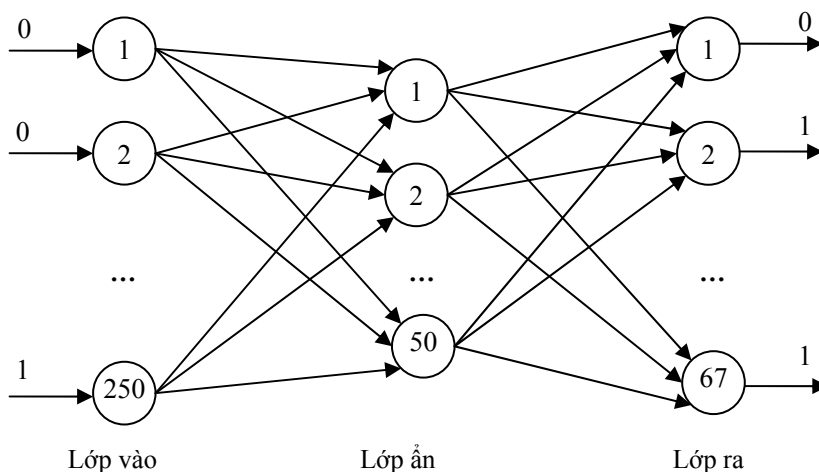
Mạng BackPropagation được sử dụng trong bài toán nhận dạng mã vạch EAN-8 đều có ba lớp: lớp vào, lớp ẩn và lớp ra.

Số nơron của lớp vào chính là số thành phần của vectơ đặc trưng cho ảnh mã vạch. Do đặc điểm của mã vạch là những vạch đen và trắng thẳng đứng có độ rộng xác định nên ta chỉ cần lấy số điểm ảnh của mã vạch quét theo chiều ngang của ma trận ảnh. Xác định theo thí nghiệm trên các mẫu sinh ra từ tiện ích BarEAN, số điểm ảnh luôn là 250 không phụ thuộc vào chiều cao và chiều rộng mã vạch. Do đó, số nơron lớp vào của hệ nhận dạng mã vạch EAN-8 là 250.

Số nơron lớp ra tùy thuộc vào cách mã hóa đặc trưng để xác định từng mẫu đầu ra mong muốn. Theo cấu trúc của EAN-8, mỗi mẫu EAN-8 sẽ được đặc trưng bởi 67 bit (vạch bảo vệ trái và phải gồm 3 bit, vạch bảo vệ giữa gồm 5 bit, chuỗi 8 số thập phân mỗi số được mã hóa bằng 7 bit). Do đó số nơron lớp ra của hệ nhận dạng mã vạch EAN-8 là 67. Với cách lựa chọn này thì một mẫu đầu ra sẽ được đặc trưng bởi tập 67 giá trị nhị phân. Sau đó ta dựng một hàm chuyển đổi để tìm ra chuỗi 8 số thập phân mà 67 giá trị nhị phân này biểu diễn, chuỗi 8 số thập phân đó chính là mã EAN-8 của sản phẩm, từ chuỗi này ta sẽ tìm ra các thông tin liên quan đến sản phẩm được lưu trong cơ sở dữ liệu (tên sản phẩm, giá sản phẩm, đơn vị tính).

Vì tính phức tạp cũng tăng lên khi số lớp ẩn tăng lên, do đó với bài toán đặt ra cho đồ án, hiện tại mạng BP được sử dụng xác định các tham số học gồm một lớp ẩn. Số nơron lớp ẩn được chọn theo thực nghiệm. Trong hệ nhận dạng mã vạch EAN-8 số nơron lớp ẩn được chọn là 50.

Hình dưới đây mô tả cấu trúc mạng BP sử dụng trong hệ nhận dạng:



**Hình 6.5 Mạng Back Propagation sử dụng trong hệ nhận dạng EAN-8**

Việc tính toán các tham số khác dùng trong quá trình huấn luyện mạng như trọng khởi tạo, hằng số học, sử dụng hàm kích hoạt nào được xác định theo thử nghiệm.

**c) Kết quả**

Hệ nhận dạng mã vạch EAN-8 sử dụng mạng nơron BackPropagation được xây dựng bằng ngôn ngữ Java do những tiện ích mà ngôn ngữ này đem lại (*phụ lục 2*: xây dựng hệ nhận dạng mã vạch EAN-8 bằng ngôn ngữ Java). Java dựa trên C++ nhưng nó lại đơn giản hơn C++ (vì nó lược bỏ các tính năng khó nhất của C++) mà vẫn thân thiện với người dùng. Bên cạnh đó Java cũng có tính độc lập với cấu trúc nghĩa là không phụ thuộc vào các loại máy, hệ điều hành; các trình ứng dụng viết bằng Java có thể dùng được trên hầu hết các máy tính. Ngoài ra Java cũng hỗ trợ đa luồng nghĩa là cho phép xây dựng trình ứng dụng có nhiều quá trình có thể xảy ra đồng thời.

Thử nghiệm dựng mạng nơron BP để nhận dạng mã vạch EAN-8 được thực thi trên máy tính Intel(R) Pentium(R) 4 CPU 1.80GHz, bộ nhớ 256 MB RAM.

Với những điều kiện trên thì các kết quả đó đạt được có thể tổng kết thông qua các bảng sau:

*Bảng 6.6 Bảng thống kê độ hội tụ với các trọng khởi tạo khác nhau*

Giá trị trọng khởi tạo	Số lượng vũng học (Lỗi cực đại: 0.01)	Số lượng vũng học (Lỗi cực đại: 0.001)
0.5	275	757
0.3	265	1728
0.2	210	151
0.19	201	478
0.18	313	482
0.17	299	864
0.16	321	890

*Bảng 6.7 Bảng thống kê độ hội tụ với các hằng số học khác nhau*

Giá trị hằng số học	Chất lượng huấn luyện mạng	
	Số vũng học (Lỗi cực đại: 0.01)	Phần trăm nhận dạng đúng
0.5	Khụng hội tụ	0%
0.3	Khụng hội tụ	0%
0.2	Khụng hội tụ	4%
0.1	1123	98%
0.05	185	100%
0.02	778	100%
0.01	634	100%

Bảng 6.8 Bảng thống kê độ hội tụ với số neuron lớp ẩn khác nhau

Số lượng neuron lớp ẩn	Chất lượng huấn luyện mạng	
	Số vòng học (Lỗi cực đại: 0.01)	Phần trăm nhận dạng đúng
30	948	88%
40	345	92%
45	233	100%
50	208	100%
55	216	100%
60	562	100%
70	1348	100%

Bảng 6.8 Thời gian huấn luyện với số lượng mẫu khác nhau

Số lượng mẫu huấn luyện	Số vòng học	Thời gian huấn luyện (giờ:phút:giây)	Phần trăm nhận dạng đúng
1	11	0:0:7	100%
10	138	0:0:20	100%
25	199	0:0:27	100%

**Kết luận:** Ta đã xem xét các ví dụ nhận dạng ảnh đơn giản. Nhiều ứng dụng mạng nơ ron khác như nhận dạng ký tự tiếng Việt, nhận dạng chữ viết tay, nhận dạng mặt người, nhận dạng tiền xu Việt nam hiện thời cũng đã cho nhiều các kết quả.

## CÂU HỎI VÀ BÀI TẬP

1. Thế nào là học (Learning) ? Trình bày học tham số và học cấu trúc.
2. Trình bày cấu trúc và các thành phần của mạng nơ ron sinh học.
3. Trình bày cấu trúc và các thành phần mạng nơ ron sinh học.
4. Trình bày cấu trúc và luật học mạng nơ ron Perceptron.
5. Trình bày cấu trúc và luật học mạng nơ ron ADALINE.
6. Trình bày cấu trúc và luật học mạng nơ ron Hopfield rời rạc.
7. Trình bày cấu trúc mạng nơ ron Lan truyền ngược.
8. Trình bày luật học mạng nơ ron Lan truyền ngược
9. Viết thuật toán luật học mạng nơ ron Lan truyền ngược.
10. Trình bày phương pháp và ý tưởng ứng dụng mạng nơ ron trong nhận dạng ảnh.
11. Trình bày phương pháp và ý tưởng ứng dụng mạng nơ ron trong nhận dạng ảnh các khối hình học.
12. Trình bày phương pháp và ý tưởng ứng dụng mạng nơ ron trong nhận dạng ảnh mã vạch.
13. Trình bày phương pháp thống kê trong nhận dạng ảnh.
14. Trình bày thuật toán nhận dạng ảnh theo khoảng cách.

## CHƯƠNG 7: NÉN DỮ LIỆU ẢNH

**Học xong phần này sinh viên có thể nắm được:**

1. Một số khái niệm (thuật ngữ) như: nén, tỉ lệ nén, các ý tưởng dẫn đến các phương pháp nén khác nhau và cách phân loại, đánh giá các phương pháp nén
2. Các phương pháp nén ảnh thế hệ thứ nhất
3. Mã hóa theo các biến đổi của thế hệ thứ nhất

### 7.1 TỔNG QUAN VỀ NÉN DỮ LIỆU ẢNH

#### 7.1.1 Một số khái niệm

*Nén dữ liệu (Data Compression)*

Nén dữ liệu nhằm làm giảm lượng thông tin “*du thừa*” trong dữ liệu gốc và do vậy, lượng thông tin thu được sau khi nén thường nhỏ hơn dữ liệu gốc rất nhiều. Với dữ liệu ảnh, kết quả thường là 10:1. Một số phương pháp còn cho kết quả cao hơn. Theo kết quả nghiên cứu được công bố gần đây tại Viện Kỹ thuật Georfie, kỹ thuật nén fractal cho tỉ số nén là 30 trên 1 [6].

Ngoài thuật ngữ “*nén dữ liệu*”, do bản chất của kỹ thuật này nó còn có một số tên gọi khác như : giảm độ dư thừa, mã hóa ảnh gốc.

Từ hơn hai thập kỷ nay, có rất nhiều kỹ thuật nén đã được công bố trên các tài liệu về nén và các phần mềm nén dữ liệu đã xuất hiện ngày càng nhiều trên thương trường. Tuy nhiên, chưa có phương pháp nén nào được coi là phương pháp vạn năng (Universal) vì nó phụ thuộc vào nhiều yếu tố và bản chất của dữ liệu gốc. Trong chương này, chúng ta không thể hy vọng xem xét tất cả các phương pháp nén. Hơn thế nữa, các kỹ thuật nén dữ liệu chung đã được trình bày trong nhiều tài liệu chuyên ngành. Ở đây, chúng ta chỉ đề cập các phương pháp nén có đặc thù riêng cho dữ liệu ảnh.

#### **Tỷ lệ nén (Compression Rate)**

Tỷ lệ nén là một trong các đặc trưng quan trọng nhất của mọi phương pháp nén. Tuy nhiên, về cách đánh giá và các kết quả công bố trong các tài liệu cũng cần quan tâm xem xét. Nhìn chung, người ta định nghĩa tỷ lệ cơ bản của phương pháp nén. Nhiều khi tỷ lệ nén cao cũng chưa thể nói phương pháp đó hiệu quả hơn các phương pháp khác, vì còn các chi phí như thời gian, không gian và thậm chí cả độ phức tạp tính toán nữa. Thí dụ như nén phục vụ trong truyền dữ liệu: vấn đề đặt ra là hiệu quả nén có tương hợp với đường truyền không.

Cũng cần phân biệt dữ liệu với nén băng truyền. Mục đích chính của nén là giảm lượng thông tin dư thừa và dẫn tới giảm kích thước dữ liệu. Tuy vậy, đôi khi quá trình nén cũng làm giảm băng truyền tín hiệu số hóa thấp hơn so với truyền tín hiệu tương tự.

#### 7.1.2 Các loại dư thừa dữ liệu

Như trên đã nói, nén nhằm mục đích giảm kích thước dữ liệu bằng cách loại bỏ dư thừa dữ liệu. Việc xác định bản chất các kiểu dư thừa dữ liệu rất có ích cho việc xây dựng các phương



pháp nén dữ liệu khác nhau. Nói một cách khác, các phương pháp nén dữ liệu khác nhau là do sử dụng các kiểu dư thừa khác nhau. Người ta coi có 4 kiểu dư thừa chính :

- Sự phân bố ký tự :

Trong một dãy ký tự, có một số ký tự có tần suất xuất hiện nhiều hơn so với các dãy khác. Do vậy, ta có thể mã hóa dữ liệu một cách cô đọng hơn. Các dãy ký tự có tần suất cao được thay bởi một từ mã nhị phân với số bit nhỏ; ngược lại các dãy có tần suất xuất hiện thấp sẽ được mã hóa bởi từ mã có nhiều bit hơn. Đây chính là bản chất của phương pháp mã hóa từ hóa Huffman.

- Sự lặp lại của các ký tự :

Kỹ thuật nén dùng trong trường hợp này là thay dãy lặp đó bởi dãy mới gồm hai thành phần: số lần lặp và kí hiệu dùng để mã. Phương pháp mã hóa kiểu này có tên là mã hóa loạt dài RLC (Run Length Coding).

- Những mẫu sử dụng tần suất:

Có thể có dãy ký hiệu nào đó xuất hiện với tần suất tương đối cao. Do vậy, có thể mã hóa bởi ít bit hơn. Đây là cơ sở của phương pháp mã hóa kiểu từ điển do Lempel-Ziv đưa ra và có cải tiến vào năm 1977, 1978 và do đó có tên gọi là phương pháp nén LZ77, LZ78. Năm 1984, Terry Welch đã cải tiến hiệu quả hơn và đặt tên là LZW (Lempel-Ziv-Welch).

- Độ dư thừa vị trí:

Do sự phụ thuộc lẫn nhau của dữ liệu, đôi khi biết được ký hiệu (giá trị) xuất hiện tại một vị trí, đồng thời có thể đoán trước sự xuất hiện của các giá trị ở các vị trí khác nhau một cách phù hợp. Chẳng hạn, ảnh biểu diễn trong một lưới hai chiều, một số điểm ở hàng dọc trong một khối dữ liệu lại xuất hiện trong cùng vị trí ở các hàng khác nhau. Do vậy, thay vì lưu trữ dữ liệu, ta chỉ cần lưu trữ vị trí hàng và cột. Phương pháp nén dựa trên sự dư thừa này gọi là phương pháp mã hóa dự đoán.

### 7.1.3 Phân loại phương pháp nén

Có nhiều cách phân loại các phương pháp nén khác nhau. Cách thứ nhất dựa vào nguyên lý nén. Cách này phân các phương pháp nén thành hai họ lớn:

- Nén chính xác hay nén không mất thông tin: họ này bao gồm các phương pháp nén mà sau khi giải nén ta thu được chính xác dữ liệu gốc.

- Nén có mất thông tin: họ này bao gồm các phương pháp mà sau khi giải nén ta không thu được dữ liệu như bản gốc. Phương pháp này lợi dụng tính chất của mắt người, chấp nhận một số vặn xoắn trong ảnh khi khôi phục lại. Tất nhiên, các phương pháp này chỉ có hiệu quả khi mà độ vặn xoắn chấp nhận được bằng mắt thường hay với dung sai nào đấy.

Cách phân loại thứ hai dựa vào cách thức thực hiện nén. Theo cách này, người ta cũng phân thành hai họ:

- Phương pháp không gian (Spatial Data Compression): Các phương pháp thuộc họ này thực hiện nén bằng các tác động trực tiếp lên việc lấy mẫu của ảnh trong miền không gian.

- Phương pháp sử dụng biến đổi (Transform Coding): gồm các phương pháp tác động lên sự biến đổi của ảnh gốc mà không tác động trực tiếp như họ trên.

Có một cách phân loại khác nữa, cách phân loại thứ ba, dựa vào triết lý của sự mã hóa. Cách này cũng phân các phương pháp nén thành hai họ:

- Các phương pháp nén thể hệ thứ nhất: Gồm các phương pháp mà mức độ tính toán là đơn giản, thí dụ việc lấy mẫu, gán từ mã, v.v.
- Các phương pháp nén thể hệ thứ hai: dựa vào độ bão hòa của tỷ lệ nén.  
Trong cách trình bày dưới đây ta sẽ theo cách phân loại này.

## 7.2 CÁC PHƯƠNG PHÁP NÉN THỂ HỆ THỨ NHẤT

Trong lớp các phương pháp này, ta lần lượt xem xét các phương pháp:

- Mã hóa loạt dài RLC (Run Length Coding)
- Mã hóa Huffman
- Mã hóa LZW (Lempel Ziv-Wench)
- Mã hóa khối (Block Coding)

### 7.2.1 Phương pháp mã hóa loạt dài

Phương pháp mã hóa loạt dài lúc đầu được phát triển dành cho ảnh số 2 mức: mức đen (1), và mức trắng (0) như các văn bản trên nền trắng, trang in, các bản vẽ kỹ thuật.

Nguyên tắc của phương pháp là phát hiện một loạt các bit lặp lại, thí dụ như một loạt các bit 0 nằm giữa hai bit 1, hay ngược lại, một loạt bit 1 nằm giữa hai bit 0. Phương pháp này chỉ có hiệu quả khi chiều dài dãy lặp lớn hơn một ngưỡng nào đó. Dãy các bit lặp gọi là loạt hay mạch (run). Tiếp theo, thay thế chuỗi đó bởi một chuỗi mới gồm 2 thông tin: chiều dài chuỗi và bit lặp (ký tự lặp). Như vậy, chuỗi thay thế sẽ có chiều dài ngắn hơn chuỗi cần thay.

Cần lưu ý rằng, đối với ảnh, chiều dài của chuỗi lặp có thể lớn hơn 255. Nếu ta dùng 1 byte để mã hóa thì sẽ không đủ. Giải pháp được dùng là tách các chuỗi đó thành hai chuỗi: một chuỗi có chiều dài 255, chuỗi kia là số bit còn lại.

Phương pháp RLC được sử dụng trong việc mã hóa lưu trữ các ảnh Bitmap theo dạng PCX, BMP.

Phương pháp RLC có thể chia thành 2 phương pháp nhỏ: phương pháp dùng chiều dài từ mã cố định và phương pháp thích nghi như kiểu mã Huffman. Giả sử các mạch gồm  $M$  bits. Để tiện trình bày, đặt  $M = 2^m - 1$ . Như vậy mạch cũ được thay bởi mạch mới gồm  $m$  bits.

Với cách thức này, mọi mạch đều được mã hóa bởi từ mã có cùng độ dài. Người ta cũng tính được, với  $M = 15, p = 0,9$ , ta sẽ có  $m = 4$  và tỷ số nén là 1,95.

Với chiều dài cố định, việc cài đặt thuật toán là đơn giản. Tuy nhiên, tỷ lệ nén sẽ không tốt bằng chiều dài biến đổi hay gọi là mã RLC thích nghi.

### 7.2.2 Phương pháp mã hóa Huffman

#### Nguyên tắc

Phương pháp mã hóa Huffman là phương pháp dựa vào mô hình thông kê. Dựa vào dữ liệu gốc, người ta tính tần suất xuất hiện của các ký tự. Việc tính tần suất được thực hiện bởi cách duyệt tuần tự tệp gốc từ đầu đến cuối. Việc xử lý ở đây tính theo bit. Trong phương pháp này người ta gán cho các ký tự có tần suất cao một từ mã ngắn, các ký tự có tần suất thấp từ mã dài. Nói một cách khác, các ký tự có tần suất càng cao được gán mã càng ngắn và ngược lại. Rõ ràng với cách thức này, ta đã làm giảm chiều dài trung bình của từ mã hóa bằng cách dùng chiều dài

biến đổi. Tuy nhiên, trong một số tình huống khi tần suất là rất thấp, ta có thể không được lợi một chút nào, thậm chí còn bị thiệt một ít bit.

### Thuật toán

Thuật toán bao gồm 2 bước chính:

- Giai đoạn thứ nhất: tính tần suất của các ký tự trong dữ liệu gốc: duyệt tệp gốc một cách tuần tự từ đầu đến cuối để xây dựng bảng mã. Tiếp sau đó là sắp xếp lại bảng mã theo thứ tự tần suất giảm dần.

- Giai đoạn thứ hai: mã hóa: duyệt bảng tần suất từ cuối lên đầu để thực hiện ghép 2 phần tử có tần suất xuất hiện thấp nhất thành một phần tử duy nhất. Phần tử này có tần suất bằng tổng 2 tần suất thành phần. Tiến hành cập nhật lại bảng và đương nhiên loại bỏ 2 phần tử đã xét. Quá trình được lặp lại cho đến khi bảng chỉ có một phần tử. Quá trình này gọi là quá trình tạo cây mã Huffman vì việc tập hợp được tiến hành nhờ một cây nhị phân 2 nhánh. Phần tử có tần suất thấp ở bên phải, phần tử kia ở bên trái. Với cách tạo cây này, tất cả các bit dữ liệu/ký tự là nút lá; các nút trong là các nút tổng hợp. Sau khi cây đã tạo xong, người ta tiến hành gán mã cho các nút lá. Việc mã hóa rất đơn giản: mỗi lần xuống bên phải ta thêm 1 bit “1” vào từ mã; mỗi lần xuống bên trái ta thêm một bit “0”. Tất nhiên có thể làm ngược lại, chỉ có giá trên mã thay đổi còn tổng chiều dài là không đổi. Cũng chính do lý do này mà cây có tên gọi là cây mã Huffman như trên đã gọi.

Quá trình giải nén tiến hành theo chiều ngược lại khá đơn giản. Người ta cũng phải dựa vào bảng mã tạo ra trong giai đoạn nén (bảng này được giữ lại trong cấu trúc của tệp nén cùng với dữ liệu nén). Thí dụ, với một tệp dữ liệu mà tần suất các ký tự cho bởi.

Ký tự	Tần suất
“1”	152
“2”	323
“3”	412
“4”	226
“5”	385
“6”	602
“7”	92
“8”	112
“9”	87
“0”	1532
“,”	536
“+”	220
“-”	315
“ ”	535

**Bảng tần suất**

Ký tự	Tần suất	Xác suất
“0”	1532	0.2770
“6”	602	0.1088
“,”	536	0.0969
“ ”	535	0.0967
“3”	112	0.0746
“5”	385	0.0696
“2”	323	0.0585
“-”	315	0.0569
“4”	226	0.0409
“+”	220	0.0396
“1”	152	0.0275
“8”	112	0.0203
“7”	92	0.0167
“9”	87	0.0158

**Bảng tần suất theo thứ tự giảm dần**

Lưu ý rằng, trong phương pháp Huffman, mã của ký tự là duy nhất và không mã nào là phần bắt đầu của mã khác. Vì vậy, khi đọc tệp nén từng bit từ đầu đến cuối ta có thể duyệt cây mã cho đến một lá, tức là ký tự đã được giải nén.

Bảng từ mã gán cho các ký tự bởi mã Huffman

“0”	10	“-”	0110
“6”	010	“4”	11110
“.”	001	“+”	11011
“ ”	000	“1”	111111
“3”	1110	“8”	111110
“5”	1100	“7”	110101
“2”	0111	“9”	110100

### 7.2.3 Phương pháp LZW

#### Mở đầu

Khái niệm nén từ điển được Jacob Lempel và Abraham Ziv đưa ra lần đầu tiên vào năm 1977, sau đó phát triển thành một họ giải thuật nén từ điển LZ. Năm 1984, Terry Welch đã cải tiến giải thuật LZ thành một giải thuật mới hiệu quả hơn và đặt tên là LZW. Phương pháp nén từ điển dựa trên việc xây dựng từ điển lưu các chuỗi ký tự có tần suất lặp lại cao và thay thế bằng từ mã tương ứng mỗi khi gặp lại chúng. Giải thuật LZW hay hơn các giải thuật trước nó ở kỹ thuật tổ chức từ điển cho phép nâng cao tỉ lệ nén.

Giải thuật nén LZW được sử dụng cho tất cả các loại file nhị phân. Nó thường được dùng để nén các loại văn bản, ảnh đen trắng, ảnh màu, ảnh đa mức xám... và là chuẩn nén cho các dạng ảnh GIF và TIFF. Mức độ hiệu quả của LZW không phụ thuộc vào số bit màu của ảnh.

#### Phương pháp

Giải thuật nén LZW xây dựng một từ điển lưu các mẫu có tần suất xuất hiện cao trong ảnh. Từ điển là tập hợp những cặp *từ vựng* và *nghĩa* của nó. Trong đó, *từ vựng* sẽ là các từ mã được sắp xếp theo thứ tự nhất định. *Nghĩa* là một chuỗi con trong dữ liệu ảnh. Từ điển được xây dựng đồng thời với quá trình đọc dữ liệu. Sự có mặt của một chuỗi con trong từ điển khẳng định rằng chuỗi đó đã từng xuất hiện trong phần dữ liệu đã đọc. Thuật toán liên tục “tra cứu” và cập nhật từ điển sau mỗi lần đọc một ký tự ở dữ liệu đầu vào.

Do kích thước bộ nhớ không phải vô hạn và để đảm bảo tốc độ tìm kiếm, từ điển chỉ giới hạn 4096 ở phần tử dùng để lưu lớn nhất là 4096 giá trị của các từ mã. Như vậy độ dài lớn nhất của từ mã là 12 bits ( $4096 = 2^{12}$ ). Cấu trúc từ điển như sau:

0	0	
0	1	
1	...	
...	...	
...	...	
255	255	
256	256	(Clear Code)
257	257	(End Of Information)
258	Chuỗi	
259	Chuỗi	
...	...	
...	...	
4095	...	
	Chuỗi	

+ 256 từ mã đầu tiên theo thứ tự từ 0...255 chứa các số nguyên từ 0...255. Đây là mã của 256 kí tự cơ bản trong bảng mã ASCII.

+ Từ mã thứ 256 chứa một mã đặc biệt là “mã xóa” (CC – Clear Code). Mục đích việc dùng mã xóa nhằm khắc phục tình trạng số mẫu lặp trong ảnh lớn hơn 4096. Khi đó một ảnh được quan niệm là nhiều mảnh ảnh, và từ điển là một bộ từ điển gồm nhiều từ điển con. Cứ hết một mảnh ảnh người ta lại gửi một mã xóa để báo hiệu kết thúc mảnh ảnh cũ, bắt đầu mảnh ảnh mới đồng thời khởi tạo lại từ điển cho mảnh ảnh mới. Mã xóa có giá trị là 256.

+ Từ mã thứ 257 chứa mã kết thúc thông tin (EOI – End Of Information). Mã này có giá trị là 257. Như chúng ta đã biết, một file ảnh GIF có thể chứa nhiều ảnh. Mỗi một ảnh sẽ được mã hóa riêng. Chương trình giải mã sẽ lặp đi lặp lại thao tác giải mã từng ảnh cho đến khi gặp mã kết thúc thông tin thì dừng lại.

+ Các từ mã còn lại (từ 258 đến 4095) chứa các mẫu thương lặp lại trong ảnh. 512 phân tử đầu tiên của từ điển biểu diễn bằng 9 bit. Các từ mã từ 512 đến 1023 biểu diễn bởi 10 bit, từ 1024 đến 2047 biểu diễn bởi 11 bit và từ 2048 đến 4095 biểu diễn bởi 12 bit.

**Ví dụ minh họa cơ chế nén của LZW**

Cho chuỗi đầu vào là “ABCBCABCABCD” (Mã ASCII của A là 65, B là 66, C là 67)

Từ điển ban đầu gồm 256 kí tự cơ bản.

Đầu vào	Đầu ra	Thực hiện
A(65)		A đã có trong từ điển => Đọc tiếp
B(66)	65	Thêm vào từ điển mã 258 đại diện cho chuỗi AB
C(67)	66	Thêm vào từ điển mã 259 đại diện cho chuỗi BC
B	67	Thêm vào từ điển mã 260 đại diện cho chuỗi CB
C		BC đã có trong từ điển => Đọc tiếp
A	259	Thêm vào từ điển mã 261 đại diện cho chuỗi BCA
B		AB đã có trong từ điển => Đọc tiếp
C	258	Thêm vào từ điển mã 262 đại diện cho chuỗi ABC
A	67	Thêm vào từ điển mã 263 đại diện cho chuỗi CA
B		AB đã có trong từ điển => Đọc tiếp
C		ABC đã có trong từ điển => Đọc tiếp
D	262	Thêm vào từ điển mã 263 đại diện cho chuỗi ABCD

Chuỗi đầu ra sẽ là :

$$65 - 66 - 667 - 259 - 258 - 67 - 262$$

Đầu vào có kích thước :  $12 \times 8 = 96 \text{ bits}$ . Đầu ra có kích thước là :  $4 \times 8 + 3 \times 9 = 59 \text{ bits}$

Tỉ lệ nén là  $96 : 59 \cong 1,63$

Thuật toán

- Giá trị cờ INPUT = TRUE khi vẫn còn dữ liệu đầu vào và ngược lại.

- Chức năng của các hàm :

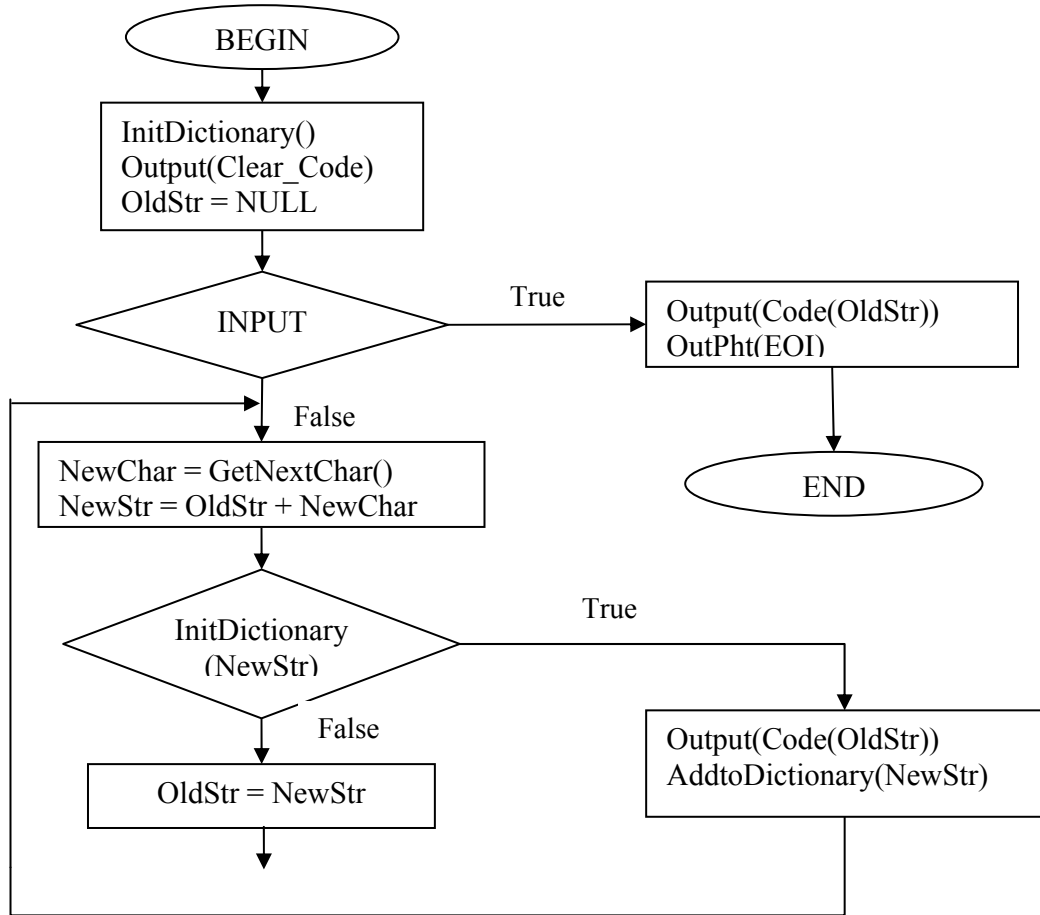
+ Hàm *InitDictionary()*: hàm này có chức năng khởi tạo từ điển. Đặt giá trị cho 256 phần tử đầu tiên. Gán mã xóa (Clear Code) cho phần tử thứ 256 và mã kết thúc thông tin (End Of Information) cho phần tử thứ 257. Xóa giá trị tất cả các phần tử còn lại.

+ Hàm *Output()*: gửi chuỗi bit ra file. Chuỗi này có độ dài là 9,10,11 hoặc 12 tùy thuộc vào vị trí trong từ điển của từ mã gửi ra. Các chuỗi bit này được nối tiếp vào với nhau.

+ Hàm *GetNextChar()*: trả về kí tự từ chuỗi kí tự đầu vào. Hàm này cập nhật giá trị của cờ INPUT xác định xem còn dữ liệu đầu vào nữa hay không.

+ Hàm *AddtoDictionary()*: sẽ được gọi khi có một mẫu mới xuất hiện. Hàm này sẽ cập nhật mẫu này vào phần tử tiếp theo trong từ điển. Nếu từ điển đã đầy nó sẽ gửi ra mã xóa (Clear Code) và gọi đến hàm *InitDictionary()* để khởi tạo lại từ điển.

+ Hàm *Code()*: trả về từ mã ứng với một chuỗi.



**Hình 7.3. Sơ đồ thuật toán nén LZW.**

Tư tưởng của đoạn mã trên có thể hiểu như sau: nếu còn dữ liệu đầu vào thì tiếp tục đọc. Một chuỗi mới sẽ được tạo ra từ chuỗi cũ (chuỗi này ban đầu trống, chuỗi này phải là chuỗi đã tồn tại trong từ điển) và kí tự vừa đọc vào. Sau đó kiểm tra xem chuỗi mới đã có trong từ điển chưa. Mục đích của công việc này là hi vọng kiểm tra xem chuỗi có số kí tự lớn nhất đã tồn tại trong từ điển. Nếu tồn tại ta lại tiếp tục đọc một kí tự tiếp theo và lặp lại công việc. Nếu chưa có trong từ điển, thì gửi chuỗi cũ ra ngoài và thêm chuỗi mới vào từ điển. Có thể xem lại phần ví dụ để hiểu rõ hơn.

### Giải nén dữ liệu nén bằng LZW

Giải thuật giải nén gần như ngược lại với giải thuật nén. Với giải thuật nén, một từ mã ứng với một chuỗi sẽ được ghi ra tệp khi chuỗi ghép bởi chuỗi trên với kí tự vừa đọc chưa có mặt trong từ điển. Người ta cũng cập nhật ngay vào từ điển từ mã ứng với chuỗi tạo bởi chuỗi cũ với kí tự vừa đọc. Kí tự này đồng thời là kí tự đầu tiên trong chuỗi tương ứng với từ mã sẽ được ghi ra tiếp theo. Đây là điểm mấu chốt cho phép xây dựng thuật toán giải nén.

Thuật toán được mô tả như sau :

```

While(GetnextCode != EOI) do
  Begin
    if FIRST_CODE /*Mã đầu tiên của mỗi mảnh ảnh*/
      Then Begin
        OutBuff(code);
      End
    End
  End

```

```

        OldStr := code;
    End;
    if code = CC /*Mã xóa*/
    Then Begin
        InitDictionary();
        FIST_CODE = TRUE;
    End;
    NewStr := DeCode(code);
    OutBuff(NewStr);
    OldString = OldStr + FirstChar(NewStr);
    AddtoDictionary(OldStr);
    OldString := NewStr;
End;

```

+ Giá trị cờ FIRST\_CODE = TRUE chỉ mã vừa đọc là mã đầu tiên của mỗi mảnh ảnh. Mã đầu tiên có cách xử lý hơi khác so với các mã tiếp theo.

+ Mã CC báo hiệu hết một mảnh ảnh. Mã EOF báo hiệu hết toàn bộ thông tin ảnh.

+ Chức năng của các hàm:

- *GetNextCode()*: hàm này đọc thông tin đầu vào (dữ liệu nén) trả về mã tương ứng. Chúng ta nhớ lại rằng, dữ liệu nén gồm chuỗi các từ mã nối tiếp nhau. Ban đầu là 9 bit, sau đó tăng lên 10 bit rồi 11, 12 bit. Nhiệm vụ của hàm này không phải đơn giản. Để biết được tại thời điểm hiện thời, từ mã dài bao nhiêu bit ra phải luôn theo dõi từ điển và cập nhật độ dài từ mã tại các phần tử thứ 512, 1024, 2048.
- *OutBuff()*: hàm này gửi chuỗi giá trị đã giải mã ra vùng nhớ đệm.
- *DeCode()*: hàm này tra cứu từ điển và trả về chuỗi kí tự tương ứng với từ mã.
- *FirstChar()*: lấy kí tự đầu tiên của một chuỗi. Kí tự vừa xác định nối tiếp vào chuỗi kí tự cũ (đã giải mã ở bước trước) ta được chuỗi kí tự có mặt trong từ điển khi nén. Chuỗi này sẽ được thêm vào từ điển giải nén.
- *OutPut()*: gửi chuỗi bit ra file. Chuỗi bit này có độ dài là 9, 10, 11 hoặc 12 tùy thuộc vào vị trí trong từ điển của từ mã gửi ra. Các chuỗi bit này được nối tiếp vào với nhau.

### Trường hợp ngoại lệ và cách xử lý

Đối với giải thuật LZW tồn tại một trường hợp được sinh ra nhưng chương trình giải nén có thể không giải mã được. Giả sử  $c$  là một ký tự,  $S$  là một chuỗi có độ dài lớn hơn 0. Nếu mã  $k$  của từ điển chứa giá trị là  $cS$ . Ngay sau đó  $k'$  được dùng thay thế cho  $cSc$ . Trong chương trình giải nén,  $k'$  sẽ xuất hiện trước khi nó được định nghĩa. Rất may là từ mã vừa đọc trong trường hợp này bao giờ cũng có nội dung trùng với tổ hợp của từ mã cũ với kí tự đầu tiên của nó. Điều này giúp cho quá trình cài đặt chương trình khắc phục được trường hợp ngoại lệ một cách dễ dàng.

## 7.2.4 Phương pháp mã hóa khối

### Nguyên tắc



Phương pháp này lúc đầu được phát triển cho ảnh số 2 mức xám, sau đó hoàn thiện thêm bởi các phương pháp thích nghi và mở rộng cho ảnh số đa cấp xám.

Cho một ảnh số  $I(x, y)$  kích thước  $M \times N$ . Người ta chia nhỏ ảnh số thành các khối hình chữ nhật kích thước  $k \times l$ , ( $k, l$ ) là rất nhỏ so với  $M, N$ . Như vậy ảnh gốc coi như gồm các khối con xếp cạnh nhau và có  $N \times M / (k \times l)$  khối con.

Ta có thể dùng phương pháp mã hóa Huffman cho từng khối của ảnh gốc, nghĩa là gán cho mỗi từ khối một từ mã nhị phân như ở phần trên. Một khó khăn gặp phải khi dùng mã hóa tối ưu Huffman đó là số lượng khối quá lớn. Giải pháp ở đây là dùng mã hóa gần tối ưu, đơn giản hơn để thực hiện mã hóa.

Giả thiết các khối là độc lập nhau và số cấu hình là  $2^{kl}$ . Gọi  $p(I, k, l)$  là xác suất xuất hiện cấu hình  $I$ , entropy tương ứng là:

$$H(h, l) = - \sum_{i=1}^{2^{kl}} p(i, k, l) \log_2 p(i, k, l)$$

Giá trị  $H(k, l)$  có thể được diễn giải là số bit / khối .

Các từ mã gán cho các khối  $k \times l$  được tạo bởi các điểm trắng ( cấu hình trội ) là “0”.

Các từ mã gán cho các khối  $k \times l$  khác gồm  $k \times l$  màu ( “1” cho đen , “0” cho trắng ) đi tiếp sau bit 1 tiền tố “1”.

Việc mã hóa theo số khối cũng được sử dụng nhiều trong các phương pháp khác như phương pháp dùng biến đổi sẽ trình bày trong mục 8.3 để giảm bớt không gian lưu trữ .

### Thuật toán

Giả sử  $p(0, k, x)$  là xác suất của khối chỉ tạo bởi các điểm trắng đã biết , tỷ số nên có thể tính được dễ dàng. Xác suất này có thể được thiết lập bởi mô hình lý thuyết cho một khối đặc biệt. Do vậy, ta chia khối làm hai loại: khối một chiều và khối hai chiều.

+Khối một chiều:

Xác suất  $P(0, k, l)$  tính được nhờ vào mô hình của quá trình Markov bậc một. Quá trình này được biểu diễn nhiều ma trận dịch chuyển trạng thái  $\Pi$  :

$$\Pi = \begin{pmatrix} p(t/t) & p(d/t) \\ p(t/d) & p(d/d) \end{pmatrix} \quad (7.1)$$

Với:

-  $p(t/t)$  là xác suất có điều kiện trắng sang trắng.

-  $p(d/d)$  là xác suất có điều kiện đen sang đen. Các xác suất khác có ý nghĩa tương tự.

Như vậy:

$$p(0, k, l) = p(t)p(t/t)^{k-1} \quad (7.2)$$

Điều này có thể giải thích như sau: xác suất xuất hiện một khối  $k \times 1$  chỉ gồm các điểm trắng bằng xác suất xuất hiện một điểm trắng tiếp theo  $k-1$  dịch chuyển trắng sang trắng.

Dựa vào các quan hệ trên, ta tính được tỉ số nén  $C_r$ .

$$C_r = \frac{l}{k[(1-p(t))p(t/t)^{k-1}] + l} \quad (7.3)$$

+Khối hai chiều:

Xác suất  $p(0, k, l)$  của các khối toàn trắng cũng tính được một cách tương tự như trên:

$$p(0, k, l) = p(t)p(t/t)^{k-1} [p(t/t).p(t/X = t, Y = t)^{l-1}]^{k-1} \quad (7.4)$$

Mối quan hệ này tương đương:

$$p(0, k, l) = p(t)p(t/t)^{k+l-2} p(t/X = t, Y = t)^{(l-1)(k-1)} \quad (7.5)$$

Và tỷ số nén sẽ cho bởi công thức:

$$C_r = \frac{l}{\left[ (1 - p(t)) p(t/t)^{k+l-2} \right] + \frac{l}{kl}} \quad (7.6)$$

Thực tế, khi cài đặt người ta hay chọn khối vuông và giá trị thích hợp của  $k$  từ 4 đến 5

### 7.2.5. Phương pháp thích nghi

Thuật ngữ “thích nghi” thường dùng để chỉ sự thích hợp của các từ mã theo một nghĩa nào đấy. Như trong phương pháp RLC ở trên, thay vì dùng chiều dài từ mã cố định  $m$  bit, người ta dùng chiều dài biến đổi và trên cơ sở đó có phương pháp RLC thích hợp.

Trong phương pháp mã hóa khối, người ta sử dụng chiều dài khối cố định gồm  $k \times l$  điểm ảnh. Tuy nhiên, với ảnh không thuần nhất, phương pháp mã hóa này bộc lộ nhiều nhược điểm. Vì rằng, với ảnh không đồng nhất, chính sự không thuần nhất của ảnh quyết định sự thích nghi với điều kiện cục bộ.

Một cải tiến cho vấn đề này là cố định một kích thước của khối, còn kích thước kia coi như là hàm của một tác động trung bình theo hàng (với  $l = l$ ) hay theo một nhóm hàng ( $l > l$ ). Tác động được quan tâm cũng giống như các phương pháp là sự dịch chuyển các điểm trắng sang đen trên hàng.

Một cách lý thuyết người ta cũng tính được giá trị tối ưu của  $k(k_{opt})$ :

$$k_{opt} = \begin{cases} \sqrt{\frac{N}{T}} & l = l \\ \sqrt{lI} & l > l \end{cases} \quad (7.7)$$

$N$  là số điểm ảnh trên hàng

Trên cơ sở này, người ta áp dụng mã hóa khối tự động thích nghi cho một số ứng dụng [6]:

- Mã đoạn hay khối  $k \times l$  tự động thích nghi với tác động cục bộ.
- Mã đoạn hay khối  $k \times l$  tự động thích nghi 1 chiều.
- Mã khối  $k \times l$  tự động thích nghi 2 chiều.

## 7.3. PHƯƠNG PHÁP MÃ HÓA DỰA VÀO BIẾN ĐỔI THỂ HỆ THỨ NHẤT:

Tuy bản chất của các phương pháp nén dựa vào biến đổi rất khác với các phương pháp đã trình bày ở trên, song theo phân loại nén, nó vẫn được xếp vào vào họ thứ nhất. Vì có các đặc thù riêng nên chúng ta xếp trong phần này.

### 7.3.1 Nguyên tắc chung

Các phương pháp mã hóa dựa vào biến đổi làm giảm lượng thông tin dư thừa không tác động lên miền không gian của ảnh số mà tác động lên miền biến đổi. Các biến đổi được dùng ở đây là các biến đổi tuyến tính như: biến đổi KL, biến đổi Fourier, biến đổi Hadamard, Sin, Cosin vv...

Vì ảnh số thường có kích thước rất lớn, nên trong cài đặt người ta thường chia ảnh thành các khối chữ nhật nhỏ. Thực tế, người ta dùng khối vuông kích thước cỡ 16x16. sau đó biến đổi từng khối một cách độc lập.

Chúng ta đã biết, dạng chung của biến đổi tuyến tính 2 chiều là:

$$X(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} a(m, n, k, l)x(k, l)$$

-  $x(k, l)$  là tín hiệu vào

-  $a(m, n, k, l)$  là các hệ số của biến đổi – là phần tử của ma trận biến đổi  $A$ .

Ma trận này gọi là *nhân* của biến đổi. Cách xác định các hệ số này là phụ thuộc vào từng loại biến đổi sử dụng. Đối với phần lớn các biến đổi 2 chiều, nhân có tính đối xứng và tách được :

$$A[m, n, k, l] = A'[m, k] A''[n, l]$$

Nếu biến đổi là KL thì các hệ số đó chính là các phần tử của vectơ riêng.

### 7.3.2 Thuật toán mã hóa dùng biến đổi 2 chiều

Các phương pháp mã hóa dùng biến đổi 2 chiều thường có 4 bước sau:

#### B1. Chia ảnh thành khối

Ảnh được chia thành các khối nhỏ kích thước  $k \times l$  và biến đổi các khối đó một cách độc lập để thu được các khối  $V_i, i=0, 1, \dots, B$  với  $B = M \times N / (k \times l)$ .

#### B2. Xác định phân phối bit cho từng khối

Thông thường các hệ số hiệp biến của các biến đổi là khác nhau. Mỗi hệ số yêu cầu lượng hóa với một số lượng bit khác nhau.

#### B3. Thiết kế bộ lượng hóa

Với phần lớn các biến đổi, các hệ số  $v(0, 0)$  là không âm. Các hệ số còn lại có trung bình 0. Để tính các hệ số, ta có thể dùng phân bố Gauss hay Laplace. Các hệ số được mã hóa bởi số bit khác nhau, thường từ 1 đến 8 bit. Do vậy cần thiết kế 8 bộ lượng hóa. Để dễ cài đặt, tín hiệu vào  $v_1(k, l)$  được chuẩn hóa để có dạng:

$$v_1(k, l) = v_1(k, l) / \sigma_{k,l} \quad (k, l) \neq (0, 0)$$

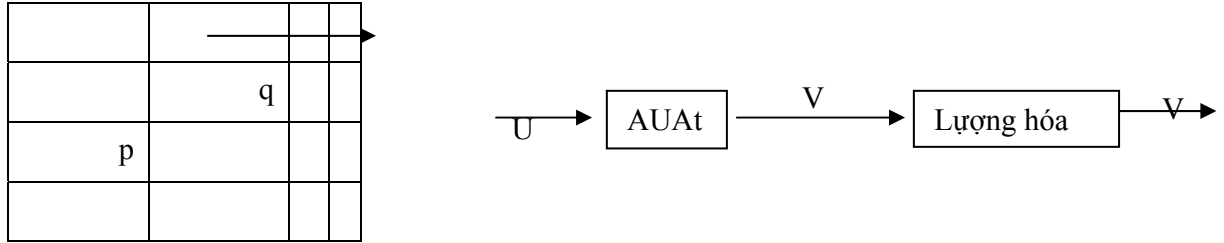
Trước khi thiết kế bộ lượng hóa, người ta tìm cách loại bỏ một số hệ số không cần thiết.

#### B4. Mã hóa

Tín hiệu đầu vào của bộ lượng hóa sẽ được mã hóa trên các từ bit để truyền đi hay lưu trữ lại. Quá trình mã hóa dựa vào biến đổi có thể được tóm tắt trên hình 7.4 dưới đây.

Nếu ta chọn phép biến đổi KL, cho phương pháp sẽ có một số nhược điểm: khối lượng tính toán sẽ rất lớn vì phải tính ma trận hiệp biến, tiếp sau là phải giải phương trình tìm trị riêng và vectơ riêng để xác định các hệ số. Vì lý do này, trên thực tế người ta thích dùng các biến đổi

khác như Hadamard, Haar, Sin và Cosin. Trong số biến đổi này, biến đổi Cosin thường hay được dùng nhiều hơn.



**Hình 7.4. Mã hóa và giải mã bởi mã hóa biến đổi.**

### 7.3.3 Mã hóa dùng biến đổi Cosin và chuẩn JPEG

#### a. Phép biến đổi Cosin một chiều

Phép biến đổi Cosin rời rạc (DCT) được Ahmed đưa ra vào năm 1974. Kể từ đó đến nay nó được ứng dụng rất rộng rãi trong nhiều phương pháp mã hóa ảnh khác nhau nhờ hiệu suất gần như tối ưu của nó đối với các ảnh có độ tương quan cao giữa các điểm ảnh lân cận. Biến đổi Cosin rời rạc được sử dụng trong chuẩn ảnh nén JPEG và định dạng phim MPEG.

*Phép biến đổi Cosin một chiều*

Phép biến đổi Cosin rời rạc một chiều được định nghĩa bởi:

$$X(k) = \frac{2\varepsilon_k}{N} \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\Pi(2n+1)k}{2N}\right) \quad (7.10)$$

$$\text{Trong đó: } \varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } k = 0 \\ 0 & \text{khi } k = [1, N-1] \end{cases}$$

Khi dãy đầu vào  $x(n)$  là thực thì dãy các hệ số  $X(k)$  cũng là số thực. Tính toán trên trường số thực giảm đi một nửa thời gian so với biến đổi Fourier. Để đạt được tốc độ biến đổi thỏa mãn yêu cầu của các ứng dụng thực tế, người ta đã cải tiến kỹ thuật tính toán và đưa ra nhiều thuật toán biến đổi nhanh Cosine. Một trong những thuật toán đó được giới thiệu dưới đây.

#### Phép biến đổi Cosin nhanh

Phép biến đổi Cosin nhanh viết tắt là FCT (Fast Cosine Transform), dựa vào ý tưởng đưa bài toán ban đầu về tổ hợp các bài toán biến đổi FCT trên các dãy con. Việc tiến hành biến đổi trên các dãy con sẽ đơn giản hơn rất nhiều so với dãy gốc. Vì thế, người ta tiếp tục phân nhỏ dãy tín hiệu cho đến khi chỉ còn một phần tử.

Giải thuật biến đổi Cosin nhanh không thực hiện trực tiếp trên dãy tín hiệu đầu vào  $x(n)$  mà thực hiện trên dãy  $x'(n)$  là một hoán vị của  $x(n)$ . Giả thiết số điểm cần tính FCT là lũy thừa của 2:  $N=2^M$

Dữ liệu đầu vào sẽ được sắp xếp lại như sau:

$$x'(i) = x(2i) \text{ với } i = 0, 1, \dots, \frac{N}{2} - 1$$

$$x'(N-i-1) = x(2i+1) \text{ với } i = 0, 1, \dots, \frac{N}{2} - 1$$

Như vậy, nửa đầu dãy  $x'(n)$  là các phần tử chỉ số chẵn của  $x(n)$  xếp theo chiều tăng dần của chỉ số. Nửa sau của  $x'(n)$  là các phần tử chỉ số lẻ của  $x(n)$  xếp theo chiều giảm dần của chỉ số.

Thay vào công thức (7.10) ta được:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) \cos \frac{\Pi(4n+1)k}{2N} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) \cos \frac{\Pi(4n+3)k}{2N}$$

Rút gọn biểu thức:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x'(2n) \cos \frac{\Pi(4n+1)k}{2N}$$

Chia  $X(k)$  ra làm hai dãy, một dãy bao hàm các chỉ số chẵn, còn dãy kia gồm các chỉ số lẻ.

### Phần chỉ số chẵn

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[ x'(n) + x'(n + \frac{N}{2}) \right] \cos \frac{\Pi(4n+1)2k}{2(\frac{N}{2})}$$

Có thể chuyển về dạng:

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} x'(n) \cos \frac{\Pi(4n+1)k}{2N} \quad (7.11)$$

### Các công thức:

Có thể nhận ra ngay các công trên là các phép biến đổi Cosin  $N/2$  điểm của  $g(n)$  và  $h(n)$ . Như vậy, bài toán biến đổi Cosin của dãy  $x'(n)$  đã được đưa về biến đổi Cosin của hai dãy là  $g(n)$  và  $h(n)$  có kích thước bằng một nửa  $x'(n)$ . Hai dãy  $g(n)$  và  $h(n)$  tính toán được một cách dễ dàng,  $g(n)$  là tổng của nửa đầu dãy  $x'(n)$  với nửa sau của nó,  $h(n)$  là hiệu của nửa đầu dãy  $x'(n)$  với nửa sau của nó, sau đó đem nhân với  $2C_N^n$ . Ta lặp lại quá trình chia đôi đối với các dãy con, dãy con của dãy con và cứ tiếp tục như thế. Giống như biến đổi Fourier, mỗi bước lặp cũng được coi là một tầng phân chia. Với  $N = 2^M$  thì số tầng phân chia là  $M$ .

Để dễ hình dung, đầu ra của mỗi tầng được kí hiệu là  $X_m(n)$  với  $m$  là tầng hiện thời. Ta xem  $x'(n)$  là biến đổi Cosin(0) tầng của  $x'(n)$ :

$$X_0(n) = x'(n) \quad (7.12)$$

$X_M(n)$  là biến đổi Cosin tầng  $M$  của  $x(n)$ , nó không phải là  $X(k)$ . Bởi vì cứ sau mỗi tầng, không chỉ thứ tự các phần tử trong  $X(k)$  bị xáo trộn mà các  $X(2k+1)$  còn được cộng với  $X(2k-1)$ . Đầu ra của một tầng là đầu vào của tầng tiếp theo.

$$X_1(n) = g(n) \quad \text{với } n = 0, 1, \dots, \frac{N}{2} - 1 \quad (7.14)$$

$$X_I(n + \frac{N}{2}) = h(n) \quad \text{với} \quad n = 0, 1, \dots, \frac{N}{2} - 1$$

Từ công thức tính  $g(n)$  và  $h(n)$  ta có:

$$\begin{aligned} X_I(i) &= X_0(i) + X_0(i + \frac{N}{2}) \\ X_I(i + \frac{N}{2}) &= \left[ X_0(i) - X_0(i + \frac{N}{2}) \right] 2C_N^i \quad \text{với} \quad n = 0, 1, \dots, N-1 \end{aligned} \quad (7.15)$$

Cứ sau mỗi tầng, số dãy con lại được nhân đôi. Xét phép biến đổi của tầng thứ  $m$ , chúng ta phải lặp lại công việc biến đổi cho  $2^{m-1}$  dãy con. Mỗi dãy con đóng vai trò như dãy  $x'(n)$  trong tầng thứ nhất. Số phần tử trong một dãy là:  $\frac{N}{2^{m-1}}$ . Công đoạn biến đổi trên một dãy con gọi là một khối biến đổi. Mỗi dãy con sẽ tiếp tục được phân làm hai dãy nhỏ hơn. Công thức tổng quát của mỗi khối là:

$$\begin{aligned} X_m(i) &= X_{m-1}(i) + X_{m-1}(i + \frac{N}{2^m}) \\ X_m(i + \frac{N}{2^m}) &= \left[ X_{m-1}(i) - X_{m-1}(i + \frac{N}{2^m}) \right] 2C_N^{i/m} \end{aligned} \quad (7.16)$$

với  $i = k \frac{N}{2^{m-1}}, \dots, k \frac{N}{2^{m-1}} + \frac{N}{2^m}$ , trong đó  $k=0, 1, \dots, 2^m - 1$

Phần xây dựng công thức tổng quát trong phép biến đổi nhanh Fourier được trình bày khá chi tiết ở trên, chúng ta có thể xem lại phần này để hiểu hơn về công thức tổng quát cho một khối biến đổi nhanh Cosin.

Thuật toán biến đổi nhanh Cosin có thể mô tả bằng các bước sau:

*Bước 1: Tính dãy hệ số  $C_j^i$ .*

Xác định số tầng  $M = \log_2 N$

Tầng hiện thời  $m=1$

*Bước 2: Nếu  $m \leq M$  thực hiện bước 5. Nếu không kết thúc.*

(Chưa hết các khối trong một tầng)

*Bước 3: Khối hiện thời  $k = 0$ .*

*Bước 4: Nếu  $k < 2^{m-1}$  Thực hiện bước 5. Nếu không thực hiện bước 6.*

(Chưa hết các khối trong một tầng)

*Bước 5: Tính toán  $X_m(i)$  trong khối theo công thức tổng quát (8.16), (8.17).*

Tăng  $k$  lên 1. Quay về bước 4.

*Bước 6: Tăng  $m$  lên 1. Quay về bước 2*

(Chuyển đến tầng tiếp theo)

Khác với biến đổi Fourier nhanh, trong biến đổi Cosin,  $x(n)$  không phải đầu vào trực tiếp và  $X(k)$  không phải là đầu ra trực tiếp. Ở đầu vào,  $x'(n)$  chỉ là cách sắp xếp lại  $x(n)$ . Chúng ta biết rằng tại mỗi tầng, đối với mỗi khối:

$$X(2i + 1) = X(2i + 1) + X(2i - 1)$$

Nên ở đầu ra, sau khi tính được  $X_M(n)$  chúng ta phải thực hiện việc trừ truy hồi từ tầng  $M$  về tầng 1 sau đó hoán vị lại theo thứ tự đảo bit mới thu được hệ số biến đổi  $X(k)$  cần tính.

Bài toán sắp xếp lại theo thứ tự đảo bit đã đề cập trong phần biến đổi Fourier. Bài toán trừ truy hồi cài đặt khá đơn giản.

Dãy hệ số  $C_j^i$  được tính trước một lần. trong các ứng dụng mà số điểm tính FCT không đổi hoặc chỉ nhận một số giá trị cụ thể, người ta thường tính trước  $C_j^i$  và ghi ra file. Khi thực hiện biến đổi thì đọc từ file để lấy thông tin này. Trong ứng dụng của chúng ta, ta tính trước  $C_j^i$  và lưu vào một mảng. Phép biến đổi sẽ truy cập bảng này để lấy hệ số cần thiết.

### Phép biến đổi Cosin ngược

Phép biến đổi Cosin ngược được định nghĩa bằng công thức:

$$x(n) = \sum_{k=0}^{N-1} X(k) \varepsilon_k \text{Cos} \frac{\Pi k(2n+1)}{2N} \quad (7.18)$$

$$\text{Với } \varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } k = 0 \\ 0 & \text{khi } k \neq 0 \end{cases}$$

Phép biến đổi Cosin ngược sẽ được thực hiện theo chiều ngược lại với quy trình đã tiến hành trong phép biến đổi nhanh. Tuy nhiên, công việc này không được thuận lợi như phép biến đổi FFT ngược. Từ  $X(k)$  chúng ta phải khôi phục lại  $X_M(k)$  bằng cách thực hiện các phép công truy hồi và phép hoán vị theo thứ tự đảo bit. Công thức tổng quát cho mỗi khối biến đổi ngược được xây dựng dựa trên công thức tổng quát trong biến đổi xuôi:

$$\text{Với } i = k \frac{N}{2^{m-1}}, \dots, k \frac{N}{2^{m-1}} + \frac{N}{2^m}, \text{ trong đó } k = 0, 1, \dots, 2^m - 1$$

$$X_{m-1}(i + \frac{N}{2^m}) = \frac{1}{2} X_m(i) - X_m(i + \frac{N}{2^m}) \frac{1}{2C_{N/2^{m-1}}^i} \quad (7.19)$$

$$X_{m-1}(i) = \frac{1}{2} X_m(i) + X_m(i + \frac{N}{2^m}) \frac{1}{2C_{N/2^{m-1}}^i} \quad (7.20)$$

Phép biến đổi ngược phải cài đặt riêng. Tuy vậy, tư tưởng chính của hai bài toán xuôi và ngược về cơ bản giống nhau. Đầu ra của phép biến đổi ngược sẽ là  $x'(n)$ . Muốn thu được  $x(n)$  ta phải đảo vị trí.

### b. Phép biến đổi Cosin rời rạc hai chiều

Phép biến đổi Cosin rời rạc hai chiều được định nghĩa bởi:

$$X(k_1, k_2) = \frac{4\varepsilon_{k_1}\varepsilon_{k_2}}{N_1N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) \text{Cos} \frac{\Pi(2n_1+1)k_1}{2N_1} \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \quad (7.21)$$

$$\text{Trong đó, } \varepsilon_{k_1} = 0 \text{ khi } k_1 = 0 \text{ và } \varepsilon_{k_1} = \frac{1}{\sqrt{2}} \text{ khi } k_1 = 1, 2, \dots, N_1 - 1$$

$$\varepsilon_{k_2} = 0 \text{ khi } k_2 = 0 \text{ và } \varepsilon_{k_2} = \frac{1}{\sqrt{2}} \text{ khi } k_2 = 1, 2, \dots, N_2 - 1$$

Phép biến đổi ngược được định nghĩa bởi công thức:

$$x(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(k_1, k_2) \varepsilon_{k_1} \varepsilon_{k_2} \text{Cos} \frac{\Pi(2n_1+1)k_1}{2N_1} \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \quad (7.22)$$

trong đó,  $\varepsilon_{k_1}, \varepsilon_{k_2}$  nhận các giá trị như trong công thức biến đổi xuôi.

Để nâng cao tốc độ biến đổi người ta đã phát triển các giải thuật biến đổi nhanh Cosin hai chiều. Cách làm phổ biến nhất là tận dụng phép biến đổi nhanh Cosin một chiều. Ta biến đổi công thức (7.21) về dạng:

$$X(k_1, k_2) = \frac{2\varepsilon_{k_1}}{N_1} \sum_{n_1=0}^{N_1-1} \left[ \frac{2\varepsilon_{k_2}}{N_1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \right] \text{Cos} \frac{\Pi(2n_1+1)k_1}{2N_1} \quad (7.23)$$

Đặt:

$$X'(n_1, k_2) = \frac{2\varepsilon_{k_2}}{N_1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \quad (7.24)$$

Công thức (7.23) trở thành:

$$X(k_1, k_2) = \frac{2\varepsilon_{k_1}}{N_1} \sum_{n_1=0}^{N_1-1} [X'(n_1, k_2)] \text{Cos} \frac{\Pi(2n_1+1)k_1}{2N_1} \quad (7.25)$$

Công thức (7.24) là phép biến đổi Cosin rời rạc một chiều của  $x(n_1, n_2)$ , trong đó  $n_2$  là biến số, còn  $n_1$  đóng vai trò là tham số thu được kết quả trung gian  $X'(n_1, k_2)$ . Công thức (7.25) là phép biến đổi Cosin rời rạc của  $X'(n_1, k_2)$  với  $n_1$  là biến số còn  $k_2$  là tham số. Đến đây tư tưởng của thuật toán đã rõ ràng. Khi biến đổi nhanh Cosin hai chiều của một ma trận ảnh, ta sẽ tiến hành biến đổi nhanh một chiều trên các điểm ảnh theo hàng, sau đó biến đổi nhanh một hàng theo cột của kết quả vừa thu được.

Biến đổi nhanh Cosin ngược hai chiều cũng được xây dựng dựa trên kết quả phép biến đổi nhanh Cosin ngược một chiều. Từ công thức (7.22) ta biểu diễn lại như sau:

$$x(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \left[ \sum_{k_2=0}^{N_2-1} X(k_1, k_2) \varepsilon_{k_2} \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \right] \varepsilon_{k_1} \text{Cos} \frac{\Pi(2n_1+1)k_1}{2N_1} \quad (7.26)$$

Đặt:

$$x'(n_1, n_2) = \sum_{k_2=0}^{N_2-1} [X(k_1, n_2)] \varepsilon_{k_2} \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \quad (7.27)$$

Khi đó công thức (7.26) sẽ trở thành:

$$x'(k_1, n_2) = \sum_{k_2=0}^{N_2-1} X(k_1, k_2) \varepsilon_{k_2} \text{Cos} \frac{\Pi(2n_2+1)k_2}{2N_2} \quad (7.28)$$

Công thức (7.27) là phép biến đổi Cosin ngược rời rạc một chiều của  $X(k_1, k_2)$ , trong đó  $k_2$  là biến số, còn  $k_1$  đóng vai trò là tham số thu được kết quả trung gian  $x'(k_1, n_2)$ . Công thức (7.28)



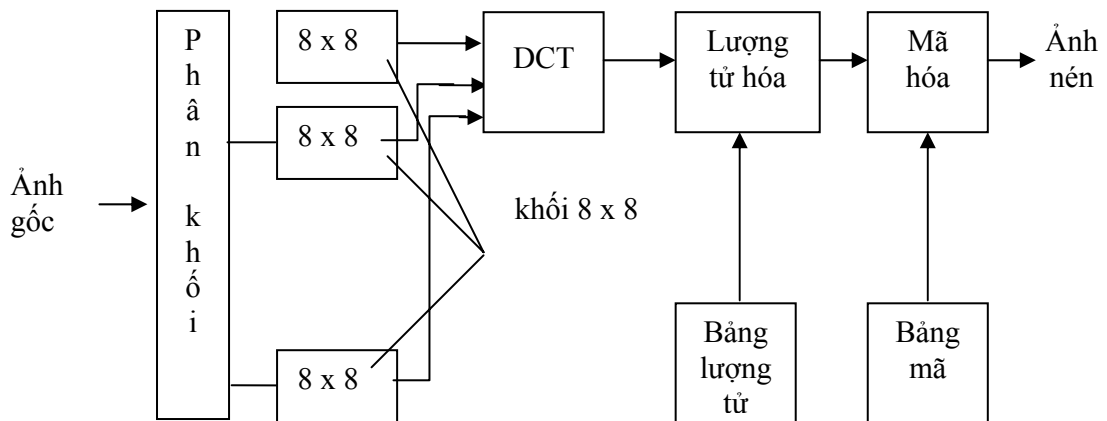
là phép biến đổi Cosin ngược rời rạc của  $x'(k_1, n_2)$  với  $k_1$  là biến số còn  $n_2$  là tham số. Như vậy, muốn khôi phục lại ảnh ban đầu từ ma trận hệ số biến đổi chúng ta sẽ biến đổi nhanh Cosin ngược rời rạc một chiều các hệ số theo hàng, sau đó đem biến đổi nhanh Cosin rời rạc một chiều theo cột các kết quả trung gian vừa tính được.

**c. Biến đổi Cosin và chuẩn nén JPEG**

JPEG là viết tắt của Joint Photographic Expert Group (nhóm các chuyên gia phát triển ảnh này). Chuẩn JPEG được công nhận là chuẩn ảnh quốc tế năm 1990 phục vụ các ứng dụng truyền ảnh cho các lĩnh vực như y học, khoa học, kỹ thuật, ảnh nghệ thuật...

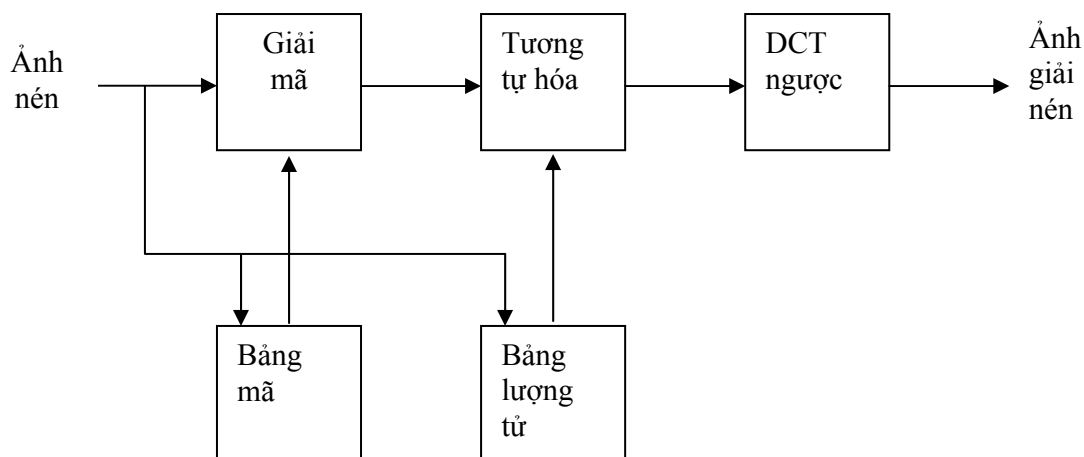
Chuẩn JPEG được sử dụng để mã hóa ảnh đa mức xám, ảnh màu. Nó không cho kết quả ổn định lắm với ảnh đen trắng. Chuẩn JPEG cung cấp giải thuật cho cả hai loại nén là nén không mất mát thông tin và nén mất mát thông tin. Trong phần dưới đây, chúng tôi trình bày chi tiết về một trong các dạng nén biến đổi chấp nhận mất mát thông tin dùng biến đổi Cosin của chuẩn JPEG: Biến đổi Cosin tuần tự (Sequential DTC - based). Biến đổi Cosin tuần tự là kỹ thuật đơn giản nhất nhưng được dùng phổ biến nhất và nó đáp ứng được hầu hết các đặc tính cần thiết cho phần lớn các ứng dụng.

Mã hóa JPEG bao gồm nhiều công , sSơ đồ thuật toán nén và giải nén được mô tả dưới đây.



**Hình 7.5 Sơ đồ thuật toán nén JPEG**

Quá trình giải nén sẽ được làm ngược lại, người ta giải mã từng phần ảnh nén tương ứng với phương pháp nén đã sử dụng trong phần nén nhờ các thông tin liên quan ghi trong phần header của file nén. Kết quả thu được là hệ số đã lượng tử. Các hệ số này được khôi phục về giá trị trước khi lượng tử hóa bằng bộ tương tự hóa. Tiếp đó đem biến đổi Cosin ngược ta được ảnh ban đầu với độ trung thực nhất định.



**Hình 7.5** Sơ đồ thuật toán giải nén JPEG

Bảng mã và bảng lượng tử trong sơ đồ giải nén được dựng lên nhờ những thông tin ghi trong phần cấu trúc đầu tệp ( Header) của tệp ảnh nén. Quá trình nén chịu trách nhiệm tạo ra và ghi lại những thông tin này. Phần tiếp theo sẽ phân tích tác dụng của từng khối trong sơ đồ.

**+ Phần khối**

Chuẩn nén JPEG phân ảnh ra các khối 8x8. Công đoạn biến đổi nhanh Cosin hai chiều cho các khối 8x8 tỏ ra hiệu quả hơn. Biến đổi Cosin cho các khối có cùng kích thước có thể giảm được một phần các tính toán chung như việc tính hệ số  $C_j^i$  cho 3 tầng ( $8 = 2^3$ ), số các hệ số là:  $4 + 2 + 1 = 7$

Nếu với một ảnh 1024 x 1024, phép biến đổi nhanh Cosin một chiều theo hàng ngang hoặc hàng dọc ta phải qua 10 tầng ( $1024 = 2^{10}$ ). Số các hệ số  $C_j^i$  là:  $512 + 256 + 128 + 64 + 8 + 4 + 2 + 1 = 1021$ . Thời gian tính toán các hệ số  $C_j^i$  với toàn bộ ảnh 1024 x 1024 lớn gấp 150 lần so với thời gian tính toán các hệ số này cho các khối.

Biến đổi Cosin đối với các khối có kích thước nhỏ sẽ làm tăng độ chính xác khi tính toán với số dấu phẩy tĩnh, giảm thiểu sai số do làm tròn sinh ra.

Do điểm ảnh hàng xóm có độ tương quan cao hơn, do đó phép biến đổi Cosin cho từng khối nhỏ sẽ tập trung năng lượng hơn và một số ít các hệ số biến đổi. Việc loại bớt một số hệ số năng lượng thấp trong các khối chỉ tạo ra mất mát thông tin cục bộ giúp nâng cao chất lượng ảnh.

Ảnh sẽ được chia làm B khối:

$$B = \left(\frac{M'}{k}\right) \times \left(\frac{N'}{l}\right) = M_B \times N_B$$

Các khối được xác định bởi bộ số  $(m,n)$  với  $m = [0 \dots M_B-1]$  và  $n = [0 \dots N_B-1]$ , ở đây  $m$  chỉ thứ tự của khối theo chiều rộng,  $n$  chỉ thứ tự của khối theo chiều dài. Phân tích khối thực chất là xác định tương quan giữa tọa độ riêng trong khối với tọa độ thực của điểm ảnh trong ảnh ban đầu. Nếu ảnh ban đầu kí hiệu Image[i,j] thì ma trận biểu diễn khối  $(m,n)$  là  $x[u, v]$  được tính:

$$x[u,v] = Image[mk + u, nl + v]$$

**+ Biến đổi**

Biến đổi là một trong những công đoạn lớn trong các phương pháp nén sử dụng phép biến đổi. Nhiệm vụ của công đoạn biến đổi là tập trung năng lượng vào một số ít các hệ số biến đổi.

Công thức biến đổi cho mỗi khối là:

$$X(k_1, k_2) = \frac{\varepsilon_{k_1} \varepsilon_{k_2}}{4} \sum_{n_1=0}^7 \sum_{n_2=0}^7 x(n_1, n_2) \text{Cos} \frac{(2n_1 + 1)k_1 \Pi}{16} \text{Cos} \frac{(2n_2 + 1)k_2 \Pi}{16} \quad (7.29)$$

Trong đó:

$$\varepsilon_{k_1} = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } k_1 = 0 \\ 0 & \text{khi } 0 < k_1 < 8 \end{cases}$$

$$\varepsilon_{k_2} = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } k_2 = 0 \\ 0 & \text{khi } 0 < k_2 < 8 \end{cases}$$

Thuật toán biến đổi nhanh Cosin hai chiều cho mỗi khối trong trường hợp này sẽ bao gồm 16 phép biến đổi nhanh Cosin một chiều. Đầu tiên, người ta biến đổi nhanh Cosin một chiều cho các dãy điểm ảnh trên mỗi hàng. Lần lượt thực hiện cho 8 hàng. Sau đó đem biến đổi nhanh Cosin một chiều theo từng cột của ma trận vừa thu được sau 8 phép biến đổi trên. Cũng lần lượt thực hiện cho 8 cột. Ma trận cuối cùng sẽ là ma trận hệ số biến đổi của khối tương ứng.

Trong sơ đồ giải nén ta phải dùng phép biến đổi Cosin ngược. Công thức biến đổi ngược cho khối 8x8:

$$x(n_1, n_2) = \frac{\varepsilon_1 \varepsilon_2}{4} \sum_{k_1=0}^7 \sum_{k_2=0}^7 X(k_1, k_2) \text{Cos} \frac{(2n_1 + 1)k_1 \Pi}{16} \text{Cos} \frac{(2n_2 + 1)k_2 \Pi}{16} \quad (8.30)$$

trong đó:

$$\varepsilon_{k_1} = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } k_1 = 0 \\ 0 & \text{khi } 0 < k_1 < 8 \end{cases}$$

$$\varepsilon_{k_2} = \begin{cases} \frac{1}{\sqrt{2}} & \text{khi } k_2 = 0 \\ 0 & \text{khi } 0 < k_2 < 8 \end{cases}$$

**+ LƯỢNG TỬ HÓA**

Khối lượng tử hóa trong sơ đồ nén đóng vai trò quan trọng và quyết định tỉ lệ nén của chuẩn nén j. Đầu vào của khối lượng tử hóa là các ma trận hệ số biến đổi Cosin của các khối điểm ảnh.

Để giảm số bộ lượng tử, người ta tìm cách quy các hệ số ở các khối về cùng một khoảng phân bố. Chuẩn nén  $j$  chỉ sử dụng một bộ lượng tử hóa. Giả sử rằng các hệ số đều có hàm tính xác suất xuất hiện như nhau. Chúng ta sẽ căn chỉnh lại hệ số  $y_j$  bằng phép gán:

$$y_j = \frac{y_j - \mu_j}{\sigma_j}$$

Với  $\mu_j$  là trung bình cộng của hệ số thứ  $j$

$\sigma_j$  là độ lệch cơ bản của hệ số thứ  $j$ .

Như vậy chúng ta sẽ đồng nhất được mức quyết định và mức tạo lại cho tất cả các hệ số. Do đó, các hệ số được biểu diễn cùng bằng một số lượng bit. Có nhiều cách tiếp cận để tính được các mức quyết định và mức tạo lại. Lloyd – Max đưa ra giải thuật sau:

**Bước 1:** Chọn giá trị khởi tạo:

$$d_0 = y_L$$

$$d_N = y_H$$

$$r_0 = d_0$$

$N$  là mức lượng tử

**Bước 2:** Cho  $i$  biến thiên từ  $1$  đến  $N-1$  thực hiện các công việc sau:

a. Tính  $d_i$  theo công thức:

$$r_{i-1} = \frac{\int_{d_{i-1}}^{d_i} y \cdot p(y) dy}{\int_{d_{i-1}}^{d_i} p(y) dy}$$

b. Tính  $r_i$  theo công thức:

$$r_i = 2d_i - r_{i-1}$$

**Bước 3:** Tính

$$r^t = \frac{\int_{d_{N-1}}^{d_N} y \cdot p(y) dy}{\int_{d_{N-1}}^{d_N} p(y) dy}$$

**Bước 4:** Nếu  $r_{N-1} \neq r^t$  điều chỉnh lại  $r_0$  và lặp lại từ bước 2 đến bước 4.

Trong quá trình cài đặt tạo ra một bộ lượng tử hóa, Lloyd và Max đã có nhiều cải tiến để tính toán dễ dàng hơn. Xác định  $d_i$  bằng công thức trong bước 2a được tiến hành theo phương pháp Newton-Raphson. Sau đây là các bước mô tả toàn bộ công việc của khối lượng tử hóa tác động lên các hệ số biến đổi Cosin:

**Bước 1:** Tính trung bình cộng  $\mu$  và độ lệch cơ bản  $\sigma$  cho từng hệ số ở mỗi vị trí trong khối

$$\sigma_j = \frac{\sum y_i}{n}$$

$$\sigma_i = \frac{n \sum y_j^2 - (\sum y_j)^2}{n(n-1)}$$

Với  $y_j$  là hệ số thứ  $j$ ,  $n$  là số khối.

**Bước 2:** Lựa chọn tỉ lệ số hệ số giữ lại trong một khối.

**Bước 3:** Giữ lại các hệ số có độ lệch cơ bản lớn hơn

**Bước 4:** Lập ma trận  $T$  sao cho:  $T_{ij} = 1$  nếu hệ số  $(i,j)$  được giữ lại.

**Bước 5:** Căn chỉnh lại giá trị của các hệ số xoay chiều được giữ lại ở các khối:

$$C_{ij} = \frac{C_{ij} - \mu_{ij}}{\sigma_{ij}}$$

**Bước 6:** Tính phân bố của các giá trị xoay chiều đã căn chỉnh.

**Bước 7:** Tính độ lệch cơ bản  $\sigma_j$  của các phân bố vừa tính.

**Bước 8:** Lượng tử hóa các hệ số xoay chiều bằng cách sử dụng bộ lượng tử Lloyd-Max sau khi đã điều chỉnh mức quyết định và mức tạo lại của nó theo cách sau:

$$d_i \leftarrow d_i \times \sigma_s$$

$$r_i \leftarrow r_i \times \sigma_s$$

$$dN = -d0$$

Thành phần một chiều sẽ không lượng tử hóa. Đến đây, ta chuyển sang bước nén.

#### + Nén dữ liệu

Đầu vào của khối nén gồm hai thành phần: thành phần các hệ số một chiều và thành phần các hệ số xoay chiều.

**Thành phần các hệ số một chiều**  $C_i(0, 0)$  với  $i = 0, 1, \dots, 63$  chứa phần lớn năng lượng tín hiệu hình ảnh. Người ta không nén trực tiếp các giá trị  $C_i(0, 0)$  mà xác định độ lệch của  $C_i(0, 0)$ :

$$d_i = C_{i+1}(0, 0) - C_i(0, 0)$$

$d_i$  có giá trị nhỏ hơn nhiều so với  $C_i$  nên trong biểu diễn dấu phẩy động theo chuẩn IEE754 thường chưa nhiều chuỗi bit 0 nên có thể cho hiệu suất nén cao hơn. Giá trị  $C_0(0, 0)$  và các độ lệch  $d_i$ , được ghi ra một tệp tạm. Tệp này được nén bằng phương pháp nén Huffman.

**Thành phần các hệ số xoay chiều**  $C_l(m, n)$  với  $1 \leq m \leq 7$ ,  $1 \leq n \leq 7$  chứa các thông tin chi tiết của ảnh. Để nâng cao hiệu quả nén cho mỗi bộ hệ số trong một khối, người ta xếp chúng lại theo thứ tự ZigZag.

Tác dụng của sắp xếp lại theo thứ tự ZigZag là tạo ra nhiều loại hệ số giống nhau. Chúng ta biết rằng năng lượng của khối hệ số giảm dần từ góc trên bên trái xuống góc dưới bên phải nên việc sắp xếp lại các hệ số theo thứ tự ZigZag sẽ tạo điều kiện cho các hệ số xấp xỉ nhau (*cùng mức lượng tử*) nằm trên một dòng.

Mỗi khối ZigZag này được mã hóa theo phương pháp RLE. Cuối mỗi khối đầu ra của RLE, ta đặt dấu kết thúc khối EOB (End Of Block). Sau đó, các khối được dồn lại và mã hóa một lần bằng phương pháp mã Huffman. Nhờ có dấu kết thúc khối nên có thể phân biệt được hai khối cạnh nhau khi giải mã Huffman. Hai bảng mã Huffman cho hai thành phần hệ số tất nhiên sẽ khác nhau.

Để có thể giải nén được, chúng ta phải ghi lại thông tin như: kích thước ảnh, kích thước khối, ma trận Y, độ lệch tiêu chuẩn, các mức tạo lại, hai bảng mã Huffman, kích thước khối nén một chiều, kích thước khối nén xoay chiều... và ghi nối tiếp vào hai file nén của thành phần hệ số.

Cài đặt giải thuật cho nén thực sự phức tạp. Chúng ta phải nắm được các kiến thức về nén RLE, Huffman, biến đổi Cosin, xây dựng bộ lượng tử hóa Lloyd-Max...Nén và giải nén  $j$  hơi chậm nhưng bù lại, thời gian truyền trên mạng nhanh hơn do kích thước tệp nén nhỏ.

Với những ưu điểm của mình được ISO chấp nhận là chuẩn ảnh quốc tế và được biết đến dưới mã số ISO 10918-1.

## 7.4 PHƯƠNG PHÁP MÃ HÓA THỂ HỆ THỨ HAI

Phương pháp mã hóa dựa vào biến đổi thể hệ thứ hai, như đã nói trong phần giới thiệu chung, có thể phân thành hai lớp nhỏ:

Lớp phương pháp sử dụng các phép toán cục bộ để tổ hợp đầu ra theo cách thức hợp lý và lớp phương pháp sử dụng biểu diễn ảnh.

Dưới đây, trong lớp phương pháp thứ nhất chúng ta sẽ xem xét một phương pháp có tên gọi là “Kim tự tháp Laplace”; còn trong lớp phương pháp thứ hai sẽ đề cập 2 phương pháp là vùng gia tăng và phương pháp tách-hợp.

### 7.4.1 Phương pháp Kim tự tháp Laplace (Pyramide Laplace)

Phương pháp này là tổ hợp của hai phương pháp: Mã hóa thích nghi và biến đổi. Tỷ số nén là khá cao, thường là 10/1. Về nguyên tắc, phương pháp này dựa vào mô hình phân cấp quan sát của con người.

Bắt đầu từ ảnh gốc  $x(m, n)$  qua bộ lọc dải thấp ta thu được tín hiệu  $x_1(m, n)$ . Bộ lọc này được thiết kế để tính trung bình cục bộ dựa vào đáp ứng xung 2 chiều gần với đường cong Gauss. Bộ lọc này đóng vai trò “dự đoán” với sai số  $e_1(m, n)$  tính bởi:

$$e_1(m, n) = x(m, n) - x_1(m, n) \quad (7.31)$$

Như vậy là mã hóa của  $x_1(m, n)$  và  $e_1(m, n)$  là tương đương với mã hóa của  $x(m, n)$ . Với cách biến đổi như trên  $e_1(m, n)$  thuộc loại dải cao. Vì mắt người ít cảm nhận được tín hiệu với tần số cao nên ta có thể dùng một lượng bit ít hơn để mã hóa cho nó. Mặt khác tín hiệu  $x_1(m, n)$  thuộc loại dải thấp, nên theo lý thuyết sẽ lấy mẫu số mẫu sẽ ít hơn.

Quá trình này được lặp lại bằng cách dùng các bộ lọc thấp khác nhau và ta sẽ thu được các tín hiệu  $x_i(m, n)$ ,  $i=1,2,\dots$ . Với mỗi lần lặp kích thước của ảnh sẽ giảm đi một lượng bằng  $\frac{f_i}{f_{i+1}}$ .

Theo cách này, ta có một cấu trúc xếp chồng tự như cấu trúc Kim tự tháp mà kích thước giảm dần từ gốc đến đỉnh. Nhân chập Gauss được dùng ở đây có kích thước 5x5. Các tín hiệu ra sau đó được lượng hóa và mẫu hóa.

Theo kết quả đã công bố [6] với bộ lọc giải thấp một chiều tách được với các trọng số:  $g(0) = 0,7$ ,  $g(-1) = g(1) = 0,25$  và  $g(-2) = g(2) = 0,1$ . Tỷ số nén dao động từ 6/1 đến 32/1. Tuy nhiên, nếu tỷ số nén cao thì ảnh kết quả sẽ có biến dạng.

### 7.4.2 Phương pháp mã hóa dựa vào biểu diễn ảnh.

Như đã biết, trong xử lý ảnh tùy theo các ứng dụng mà ta cần toàn bộ ảnh hay chỉ những đặc tính quan trọng của ảnh. Các phương pháp phân vùng ảnh trong chương sáu như hợp vùng, tách, tách và hợp là rất hữu ích và có thể để nén ảnh. Có thể có nhiều phương pháp khác, song dưới đây chúng ta chỉ đề cập đến hai phương pháp: vùng gia tăng và phương pháp tách hợp.

#### a. Mã hóa dựa vào vùng gia tăng

Kỹ thuật vùng gia tăng thực chất là hợp các vùng có cùng một tính chất nào đó. Kết quả của nó là một ảnh được phân đoạn giống như một ô trong trò xếp chữ (Puzzle). Tuy nhiên, cần lưu ý rằng tất cả các đường bao thu được không tạo nên một ảnh giống ảnh gốc.

Việc xác định tính chất miền đồng nhất xác định độ phức tạp của phương pháp. Để đơn giản, tiêu chuẩn chọn ở đây là *khoảng mức xám*. Như vậy, miền đồng nhất là tập hợp các điểm ảnh có mức xám thuộc khoảng đã chọn. Cũng cần lưu ý thêm rằng, ảnh gốc có thể có đường bao và các kết cấu (Texture). Trong miền texture, độ xám biến đổi rất chậm. Do vậy, nếu không chú ý sẽ chia ảnh thành quá nhiều miền và gây nên các bao giả. Giải pháp để khắc phục hiện tượng này là ta dùng một bộ lọc thích hợp hay lọc trung vị.

Sau giai đoạn này, ta thu được ảnh phân đoạn với các đường viền kín, độ rộng 1 pixel. Để loại bỏ các đường bao giả, ta có thể dùng phương pháp gradient (xem chương năm). Sau khi đã thu được các đường bao đúng, người ta tiến hành mã hóa (xấp xỉ) đường bao bởi các đường cong trong hình học, thí dụ bởi các đoạn thẳng hay đường cong. Nếu ảnh gốc có độ phân giải không thích hợp, người ta dùng khoảng 1,3 bit cho một điểm biên.

Phương pháp này thể hiện ưu điểm: đó là mô hình tham số. Các tham số ở đây là số vùng, độ chính xác mô tả. Tuy nhiên, tham số khoảng mức xám là quan trọng nhất vì nó có ảnh hưởng đến tỉ số nén. Một tham số cũng không kém phần quan trọng là số điểm của các đường bao bị coi là giả. Thường số điểm này không vượt quá 20 điểm.

#### b. Phương pháp tách – hợp

Cũng như đã chỉ ra trong chương sáu, phương pháp tách – hợp khắc phục được một số nhược điểm của phương pháp phân vùng dựa vào tách vùng hay hợp vùng. Trong phương pháp mã hóa này, người ta thay tiêu chuẩn chọn vùng đơn giản ở trên bằng một tiêu chuẩn khác hiệu quả hơn.

Nguyên tắc chung của phương pháp mô hình *biên – texture*. Nhìn chung đường biên dễ nhạy cảm với mắt người, còn texture thì ít nhạy cảm hơn. Người ta mong muốn rằng đường phân ranh giữa các vùng là đồng nhất với các đường bao. Lưu ý rằng cần quyết định phân vùng một phần của ảnh sao cho nó không được vắt chéo đường bao. Đây là một tiêu chuẩn kiểm tra quan trọng. Các đường bao thường nhận được bởi các bộ lọc thông cao, đẳng hướng.

Để có thể quản lý các điểm thuộc một vùng một cách tốt hơn, tiêu chuẩn kiểm tra thứ hai cũng được xem xét đó là dấu: “*các điểm nằm về một phía của đường bao có cùng dấu*”.

Nhìn chung, phương pháp gồm hai giai đoạn. giai đoạn đầu thực hiện việc tách vùng, giai đoạn sau thực hiện việc hợp vùng.

Quá trình tách thực hiện trước. Người ta chia ảnh gốc thành các vùng nhỏ kích thước 9x9. Tiếp theo, tiến hành xấp xỉ các vùng ảnh đó bằng một đa thức có bậc nhỏ hơn 3. Sau quá trình tách ta thu được trong một số vùng của ảnh các hình vuông liên tiếp. chúng sẽ tạo nên một miền gốc lớn và không nhất thiết vuông. Như vậy, trong trường hợp này phải xấp xỉ bằng rất nhiều các

đa thức giống nhau. Rõ ràng là việc mã hóa riêng biệt các đa thức là điều kiện hiệu quả và người nghĩ đến hợp các vùng để giảm độ dư thừa này.

Quá trình hợp được tiến hành như sau: nếu hai vùng có thể được xấp xỉ bởi 2 đa thức tương tự, người ta hợp chúng làm một và chỉ dùng một đa thức xấp xỉ. Nếu mức độ thay đổi là thấp, ta sẽ có nhiều cặp vùng tương tự. Để có thể nhận được kết quả không phụ thuộc vào lần hợp đầu, người ta xây dựng đồ thị “vùng kề cận”. Các nút của đồ thị này là các vùng và các liên hệ biểu diễn mối không tương đồng. Sự liên hệ với mức không tương đồng thấp chỉ ra rằng hai vùng cần hợp lại.

Sau bước hợp này, đồ thị được cập nhật lại và quá trình hợp được lặp lại cho đến khi tiêu chuẩn là thỏa mãn. Quá trình hợp dừng có thể quyết định bởi chất lượng ảnh nén hay một tiêu chuẩn nào khác.

Ta có thể thấy rằng phương pháp này khá phức tạp song bù lại nó cho tỉ số nén khá cao 60 trên 1 [6].

## 7.5 KẾT LUẬN

Mỗi phương pháp nén đều có những ưu điểm và nhược điểm. Tính hiệu quả của phương pháp không chỉ phụ thuộc vào tỉ số nén mà còn vào nhiều chỉ tiêu khác như: độ phức tạp tính toán, nhạy cảm với nhiễu, chất lượng, kiểu ảnh, v.v...

Nén là một vấn đề lớn được quan tâm nhiều và có liên quan đến nhiều lĩnh vực khác nhau. Chúng ta không hi vọng có thể trình bày tất cả trong một chương. Song dù sao, chương này cũng cung cấp một số khái niệm về các phương pháp khả dụng và một số phương pháp mới về nén dữ liệu nhất là nén ảnh. Bảng tổng kết dưới đây cung cấp cho chúng ta một cách nhìn tương đối toàn diện về các phương pháp nén.

**Bảng so sánh kết quả một số phương pháp nén**

Phương pháp	Tỷ số nén	Độ phức tạp	Chất lượng	Nhạy cảm với nhiễu	Kiểu ảnh
RLC	10	Đơn giản	Rất tốt	Lớn	Nhị phân
Dự đoán	2-4	Đơn giản	Rất tốt	Trung bình	Mọi ảnh
Biến đổi	10-15	Phức tạp	Tốt	Rất kém	Đa cấp xám
Pyramide Laplace	5-10	Trung bình	Tốt	Lớn	Đa cấp xám
Vùng gia tăng	20-30	Phức tạp	Trung bình	Rất lớn	Đa cấp xám
Tách và hợp	60-70	Rất phức tạp	Trung bình	Rất lớn	Đa cấp xám

## CÂU HỎI ÔN TẬP

- Viết một chương trình nén và giải nén theo phương pháp RLC (đơn giản, dọc, ngang, hay kết hợp).



2. Viết một chương trình nén và giải nén theo phương pháp Huffman.
3. Viết một chương trình nén và giải nén theo phương pháp LZW.
4. Viết thủ tục thực hiện biến đổi Cosin thuận.
5. Viết thủ tục thực hiện biến đổi Cosin ngược
6. Viết thủ tục thực hiện lượng tử hóa theo thuật toán Lloyd – Max.

## TÀI LIỆU THAM KHẢO

- [1] *Lương Mạnh Bá, Nguyễn Thanh Thủy*. Nhập môn xử lý ảnh số. Nhà xuất bản . Nhà xuất bản Khoa học và Kỹ thuật, 2003.
- [2] *Võ Đức Khánh, Hoàng Văn Kiém*. Giáo trình xử lý ảnh số. Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh.
- [3] *Nguyễn Kim Sách*. Xử lý ảnh và Video số. Nhà xuất bản Khoa học và Kỹ thuật, 1997.
- [4] *Nguyễn Quốc Trung*. Xử lý tín hiệu và lọc số. Nhà xuất bản Khoa học và Kỹ thuật, 2004.
- [5] *Bhabatosh Chanda, Dwijesh Dutta Mqumder*. Digital Imge Processing and Analysis. Prentice Hall of India, 2001.
- [6] *Thomas Braunl, S. Feyer, W. Rapt, M. Reinhardt* .Parallel Image Processing. Springer, 2000.
- [7] *Willam K. Pratt*. Digital Image Processing: PIKS inside, *Third Edition 2001*, John Wiley & Sons, Inc.

## GIẢI THÍCH TỪ VÀ THUẬT NGỮ VIẾT TẮT

- ACF : AutoCorrelation Function : Hàm tự tương quan
- ADALINE : ADAptive LINear Element
- ART : Adaptive Resonance Networks
- BMP : Bit MaP,
- BP : Back Propagation
- CCD : Charge-Coupled Devices: các thiết bị gộp
- CIE : Commission Internationale d'Eclairage : Một tổ chuẩn màu quốc tế.
- dpi : dot per inch.
- EP : Evolutionary Programming: Lập trình tiến hóa
- FIR : Finite Impulse Response : Bộ lọc đáp ứng xung hữu hạn
- GA : Genetic Algorithm: Giải thuật di truyền, áp dụng nhằm tối ưu toàn cục theo xác suất.
- GIF : Graphics Interchanger Format do hãng ComputerServer Incorporated (Mỹ) đề xuất năm 1990.
- JPEG : Joint Phograph Expert Group : Tên của nhóm nghiên cứu các chuẩn nén cho ảnh, thành lập năm 1982. Tên cũ là IOS. Năm 1986, JPEG chính thức được thành lập.
- KL : Karhunen Loeve, Tên một phép biến đổi ảnh được dùng trong xử lý ảnh
- PLD : Picture Language Description: Mô tả ngôn ngữ ảnh
- PC : Personal Computer: Máy tính cá nhân
- PSF : Point-Spread Function: Hàm trải điểm
- SOM : (Self-Organization Map)

## MỤC LỤC

<b>LỜI NÓI ĐẦU .....</b>	<b>1</b>
<b>CHƯƠNG 1: NHẬP MÔN XỬ LÝ ẢNH.....</b>	<b>3</b>
1.1. GIỚI THIỆU VỀ HỆ THỐNG XỬ LÝ ẢNH.....	3
1.2 NHỮNG VẤN ĐỀ CƠ BẢN TRONG HỆ THỐNG XỬ LÝ ẢNH .....	6
1.2.1 Điểm ảnh (Picture Element).....	6
1.2.2 Độ phân giải của ảnh .....	7
1.2.3 Mức xám của ảnh .....	7
1.2.4 Định nghĩa ảnh số.....	7
1.2.5 Quan hệ giữa các điểm ảnh .....	7
1.2.6 Các thành phần cơ bản của hệ thống xử lý ảnh .....	9
1.3 NHỮNG VẤN ĐỀ KHÁC TRONG XỬ LÝ ẢNH .....	10
1.3.1 Biến đổi ảnh (Image Transform).....	10
1.3.2 Nén ảnh .....	11
CÂU HỎI ÔN TẬP.....	11
<b>CHƯƠNG 2: THU NHẬN ẢNH.....</b>	<b>12</b>
2.1 CÁC THIẾT BỊ THU NHẬN ẢNH .....	12
2.1.1 Bộ cảm biến ảnh .....	12
2.1.2 Hệ tọa độ màu .....	13
2.2 LẤY MẪU VÀ LƯỢNG TỬ HÓA.....	15
2.2.1 Giới thiệu.....	15
2.2.2 Lấy mẫu.....	15
2.2.3 Lượng tử hóa .....	17
2.3 MỘT SỐ PHƯƠNG PHÁP BIỂU DIỄN ẢNH .....	18
2.3.1 Mã loạt dài.....	18
2.3.2 Mã xích.....	18
2.3.3 Mã tứ phân .....	19
2.4 CÁC ĐỊNH DẠNG ẢNH CƠ BẢN .....	19
2.4.1 Khái niệm chung .....	19
2.4.2 Quy trình đọc một tệp ảnh.....	19
2.5 CÁC KỸ THUẬT TÁI HIỆN ẢNH .....	20
2.5.1 Kỹ thuật chụp ảnh.....	20
2.5.2 Kỹ thuật in ảnh .....	20
2.6 KHÁI NIỆM ẢNH ĐEN TRẮNG VÀ ẢNH MÀU .....	22
2.6.1 Ảnh đen trắng .....	22
2.6.2 Ảnh màu .....	23
CÂU HỎI ÔN TẬP.....	23
<b>CHƯƠNG 3 : XỬ LÝ NÂNG CAO CHẤT LƯỢNG ẢNH.....</b>	<b>24</b>

3.1 CẢI THIỆN ẢNH SỬ DỤNG CÁC TOÁN TỬ ĐIỂM.....	24
3.1.1 Tăng độ tương phản (Stretching Contrast) .....	25
3.1.2 Tách nhiễu và phân ngưỡng.....	26
3.1.4 Cắt theo mức (Intensity Level Slicing).....	27
3.1.5 Trích chọn bit (Bit Extraction).....	27
3.1.6 Trừ ảnh .....	28
3.1.7 Nén dải độ sáng. ....	28
3.1.8 Mô hình hóa và biến đổi lược đồ xám .....	28
3.2 CẢI THIỆN ẢNH DÙNG TOÁN TỬ KHÔNG GIAN .....	28
3.2.1 Làm trơn nhiễu bằng lọc tuyến tính: lọc trung bình và lọc dải thông thấp .....	29
3.2.2 Làm trơn nhiễu bằng lọc phi tuyến .....	31
3.2.3 Mặt nạ gờ sai phân và làm nhẵn .....	32
3.2.4. Lọc thông thấp, thông cao và lọc dải thông.....	33
3.2.5. Khuyếch đại và nội suy ảnh.....	33
3.2.6. Một số kỹ thuật cải thiện ảnh nhị phân.....	34
3.3 KHÔI PHỤC ẢNH .....	35
3.3.1. Các mô hình quan sát và tạo ảnh .....	36
3.3.2 Các bộ lọc .....	38
CÂU HỎI ÔN TẬP CHƯƠNG 3.....	39
<b>CHƯƠNG 4: CÁC PHƯƠNG PHÁP PHÁT HIỆN BIÊN.....</b>	<b>40</b>
4.1 GIỚI THIỆU BIÊN VÀ KỸ THUẬT PHÁT HIỆN BIÊN .....	40
4.1.1 Một số khái niệm .....	40
4.1.2 Phân loại các kỹ thuật phát hiện biên .....	41
4.1.3 Quy trình phát hiện biên. ....	41
4.2 PHƯƠNG PHÁP PHÁT HIỆN BIÊN CỤC BỘ.....	42
4.2.1 Phương pháp Gradient.....	42
4.2.2. Đồ biên theo quy hoạch động .....	47
4.2.3 Một số phương pháp khác .....	49
CÂU HỎI VÀ BÀI ÔN TẬP .....	51
<b>CHƯƠNG 5: PHÂN VÙNG ẢNH .....</b>	<b>52</b>
5.1 GIỚI THIỆU .....	52
5.2 PHÂN VÙNG ẢNH THEO NGUỒN BIÊN ĐỘ.....	52
5.3 PHÂN VÙNG ẢNH THEO MIỀN ĐỒNG NHẤT .....	53
5.3.1 Phương pháp tách cây tứ phân.....	53
5.3.2 Phương pháp cục bộ .....	55
5.3.3 Phương pháp tổng hợp.....	55
5.4 PHÂN VÙNG ẢNH THEO KẾT CẤU BỀ MẶT .....	56
5.4.1 Phương pháp thống kê.....	56
5.4.2 Phương pháp cấu trúc .....	58
5.4.3 Tiếp cận theo tính kết cấu.....	59
CÂU HỎI ÔN TẬP CHƯƠNG 5.....	59

<b>CHƯƠNG 6: NHẬN DẠNG ẢNH .....</b>	<b>60</b>
6.1 GIỚI THIỆU .....	60
6.1.1 Không gian biểu diễn đối tượng, không gian diễn dịch.....	60
6.1.2 Mô hình và bản chất của quá trình nhận dạng.....	61
6.2 NHẬN DẠNG DỰA THEO MIỀN KHÔNG GIAN.....	63
6.2.1 Phân hoạch không gian.....	63
6.2.2 Hàm phân lớp hay hàm ra quyết định.....	63
6.2.3 Nhận dạng theo phương pháp thống kê.....	64
6.2.4 Thuật toán nhận dạng dựa vào khoảng cách.....	65
6.3 NHẬN DẠNG DỰA THEO CẤU TRÚC.....	65
6.3.1 Biểu diễn định tính .....	65
6.3.2 Các bước nhận dạng.....	67
6.4 NHẬN DẠNG DỰA THEO MẠNG NƠON.....	67
6.4.1 Giới thiệu mạng nơon .....	67
6.4.2 Nhận dạng ảnh các hình khối .....	79
6.4.3 Nhận dạng ảnh mã vạch .....	81
CÂU HỎI VÀ BÀI TẬP.....	86
<b>CHƯƠNG 7: NÉN DỮ LIỆU ẢNH .....</b>	<b>87</b>
7.1 TỔNG QUAN VỀ NÉN DỮ LIỆU ẢNH.....	87
7.1.1 Một số khái niệm.....	87
7.1.2 Các loại dư thừa dữ liệu .....	87
7.1.3 Phân loại phương pháp nén .....	88
7.2 CÁC PHƯƠNG PHÁP NÉN THỂ HỆ THỨ NHẤT.....	89
7.2.1 Phương pháp mã hóa loạt dài .....	89
7.2.2 Phương pháp mã hóa Huffman.....	89
7.2.3 Phương pháp LZW.....	91
7.2.4 Phương pháp mã hóa khối.....	95
7.2.5. Phương pháp thích nghi.....	97
7.3. PHƯƠNG PHÁP MÃ HÓA DỰA VÀO BIẾN ĐỔI THỂ HỆ THỨ NHẤT: .....	97
7.3.1 Nguyên tắc chung.....	97
7.3.2 Thuật toán mã hóa dùng biến đổi 2 chiều.....	98
7.3.3 Mã hóa dùng biến đổi Cosin và chuẩn JPEG .....	99
7.4 PHƯƠNG PHÁP MÃ HÓA THỂ HỆ THỨ HAI.....	109
7.4.1 Phương pháp Kim tự tháp Laplace (Pyramide Laplace).....	109
7.4.2 Phương pháp mã hóa dựa vào biểu diễn ảnh.....	109
7.5 KẾT LUẬN .....	111
CÂU HỎI ÔN TẬP.....	111
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>113</b>
<b>GIẢI THÍCH TỪ VÀ THUẬT NGỮ VIẾT TẮT.....</b>	<b>114</b>
<b>MỤC LỤC .....</b>	<b>115</b>

# XỬ LÝ ẢNH

Mã số: 412XLA450

Chịu trách nhiệm bản thảo

TRUNG TÂM ĐÀO TẠO BƯU CHÍNH VIỄN THÔNG 1