C BÁCH KHOA HÀ NỘI

LƯƠNG MẠNH BÁ
NGUYỄN THANH THUỶ

Nhập môn Xử lý ảnh số

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

LƯƠNG MẠNH BÁ (chủ biên)
NGUYỄN THANH THUỶ

NHẬP MÔN
XỬ LÝ ẢNH SỐ

(Xuất bản lần thứ tư có chỉnh lý bổ sung)

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI - 2006

Chịu trách nhiệm xuất bản: **PGS, TS. Tô Đăng Hải**

Biên tập: **Đặng Đình Thạch, Ngọc Khuê**

Sửa bản in: **Đình Thạch**

In 800 cuốn khổ 16 x 24 cm tại Công ty cổ phần In Hàng không
Quyết định xuất bản số: 136-2006/CXB/275-06 KHKT
In xong và nộp lưu chiểu tháng 1 năm 2007

LỜI TỰA

(Cho lần xuất bản thứ ba)

Trong lần xuất bản thứ hai, cuốn sách đã được chỉnh lý về nội dung cũng như hình thức. Chúng tôi đã nhận được nhiều e-mail và thư góp ý của các thầy cô, bạn bè đồng nghiệp, và nhiều bạn sinh viên. Để phục vụ bạn đọc được tốt hơn, trong lần xuất bản này chúng tôi đã sửa chữa một số lỗi ấn loát, biên tập và đưa thêm một số bài tập để củng cố các kỹ thuật tính nhằm giúp cho quá trình tự đọc, cũng như giúp cho sinh viên hiểu rõ hơn qui trình và các ký thuật trước khi bắt tay viết các chương trình thử nghiệm hay các ứng dụng.

Chúng tôi xin chân thành cám ơn các thầy cô trong Khoa Công Nghệ thông tin, các Giáo sư, các đồng nghiệp đã cho những ý kiến quan trọng góp phần nâng cao chất lượng cuốn sách. Chúng tôi cũng chân thành cám ơn các bạn sinh viên các trường đại học đã sử dụng tài liệu này trong quá trình học tập, nghiên cứu và cho những góp ý chân tình. Chúng tôi hy vọng cuốn sách sẽ có ích cho các bạn và mong nhận được các ý kiến đóng góp để hoàn thiện hơn nữa cho các lần tái bản sau này. Địa chỉ liên lạc:

Email: balm@it-hut.edu.vn

thuynt@it-hut.edu.vn

Hà Nội, tháng 11 năm 2002

Các tác giả

LỜI MỞ ĐẦU

<Lần xuất bản thứ nhất>

Xử lý ảnh là một lĩnh vực đang được quan tâm và đã trở thành một môn học chuyên ngành của sinh viên hệ kỹ sư, hệ cử nhân ngành Công Nghệ Thông Tin, cũng như một số ngành kỹ thuật khác trong các trường Đại học kỹ thuật. Tuy nhiên, tài liệu và giáo trình còn là một điều nan giải đối với sinh viên. Hiện tại chỉ có ít tài liệu bằng tiếng Anh, tiếng Pháp hay tiếng Nga. Tài liệu bằng tiếng Việt còn rất ít. Với mong muốn đóng góp vào sự nghiệp đào tạo và nghiên cứu trong lĩnh vực này, chúng tôi biên soạn cuốn sách "Nhập môn xử lý ảnh số" để làm tài liệu tham khảo trong việc giảng dạy và triển khai ứng dụng.

Xử lý ảnh có liên quan đến nhiều ngành khác như: hệ thống tin học, lý thuyết thông tin, lý thuyết thống kê, trí tuệ nhân tạo, nhận dạng, v.v. Do đó để có thể tiếp thu kiến thức một cách thuận lợi, tài liệu được phân bố một cách hợp lý giữa các vấn đề, không đi sâu nhiều vào các phát biểu toán học gây khó hiểu, khó theo dõi. Ngoài phần lý thuyết , chúng tôi cũng đề cập tới các kỹ thuật được sử dụng trong xử lý ảnh hiện nay. Các lời giải được thể hiện dưới dạng giải thuật bằng ngôn ngữ thuật giải. Ngoài ra, có một số modul chương trình viết bằng Turbo C được giới thiệu trong phần phụ lục.

Với ý tưởng trên, để trình bày một cách lôgic các vấn đề và các kỹ thuật trong xử lý ảnh, tài liệu được phân bố thành 8 chương và hai phụ lục. Chương 1 là phần nhập môn của xử lý ảnh. Mục đích của chương này nhằm giới thiệu các giai đoạn cơ bản trong xử lý ảnh, các thành phần của một hệ xử lý ảnh số bằng máy tính. Chương 2 giới thiệu về quá trình thu nhận ảnh, cách lấy mẫu và lượng tử hoá ảnh và các kiểu tệp được dùng để lưu trữ ảnh. Chương 3 trình bày các kỹ thuật và các công cụ biến đổi ảnh như biến đổi Fourier, Karhumen Loeve. Chương 4 trình bày các kỹ thuật nâng cao chất lượng ảnh (tiền xử lý ảnh). Ảnh thu nhận được, do nhiều nguyên nhân có thể bị suy biến hoặc là do mục đích của giai đoạn phân tích - xử lý ảnh, chúng cần được tăng cường để có chất lượng cao hơn. Chương 5 và chương 6 là nội dung của giai đoạn phân tích ảnh. Trong phần này, các kỹ thuật xác định biên (một vấn đề chủ yếu trong phân tích ảnh) được đề cập một cách chi tiết bao gồm: kỹ thuật lọc vi phân, kỹ thuật Gradient, Laplace. Tiếp sau là các kỹ thuật phân đoạn ảnh thông dụng và hiệu quả như: Quad-Tree, Merge, làm mảnh biên, nhị phân hoá biên. Chương 7 đề cập tới vấn đề nhận dạng và các ứng dụng của nó trong nhận dạng chữ viết, nhận dạng vân tay. Chương 8 giới thiệu về nén ảnh và các phương pháp nén ảnh hiện

được dùng.

Ngoài ra, trong xử lý ảnh còn nhiều vấn đề rất đáng quan tâm nhưng chưa được trình bày trong tài liệu này như: mô tả đầy đủ về đối tượng, trích chọn đặc trưng, phân lớp ảnh, hoặc khôi phục ảnh từ hình chiếu rất có ích trong các ứng dụng y học.

Trong quá trình biên soạn, chúng tôi đã nhận được các ý kiến rất quý báu, sự giúp đỡ nhiệt tình của các giáo sư và bạn bè đồng nghiệp trong và ngoài khoa.

Chúng tôi xin cảm ơn PGS, TS Nguyễn Văn Ba, ThS Đỗ Văn Uy đã dành nhiều thời gian đọc kỹ bản thảo và cho nhiều ý kiến quý báu.

Chúng tôi cũng bày tỏ lòng biết ơn đối với Ban Chủ nhiệm khoa Công Nghệ Thông Tin - Đại Học Bách Khoa Hà Nội, Hội đồng Khoa học - Đào tạo Khoa đã tạo mọi điều kiện để tài liệu này sớm ra mắt bạn đọc.

Các tác giả

1

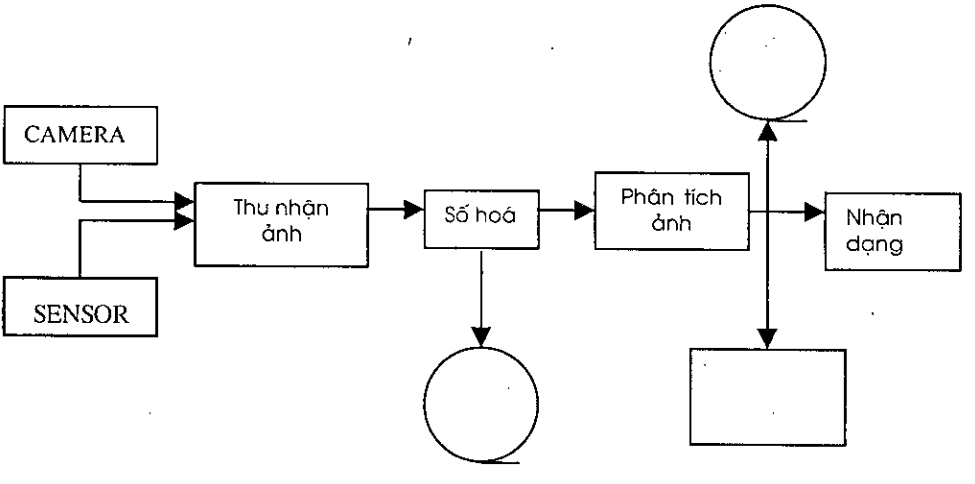
NHẬP MÔN XỬ LÝ ẢNH

INTRODUCTION TO DIGITAL IMAGE PROCESSING

1.1. TỔNG QUAN VỀ MỘT HỆ THỐNG XỬ LÝ ẢNH

Xử lý ảnh là một khoa học còn tương đối mới mẻ so với nhiều ngành khoa học khác, nhất là trên qui mô công nghiệp, song trong xử lý ảnh đã bắt đầu xuất hiện những máy tính chuyên dụng. Để có thể hình dung cấu hình một hệ thống xử lý ảnh chuyên dụng hay một hệ thống xử lý ảnh dùng trong nghiên cứu, đào tạo, trước hết chúng ta sẽ xem xét các bước cần thiết trong xử lý ảnh.

Trước hết là quá trình **thu nhận ảnh**. Ảnh có thể thu nhận qua camera. Thường ảnh thu nhận qua camera là tín hiệu tương tự (loại camera ống kiểu CCIR), nhưng cũng có thể là tín hiệu số hoá (loại CCD - Charge Coupled Device).



Hình 1.1.a. Các giai đoạn chính trong xử lý ảnh

Ảnh cũng có thể thu nhận từ vệ tinh qua các bộ cảm ứng (sensor), hay ảnh, tranh được quét trên scanner. Chi tiết về quá trình thu nhận ảnh sẽ được mô tả trong chương 2. Tiếp theo là quá trình *số hoá* (Digitalizer) để biến đổi tín hiệu tương tự sang tín hiệu rời rạc (lấy mẫu) và số hoá bằng lượng hoá, trước khi chuyển sang giai đoạn xử lý, phân tích hay lưu trữ lại.

Quá trình **phân tích ảnh** thực chất bao gồm nhiều công đoạn nhỏ. Trước hết là công việc tăng cường ảnh (Image Enhancement) để nâng cao chất lượng ảnh. Do những nguyên nhân khác nhau: có thể do chất lượng thiết bị thu nhận ảnh, do nguồn sáng hay do nhiễu, ảnh có thể bị suy biến. Do vậy cần phải tăng cường và khôi phục (Image Restoration) lại ảnh để làm nổi bật một số đặc tính chính của ảnh, hay làm cho ảnh gần giống nhất với trạng thái gốc- trạng thái trước khi ảnh bị biến dạng. Giai đoạn tiếp theo là phát hiện các đặc tính như biên (Edge Detection), phân vùng ảnh (Image Segmentation), trích chọn các đặc tính (Feature Extraction), v.v...

Cuối cùng, tuỳ theo mục đích của ứng dụng, sẽ là giai đoạn nhận dạng, phân lớp hay các quyết định khác. Các giai đoạn chính của quá trình xử lý ảnh có thể mô tả ở hình 1.1.a.

Với các giai đoạn trên, một hệ thống xử lý ảnh (cấu trúc phần cứng theo chức năng) gồm các thành phần tối thiểu như hình 1.1.b.

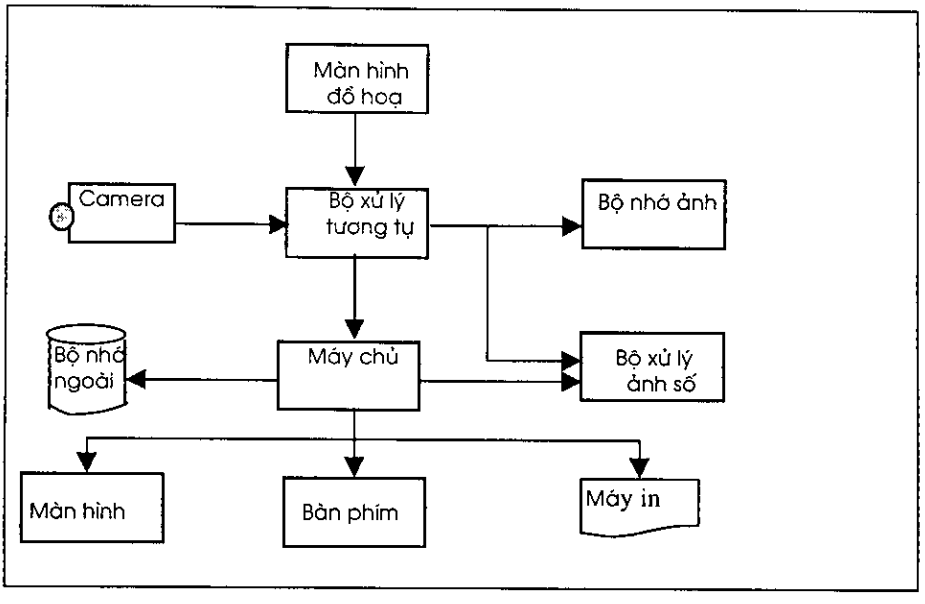
- Đối với một hệ thống xử lý ảnh thu nhận qua camera - camera như là con mắt của hệ thống. Có 2 loại camera: camera ống loại CCIR và camera CCD. Loại camera ứng với chuẩn CCIR quét ảnh với tần số 1/25 và mỗi ảnh gồm 625 dòng. Loại CCD gồm các photo điốt và làm tương ứng một cường độ sáng tại một điểm ảnh với một phần tử ảnh (pixel). Như vậy, ảnh là tập hợp các điểm ảnh. Số pixel tạo nên một ảnh gọi là độ phân giải (resolution).
- Bộ xử lý tương tự (analog processor). Bộ phận này thực hiện các chức năng sau:

  - Chọn camera thích hợp nếu hệ thống có nhiều camera.

  - Chọn màn hình hiển thị tín hiệu.

  - Thu nhận tín hiệu video thu nhận bởi bộ số hoá(digitalizer). Thực hiện lấy mẫu và mã hoá.

  - Tiền xử lý ảnh khi thu nhận: dùng kỹ thuật bảng tra (Look Up Table - LUT).



Hình 1.1b. Các thành phần chính của hệ thống xử lý ảnh.

- Bộ xử lý ảnh số. Gồm nhiều bộ xử lý chuyên dụng: xử lý lọc, trích chọn đường bao, nhị phân hoá ảnh. Các bộ xử lý này làm việc với tốc độ 1/25 giây.
- Máy chủ. Đóng vai trò điều khiển các thành phần miêu tả ở trên.
- Bộ nhớ ngoài: Dữ liệu ảnh cũng như các kiểu dữ liệu khác, để có thể chuyển giao cho các quá trình khác, nó cần được lưu trữ. Để có một ước lượng, xét thí dụ sau: một ảnh đen trắng cỡ 512 x 512 với 256 mức xám chiếm 256K bytes ($2^9 * 2^9 = 2^8 * 2^{10}$). Với một ảnh màu cùng kích thước dung lượng sẽ tăng gấp 3 lần.

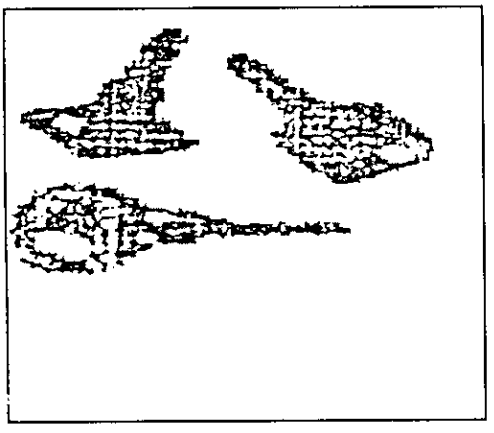
1.2 CÁC VẤN ĐỀ CƠ BẢN TRONG XỬ LÝ ẢNH

Như đã đề cập trong phần giới thiệu, chúng ta đã thấy được một cách khái quát các vấn đề chính trong xử lý ảnh. Để hiểu chi tiết hơn, trước tiên ta xem xét hai khái niệm (thuật ngữ) thường dùng trong xử lý ảnh đó là pixel (phần tử ảnh) và grey level (mức xám), tiếp theo là tóm tắt các vấn đề chính.

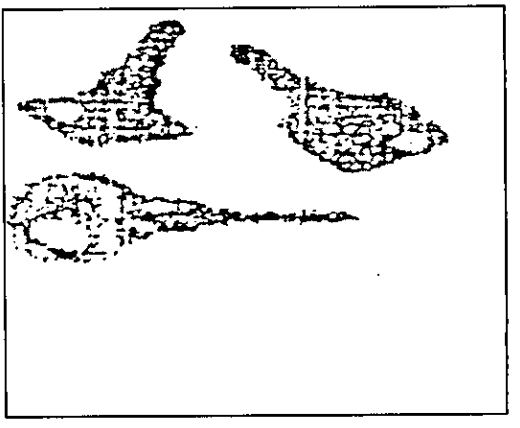
1.2.1. Một số khái niệm

- **Pixel** (Picture Element): phần tử ảnh

Ảnh trong thực tế là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính cần thiết phải tiến hành số hoá ảnh. Trong quá trình số hoá , người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng hoá thành phần giá trị (rời rạc hoá biên độ giá trị) mà về nguyên tắc bằng mắt thường không phân biệt được hai mức kề nhau. Trong quá trình này, người ta sử dụng khái niệm *Picture element* mà ta quen gọi hay viết là *Pixel* - phần tử ảnh. Ở đây cũng cần phân biệt khái niệm pixel hay đề cập đến trong các hệ thống đồ hoạ máy tính. Để tránh nhầm lẫn ta tạm gọi khái niệm pixel này là pixel thiết bị. Khái niệm pixel thiết bị có thể xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ hoạ), màn hình không liên tục mà gồm nhiều điểm nhỏ, gọi là pixel. Mỗi pixel gồm một cặp toạ độ x, y và màu.



a)Ảnh với độ phân giải 128 x128



b)Ảnh với độ phân giải 64 x 64

Hình 1.2. Biểu diễn ảnh với độ phân giải khác nhau.

Cặp toạ độ x, y tạo nên *độ phân giải* (resolution). Như màn hình máy tính có nhiều loại với độ phân giải khác nhau: màn hình CGA có độ phân giải là 320 x 200; màn hình VGA là 640 x 350,...

Như vậy, một ảnh là một tập hợp các điểm ảnh. Khi được số hoá, nó thường được biểu diễn bởi bảng hai chiều I(n,p): n dòng và p cột. Ta nói ảnh gồm n x p pixels. Người ta thường kí hiệu I(x,y) để chỉ một pixel. Thường giá trị của n chọn bằng p và bằng 256. Hình 1.2 cho ta thấy việc biểu diễn một ảnh với độ phân giải khác nhau. Một pixel có thể lưu trữ trên 1, 4, 8 hay 24 bit.

- **Gray level**: Mức xám

Mức xám là kết quả sự mã hoá tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng hoá. Cách mã hoá kinh điển thường dùng 16, 32 hay 64 mức. Mã hoá 256 mức là phổ dụng nhất do lý do kỹ thuật. Vì $2^8 = 256$ (0, 1, ..., 255), nên với 256 mức, mỗi pixel sẽ được mã hoá bởi 8 bit.

1.2.2. Biểu diễn ảnh

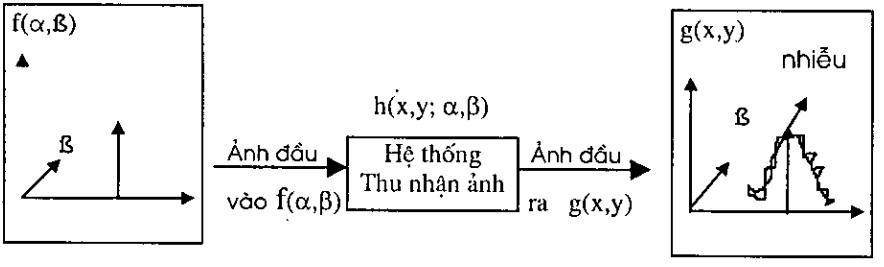
Trong biểu diễn ảnh, người ta thường dùng các phần tử đặc trưng của ảnh là pixel. Nhìn chung có thể xem một hàm hai biến chứa các thông tin như biểu diễn của một ảnh. Các mô hình biểu diễn ảnh cho ta một mô tả lôgic hay định lượng các tính chất của hàm này. Trong biểu diễn ảnh cần chú ý đến tính trung thực của ảnh hoặc các tiêu chuẩn "thông minh" để đo chất lượng ảnh hoặc tính hiệu quả của các kỹ thuật xử lý.

Việc xử lý ảnh số yêu cầu ảnh phải được mẫu hoá và lượng tử hoá. Thí dụ một ảnh ma trận 512 dòng gồm khoảng 512 x 512 pixel. Việc lượng tử hoá ảnh là chuyển đổi tín hiệu tương tự sang tín hiệu số (Analog Digital Convert) của một ảnh đã lấy mẫu sang một số hữu hạn mức xám. Vấn đề này sẽ trình bày chi tiết trong chương 2.

Một số mô hình thường được dùng trong biểu diễn ảnh: Mô hình toán, mô hình thống kê. Trong mô hình toán, ảnh hai chiều được biểu diễn nhờ các hàm hai biến trực giao gọi là các *hàm cơ sở*. Các biến đổi này sẽ trình bày kỹ trong chương 3. Với mô hình thống kê, một ảnh được coi như một phần tử của một tập hợp đặc trưng bởi các đại lượng như: kỳ vọng toán học, hiệp biến, phương sai, moment.

1.2.3. Tăng cường ảnh - khôi phục ảnh

Tăng cường ảnh là bước quan trọng, tạo tiền đề cho xử lý ảnh. Nó gồm một loạt các kỹ thuậy như: tăng cường độ tương phản, khử nhiễu, nổi màu, v...v.



Hình 1.3 Ảnh biến dạng do nhiễu.

Hình 1.3 ở trên cho ta thí dụ về sự biến dạng của ảnh do nhiễu.

Khôi phục ảnh là nhằm loại bỏ các suy giảm (degradation) trong ảnh. Với một hệ thống tuyến tính, ảnh của một đối tượng có thể biểu diễn bởi:

$$g(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h(x,y;\alpha,\beta)f(\alpha,\beta)d\alpha d(\beta+\eta(x,y))$$

Trong đó:

- $\eta(x,y)$ là hàm biểu diễn nhiễu cộng.
- $f(\alpha, \beta)$ là hàm biểu diễn đối tượng.
- $g(x,y)$ là ảnh thu nhận.
- $h((x,y; \alpha, \beta)$ là hàm tán xạ điểm (Point Spread Function - PSF).

Một vấn đề khôi phục ảnh tiêu biểu là tìm một xấp xỉ của $f(\alpha, \beta)$ khi PSF của nó có thể đo lường hay quan sát được, ảnh mờ và các tính chất sác xuất của quá trình nhiễu.

1.2.4. Biến đổi ảnh

Thuật ngữ biến đổi ảnh (Image Transform) thường dùng để nói tới một lớp các ma trận đơn vị và các kỹ thuật dùng để biến đổi ảnh. Cũng như các tín hiệu một chiều được biểu diễn bởi một chuỗi các hàm cơ sở, ảnh cũng có thể được biểu diễn bởi một chuỗi rời rạc các ma trận cơ sở gọi là *ảnh cơ sở*. Phương trình ảnh cơ sở có dạng:

$A^*_{k,l} = a_k a_l^{*T}$, với a_k là cột thứ k của ma trận A. A là ma trận đơn vị. Có nghĩa là $A A^{*T} = I$. Các $A^*_{k,l}$ định nghĩa ở trên với k,l = 0,1, ..., N-1 là ảnh cơ sở. Có nhiều loại biến đổi được dùng như :

- Biến đổi Fourier, Sin, Cosin, Hadamard,. . .
- Tích Kronecker (*[*])
- Biến đổi KL (Karhumen Loeve): biến đổi này có nguồn gốc từ khai triển của các quá trình ngẫu nhiên gọi là phương pháp trích chọn các thành phần chính.

Do phải xử lý nhiều thông tin, các phép toán nhân và cộng trong khai triển là khá lớn. Do vậy, các biến đổi trên nhằm làm giảm thứ nguyên của ảnh để việc xử lý ảnh được hiệu quả hơn. Các biến đổi này sẽ được mô tả chi tiết trong chương 3.

---

[*] Trong xử lý ảnh, việc phân tích có thể được đơn giản hơn khá nhiều do làm việc với ma trận khối gọi là tích Kronecker

- *Ma trận khối là ma trận mà các phần tử của nó lại là một ma trận.*

$$\begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ \dots & \dots & \dots & \dots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{bmatrix}$$

Ma trận *A*

với $A_{i,j}$ là ma trận $m \times n$; $i = 1, 2, \dots, m$ và $j = 1, 2, \dots, n$.

Tích Kronecker

Cho A là ma trận kích thước $M_1 \times M_2$ và B ma trận kích thước $N_1 \times N_2$.

Tích Kronecker của A và B ký hiệu là $A \otimes B$ là ma trận khối được định nghĩa:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & A_{1,M2}B \\ \dots & \dots & \dots & \dots \\ a_{M1,1}B & a_{m1,2}B & \dots & A_{m1,M2}B \end{bmatrix}$$

với $a_{i,j}$ là các phần tử của ma trận A.

Thí dụ **ma trận A** **ma trận B**

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

thì $A \otimes B = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \\ 1 & 2 & -1 & -2 \\ 3 & 4 & -3 & -4 \end{bmatrix}$

1.2.5. Phân tích ảnh

Phân tích ảnh liên quan đến việc xác định các độ đo định lượng của một ảnh để đưa ra một mô tả đầy đủ về ảnh. Các kỹ thuật được sử dụng ở đây nhằm mục đích xác định biên của ảnh. Có nhiều kỹ thuật khác nhau như lọc vi phân hay dò theo quy hoạch động. Vấn đề xác định biên cùng các kỹ thuật liên quan sẽ được trình bày trong chương 5.

Người ta cũng dùng các kỹ thuật để phân vùng ảnh. Từ ảnh thu được, người ta tiến hành kỹ thuật tách (split) hay hợp (fusion) dựa theo các tiêu chuẩn đánh giá như: màu sắc,

cường độ, v...v. Các phương pháp được biết đến như Quad-Trec, mảnh hoá biên, nhị phân hoá đường biên. Cuối cùng, phải kể đến các kỹ thuật phân lớp dựa theo cấu trúc. Vấn đề này được trình bày trong chương 6.

1.2.6. Nhận dạng ảnh

Nhận dạng ảnh là quá trình liên quan đến các mô tả đối tượng mà người ta muốn đặc tả nó. Quá trình nhận dạng thường đi sau quá trình trích chọn các đặc tính chủ yếu của đối tượng. Có hai kiểu mô tả đối tượng:

- Mô tả tham số (nhận dạng theo tham số).

- Mô tả theo cấu trúc (nhận dạng theo cấu trúc).

Trên thực tế, người ta đã áp dụng kỹ thuật nhận dạng khá thành công với nhiều đối tượng khác nhau như: nhận dạng ảnh vân tay, nhận dạng chữ (chữ cái, chữ số, chữ có dấu).

Nhận dạng chữ in hoặc đánh máy phục vụ cho việc tự động hoá quá trình đọc tài liệu, tăng nhanh tốc độ và chất lượng thu nhận thông tin từ máy tính.

Nhận dạng chữ viết tay (với mức độ ràng buộc khác nhau về cách viết, kiểu chữ, v.v...) phục vụ cho nhiều lĩnh vực.

Ngoài hai kỹ thuật nhận dạng trên, hiện nay một kỹ thuật nhận dạng mới dựa vào kỹ thuật mạng nơron đang được áp dụng và cho kết quả khả quan. Một số khái niệm về mạng nơ ron cũng như một ứng dụng mạng nơron trong nhận dạng ký tự sẽ được đề cập đến trong chương 7.

1.2.7. Nén ảnh

Dữ liệu ảnh cũng như các dữ liệu khác cần phải lưu trữ hay truyền đi trên mạng. Như đã nói ở trên, lượng thông tin để biểu diễn cho một ảnh là rất lớn. Trong phần 1.1 chúng ta đã thấy một ảnh đen trắng cỡ 512 x 512 với 256 mức xám chiếm 256K bytes. Do đó làm giảm lượng thông tin hay nén dữ liệu là một nhu cầu cần thiết. Nhiều phương pháp nén dữ liệu đã được nghiên cứu và áp dụng cho loại dữ liệu đặc biệt này. Các kĩ thuật nén ảnh sẽ được trình bày trong chương 8.

Bài tập chương I

Để có ảnh cho các phần sau, hãy tạo một ảnh bằng một số phần mềm có sẵn (như Paint Brusch) hoặc xây dựng một ảnh gồm *một số hình chữ nhật* theo thuật toán sau:

. Mỗi hình chữ nhật cho các toạ độ góc trên, góc dưới cũng như mức xám tương ứng (số mức xám hoặc nhập vào hay reo ngẫu nhiên một số từ 0 đến 255). Dữ liệu nhập vào từ bàn phím.

- Quá trình dừng khi cho mức xám là -1.

- Lưu ảnh lên tệp để cho các xử lý tiếp sau.

Chương trình viết trong bằng ngôn ngữ tự chọn.

2

THU NHẬN ẢNH

IMAGE REPRESENTATION AND MODELING

Chương này giới thiệu quá trình thu nhận ảnh cũng như các thiết bị dùng trong hệ thống xử lý ảnh. Tiếp theo là quá trình lấy mẫu và lượng tử hoá ảnh. Đồng thời cũng giới thiệu một số phương pháp biểu diễn ảnh, các kiểu tệp và cấu trúc của chúng dùng trong lưu trữ ảnh như .IMG, .PCX,TIFF, JPEG.... Cuối cùng, trình bày nguyên tắc tái hiện ảnh gồm các kỹ thuật Bayer Dithering, Rylander Pattern Matrix....

2.1. CÁC THIẾT BỊ THU NHẬN ẢNH VÀ KỸ THUẬT PHÂN TÍCH MÀU

2.1.1. Thiết bị thu nhận ảnh

Một hệ thống xử lý ảnh có thể trang bị kèm theo các hệ thống thông tin địa lý - GIS (Geographical Information System) hay hệ MORPHO(giá khoảng7 đến 8 triệu USD) hoặc có thể là hệ thống máy tính cá nhân. Các thiết bị thu ảnh thông thường gồm máy quay (camera) cộng với bộ chuyển đổi tương tự số AD(Analog to Digital) hoặc máy quét (scanner) chuyên dụng.

Các thiết bị thu nhận ảnh này có thể cho ảnh trắng đen B/W (Black & White) với mật độ từ 400 đến 1600 dpi (dot per inch) hoặc ảnh màu 600 dpi. Với ảnh B/W mức màu z là 0 hoặc 1. Với ảnh đa cấp xám, mức xám biến thiên từ 0 đến 255. Ảnh màu, mỗi điểm ảnh lưu trữ trong 3 bytes và do đó ta có $2^{8x3} = 2^{24}$ màu (cỡ 16,7 triệu màu).

Các thiết bị thu nhận ảnh này có thể cho ảnh trắng đen B/W (Black & White) với mật độ từ 400 đến 1600 dpi (dot per inch) hoặc ảnh màu 600 dpi. Với ảnh B/W mức màu z là 0 hoặc 1. Với ảnh đa cấp xám, mức xám biến thiên từ 0 đến 255. Ảnh màu, mỗi điểm ảnh lưu trữ trong 3 bytes và do đó ta có $2^{8x3} = 2^{24}$ màu (cỡ 16,7 triệu màu).

Khi dùng scanner, một dòng photodiot sẽ quét ngang ảnh (quét theo hàng) và cho ảnh với độ phân giải ngang khá tốt. Đầu ra của scanner là ảnh ma trận số mà ta quen gọi là bản đồ ảnh (ảnh Bitmap). Bộ số hoá (digitalizer) sẽ tạo ảnh vector có hướng.

Trong xử lý ảnh bằng máy tính, ta không thể không nói đến thiết bị monitor (màn hình) để hiện ảnh. Monitor có nhiều loại khác nhau:

- CGA : 640 x 320 x 16 màu,
- EGA : 640 x 350 x 16 màu,
- VGA : 640 x 480 x 16 màu,
- SVGA: 1024 x 768 x 256 màu.

Với ảnh màu, có nhiều cách tổ hợp màu khác nhau. Theo lý thuyết màu do Thomas đưa ra từ năm 1802, mọi màu đều có thể tổ hợp từ 3 màu cơ bản: Red (đỏ), Green (lục) và Blue (lơ).

Thiết bị ra ảnh có thể là máy in đen trắng, máy in màu hay máy vẽ (ploter). Máy vẽ cũng có nhiều loại: loại dùng bút, loại phun mực.

Nhìn chung, các hệ thống thu nhận ảnh thực hiện 2 quá trình:

- Cảm biến: biến đổi năng lượng quang học (ánh sáng) thành năng lượng điện.
- Tổng hợp năng lượng điện thành ảnh điện.

2.1.2. Biểu diễn màu

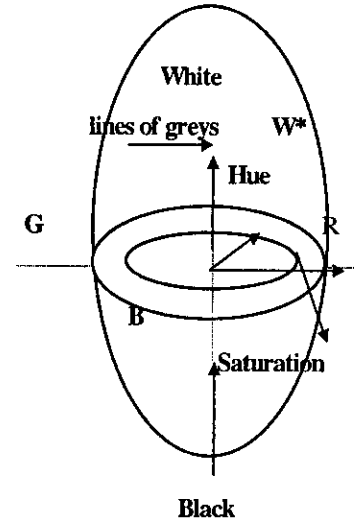
Ánh sáng màu là tổ hợp của ánh sáng đơn sắc (monochrome). Mắt người chỉ có thể cảm nhận được vài chục màu, song lại có thể phân biệt được tới hàng ngàn màu. Có 3 thuộc tính chủ yếu trong cảm nhận màu:

- Brightness: sắc màu, còn gọi là độ chói.
- Hue : sắc lượng, còn gọi là sắc thái màu.
- Saturation: độ bão hoà.

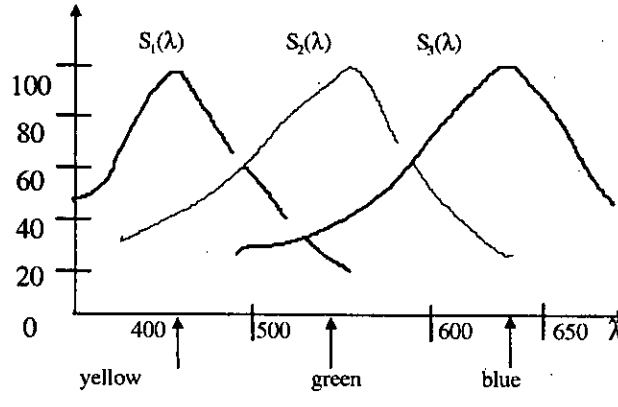
Với nguồn sáng đơn sắc, độ *hue* tương ứng với bước sóng λ. Độ bão hoà thay đổi nhanh nếu ta thêm lượng ánh sáng trắng. Hình 2.1 mô tả mối liên quan giữa các đại lượng trên và 3 màu chủ yếu R, G và B.

Với một điểm W* cố định, các kí hiệu G, R, B chỉ vị trí tương đối của các phổ màu đỏ, lục và lơ. Do sự tán sắc ánh sáng (ứng với khai triển Fourier) mà ta nhìn rõ màu. Theo Maxwell, trong võng mạc mắt có 3 loại tế bào hình nón cảm thụ 3 màu cơ bản ứng với 3 phổ hấp thụ $S_1(\lambda)$, $S_2(\lambda)$ và $S_3(\lambda)$. λ_{min}= 380 nm; λ_{max} = 780 nm.

- Một màu bất kỳ sẽ là một điểm trên vòng tròn.
- Nếu màu đen và màu trắng là như nhau thì đường tròn là lớn nhất và R là điểm bão hoà.
- S thay đổi theo bán kính
- H thay đổi theo góc θ
- W* là sắc màu

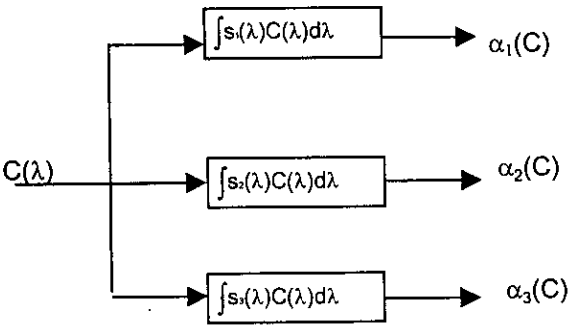


Hình 2.1. Hệ toạ độ màu RGB.



Hình 2.2. Các đường cong cảm nhận S_1, S_2 và S_3.

Theo lý thuyết 3 màu, phân bố phổ năng lượng của một nguồn sáng màu ký hiệu là C(λ) và tổ hợp màu theo nguyên tắc 3 màu có thể mô tả bằng hình 2.3 dưới đây:



Hình 2.3. Nguyên tắc tổ hợp màu.

Do đó, $\alpha_i(C) = \int_{\lambda_{min}}^{\lambda_{max}} S_i(\lambda)c(\lambda)d\lambda$ với i = 1, 2, 3. (2.1)

$\alpha_i(C)$ gọi là đáp ứng phổ (spectral responses).

Phương trình 2.1 gọi là phương trình biểu diễn màu. Nếu $C_1(\lambda)$ và $C_2(\lambda)$ là hai phân bố phổ năng lượng tạo nên các đáp ứng phổ $\alpha_1(C_1)$ và $\alpha_2(C_2)$ mà $\alpha_i(C_1) = \alpha_i(C_2)$, với i =1, 2, 3 thì hai màu C_1 và C_2 là như nhau (sánh được).

2.1.3. Tổng hợp màu và sánh màu

Một trong các vấn đề cơ bản của lý thuyết biểu diễn màu là sử dụng một tập các nguồn sáng (màu) để biểu diễn màu. Theo lý thuyết 3 màu của Thomas, người ta hạn chế số màu còn 3 màu cơ bản: đỏ, lục và lơ. Giả sử rằng ba nguồn sáng cơ bản có phân phối phổ năng lượng là $p_k(\lambda)$ với k =1, 2, 3 và:

$$\int_{\chi_{max}}^{\chi_{min}} p_k(\lambda)d\lambda = 1$$

Để sánh một màu $C(\lambda)$, giả sử rằng 3 màu cơ bản được tổ hợp theo tỉ lệ $\beta_k(\lambda)$, k =1, 2, 3, như vậy:

$\sum_{k=1}^{3} \beta_k(\lambda)p_k(\lambda)$ sẽ cho $C(\lambda)$. Thay giá trị này vào phương trình 2.1 ta có:

$$\alpha_i(C) = \int_{\lambda_{min}}^{\lambda_{max}} S_i(\lambda)[\sum_{k=1}^{3} \beta_k(\lambda)p_k(\lambda)]\, d\lambda = \sum_{k=1}^{3} \beta_k(\lambda) \int_{\chi_{max}}^{\chi_{min}} p_k(\lambda)S_i(\lambda)d\lambda$$

$$= \sum_{k=1}^{3} \beta_k(\lambda)a_{i,k} \quad \text{với } a_{i,k} = \int_{\chi_{max}}^{\chi_{min}} p_k(\lambda)S_i(\lambda)d\lambda$$

Như vậy, có thể tổng hợp màu theo phép cộng: màu X = α_1 đỏ + α_2 xanh + α_3 lơ với α_1, α_2 và α_3 là các hệ số tổng hợp. Phương pháp này hay được dùng trong các ảnh dân dụng.

Lý thuyết tổng hợp màu trên cho phép đưa ra một số luật sánh màu sau:

i) mọi màu có thể sánh bởi nhiều nhất 3 màu.

ii) nguồn sáng của một màu tổng hợp bằng tổng nguồn sáng các màu thành phần.

iii) nếu màu C_1 sánh được với màu C_1' và C_2 sánh được với màu C_2' thì:

- $\alpha_1 C_1 + \alpha_2 C_2 = \alpha_1 C_1' + \alpha_2 C_2'$: luật cộng màu

- nếu $C_1 + C_2 = C_1' + C_2'$ và $C_2 = C_2'$ thì $C_1 = C_2'$.

iv) luật bắc cầu: nếu $C_1 = C_2$ và $C_2 = C_3$ thì $C_1 = C_3$. dấu = ở trên có nghĩa là sánh được.

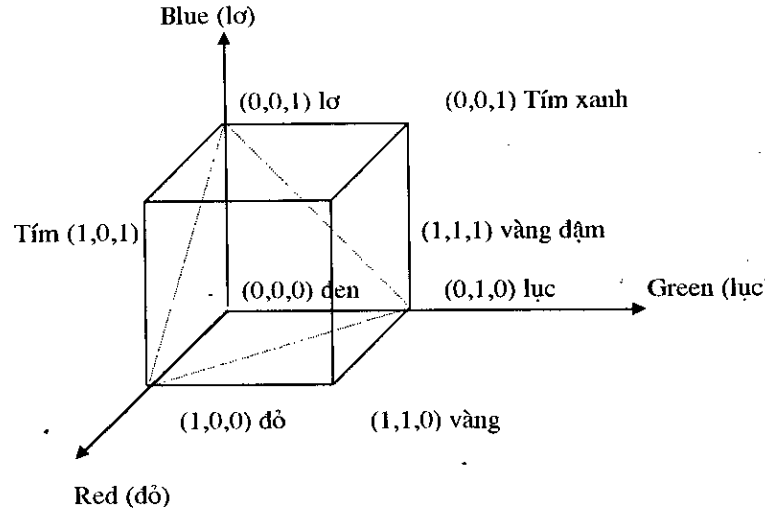
2.1.4. Hệ toạ độ màu

a) Khái niệm

Tổ chức quốc tế về chuẩn hoá màu CIE (Commision Internationale d'Eclairage) đưa ra một số các 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 tỉ lệ nào đấy. Như vậy, một pixel ảnh màu kí hiệu P_x được viết:

$$P_x = \begin{bmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{bmatrix}$$

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



Trong cách biểu diễn này ta có công thức: đỏ + lục + lơ =1. Công thức này gọi là công thức Maxwell. Trong hình vẽ trên, tam giác tạo bởi ba đường đứt đoạn gọi là tam giác Maxwell. Màu trắng trong hệ toạ độ này được tính bởi: trắng$_{CIE}$ = (đỏ $_{CIE}$ = lục $_{CIE}$ = lơ $_{CIE}$ =1).

b)Biến đổi hệ toạ độ màu

Hệ toạ độ màu do CIE đề xuất có tác dụng như một hệ qui 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): 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 $\mathscr{X}$ là không gian biểu diễn màu ban đầu;

- $\mathscr{X}'$ là không gian biểu diễn màu mới;

- A: ma trận biểu diễn phép biến đổi.

Ta có: $\mathscr{X}' = A\mathscr{X}$. Ví dụ, biến đổi hệ toạ độ màu RGB_{CIE} sang hệ toạ độ màu RGB_{NTSC} ta có:

$$P_X = \begin{bmatrix} R_{CIE} \\ G_{CIE} \\ B_{CIE} \end{bmatrix} \quad \text{và } P_{X'} = \begin{bmatrix} R_{NTSC} \\ G_{NTSC} \\ B_{NTSC} \end{bmatrix}$$

Hay viết dưới dạng ma trận ta được:

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

Ma trận A cho các biến đổi khác được trình bày chi tiết trong [12].

2.2. LẤY MẪU VÀ LƯỢNG TỬ HOÁ (IMAGE SAMPLING AND QUANTIZATION)

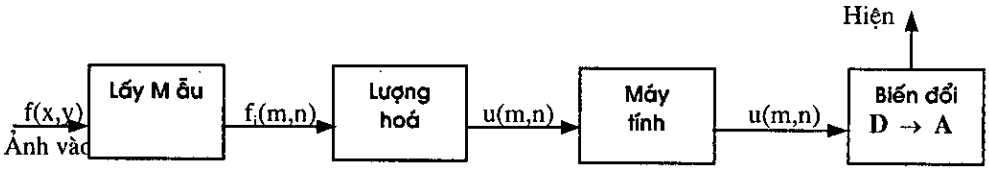
Yêu cầu cơ bản nhất trong xử lý ảnh bằng máy tính là đưa ảnh về dạng biểu diễn số

thích hợp, nghĩa là ảnh phải được biểu diễn bởi một ma trận hữu hạn tương ứng với việc lấy mẫu ảnh trên một lưới rời rạc và mỗi pixel được lượng hoá bởi một số hữu hạn bit. ảnh số được lượng hoá có thể được xử lý hay chuyển qua bước biến đổi số tương tự — DA (Digital to Analog) để tái hiện trên thiết bị hiện ảnh hoặc được lưu trữ lại.

2.2.1. Lấy mẫu (Image sampling)

Phương pháp chung để lấy mẫu là quét ảnh theo hàng và mã hoá từng hàng. Về nguyên tắc, một đối tượng, phim hay giấy trong suốt sẽ được chiếu sáng liên tục để tạo nên một ảnh điện tử trên tấm cảm quang. Tuỳ theo các loại camera mà tấm cảm quang này là chất quang dẫn hay quang truyền. Hệ thống camera ống sử dụng phương pháp *scan-out-digitalizer*; còn hệ thống camera CCD(Charge Couped Device) cho ảnh ma trận.

Camera CCD thực sự là thiết bị mẫu hoá tín hiệu 2 chiều và gọi là phương pháp *sefl-scanning matrix*. Nguyên tắc của 2 phương pháp được minh hoạ qua hình 2-6.



- $x,y \in (-\infty,+\infty)$ - $m \in [1, M], n \in [1, N]$ - $u(m,n) \in [0, L]$
- $f(x,y) \in (-\infty,+\infty)$ - $f(m, n) \in (-\infty,+\infty)$

Hình 2.4. Lấy mẫu và lượng hoá

- **Lý thuyết mẫu hoá 2 chiều**

- Ảnh với dải giới hạn (Band limited Images)

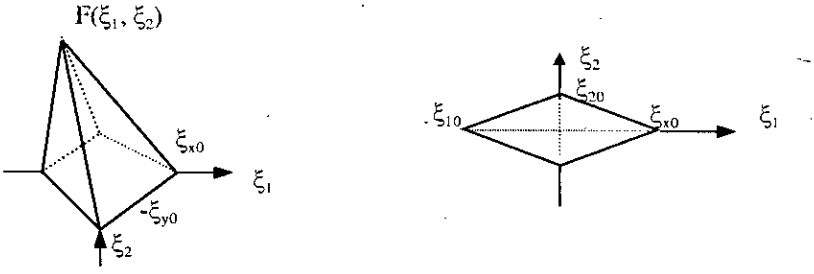
Một hàm f(x,y) gọi là dải giới hạn nếu khai triển Fourier $F(\xi_1, \xi_2)$ của nó là 0 bên ngoài miền bao (hình 2.5). $F(\xi_1, \xi_2) = 0$ với $|\xi_1| > \xi_{x0}$, $|\xi_2| > \xi_{y0}$ (2.2)

Với ξ_{x0} và ξ_{y0} là dải giới hạn theo x và y của ảnh.

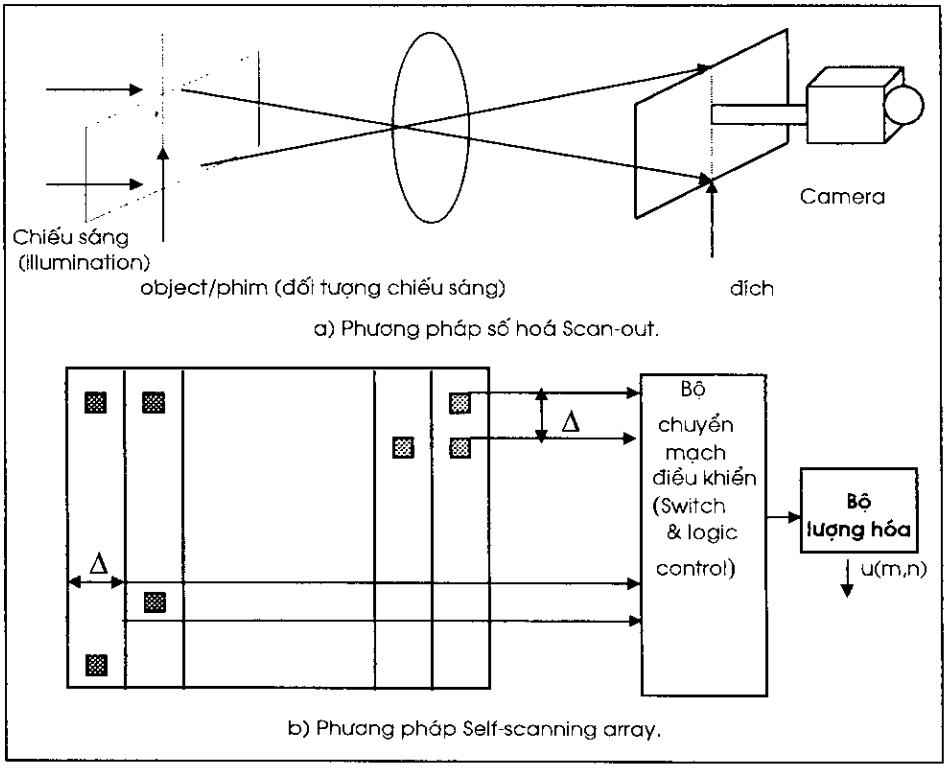
Quá trình số hoá ảnh có thể hiểu như mô hình tín hiệu dải giới hạn. Một ảnh dải giới hạn f(x,y) thoả mãn phương trình 2..2 và được lấy mẫu đều trên một lưới hình chữ nhật với bước nhảy Δx, Δy có thể khôi phục lại không có sai sót dựa trên các giá trị mẫu $f(m\Delta x,n\Delta y)$. Theo lý thuyết lấy mẫu trong xử lý tín hiệu, nếu tần số lấy mẫu theo x, y lớn hơn 2 lần dải giới hạn ξ_{x0}, ξ_{y0} hay tương đương với:

$$\frac{1}{\Delta x} = \xi_{xs} > 2\,\xi_{x0};\ \frac{1}{\Delta y} = \xi_{ys} > 2\,\xi_{y0}$$

thì có thể khôi phục được. Tỉ số này do Nyquist đề xuất và mang tên tỉ số Nyquist.



Hình 2.5. Khai triển Fourier của hàm dải giới hạn.



Hình 2.6. Phương pháp lấy mẫu & lượng hoá ảnh.

Trong xử lý ảnh, tần số lấy mẫu theo trục x và trục y thường được chọn là 8MHz (tần số max theo x và y nằm trong khoảng 4 ÷ 6 MHz). Do vậy chu kỳ lấy mẫu theo cả 2 trục là: $\frac{1}{8MHz} = 0.125 * 10^{-6}$ s

Ngoài ra cần đảm bảo thời gian quét một mặt ảnh là 1/30 s. Việc khôi phục ảnh có thể nội suy theo công thức:

$$f(x,y) = \sum_{m,n=-\infty}^{\infty} f(mx, ny)\left(\frac{\sin(x\xi_{xs} - x)\pi}{(x\xi_{xs} - m)\pi}\right)\left(\frac{\sin(y\xi_{ys} - n)\pi}{(y\xi_{ys} - m)\pi}\right) \quad (2.3)$$

Ảnh sau khi lấy mẫu là ảnh rời rạc và thường được ký hiệu f(m,n), với m, n là các giá trị nguyên. Những giá trị hay dùng của m và n là 256, 512, 1024. Tuy nhiên, giá trị của f(m,n) vẫn là liên tục và trong khoảng (-∝,+∝).

Trong thực tế, nhiễu ngẫu nhiên luôn có mặt trong tín hiệu ảnh. Do đó, lý thuyết lấy mẫu ở trên phải được mở rộng với một số kỹ thuật khác như: lưới không vuông, lưới bát giác. Để đơn giản khi trình bày, những kỹ thuật này không nêu ở đây. Độc giả có quan tâm xin tham khảo tài liệu [1].

2.2.2. Lượng hoá ảnh (Image Quantization)

2.2.2.1. Khái niệm và nguyên tắc lượng hoá ảnh

Lượng hoá ảnh là bước kế tiếp của việc lấy mẫu, nhằm thực hiện một ánh xạ từ một biến liên tục u (biểu diễn giá trị độ sáng) sang một biến rời rạc u* với các giá trị thuộc tập hữu hạn $\{r_1, r_2, \ldots, r_L\}$. Ánh xạ này thường là một hàm bậc thang (hình 2.7) tuân theo nguyên tắc sau:

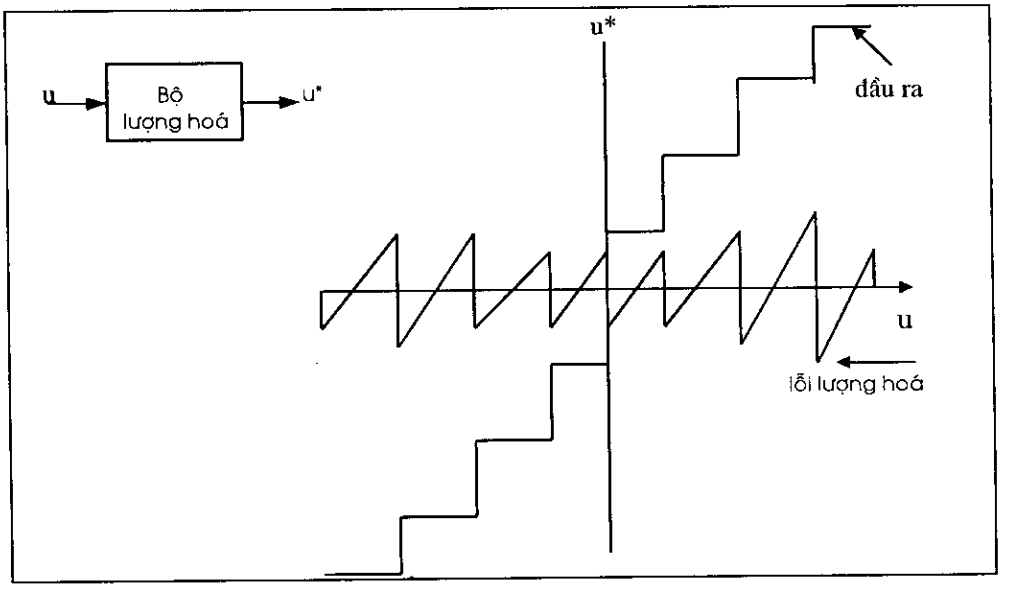
Cho $\{t_k, k=1, 2, \ldots L+1\}$ là một tập các bước dịch chuyển hay mức độ quyết định; t_1 là giá trị nhỏ nhất và t_{L+1} là giá trị lớn nhất của u.

Cơ sở lý thuyết của lượng hoá là dải độ sáng biến thiên hữu hạn từ L_{min} đến L_{max}. Người ta chia dải biến thiên đó thành một số mức (rời rạc và nguyên). Việc chia này phải thoả tiêu chí về độ nhậy của mắt: mắt người không thể phân biệt được các giá trị trong một mức; mắt người chỉ phân biệt được hai mức kề nhau. L_{min} thường là 0, còn L_{max} là một số nguyên dạng 2^q và thường chọn bằng 255 (256 mức khác nhau).

Với các khoảng chia như trên $(t_1, t_2, \ldots, t_k)$, nếu $u \in (t_i, t_{i+1})$ thì gán cho u giá trị i, hay nói cách khác u đã được *lượng hoá bởi mức i* (giá trị i).

Cách đơn giản nhất để lượng hoá là dùng lượng hoá đều. Theo phương pháp này, giả

sử đầu ra của một bộ cảm biến ảnh nhận giá trị từ 0 đến 10.0. Nếu mẫu là lượng hoá đều trên 256 mức, thì bước dịch chuyển t_k và mức xây dựng lại r_k được tính bởi:



Hình 2.7. Mô hình bộ lượng hoá.

$$t_k = \frac{10(k-1)}{256} \quad \text{với } k = 1, 2, ..., 257; \quad r_k = t_k - \frac{5}{256} \quad \text{với } k = 1, 2, ..., 256$$

Đại lượng $q = t_k - t_{k-1} = r_k - r_{k-1}$ là hằng số với các giá trị k và gọi là khoảng lượng hoá.

Trong phần này, ta chỉ xem xét các bộ lượng hoá không bộ nhớ (zero memory quantizer), có nghĩa là đầu ra chỉ phụ thuộc duy nhất là đầu vào. Các bộ lượng hoá kiểu này rất có ích trong kỹ thuật mã hoá ảnh như mã hoá điều xung PCM (Pulse Code Modulation), PCM vi phân, chuyển mã, v...v. Chú ý rằng, ánh xạ lượng hoá này không thuận nghịch, nghĩa là với một đầu ra đã cho, đầu vào là không duy nhất. Vì vậy, người ta đã nghiên cứu bổ sung nhiều kỹ thuật khác nhau để cực tiểu hoá biến dạng, tăng hiệu quả. Một kỹ thuật phổ dụng là *trung bình bình phương cực tiểu* (do Lloyd-max đề xuất) chúng ta sẽ mô tả dưới đây.

2.2.2.2. Kỹ thuật lượng hoá trung bình bình phương cực tiểu

Kỹ thuật này nhằm cực tiểu hoá sai số trung bình bình phương đối với một số mức

lượng hoá đã cho. Cho u là một biến thực ngẫu nhiên với hàm mật độ liên tục $P_u(u)$. Mong muốn ở đây là tìm được mức độ quyết định t_k và mức khôi phục lại r_k với một bộ lượng hoá L mức sao cho sai số trung bình bình phương là nhỏ nhất.

$$\text{Gọi } \varepsilon = E[(u - u^*)^2] = \int_{t_1}^{t_{L-1}} (u - u^*)^2 P_u(u)du \tag{2.4}$$

Nhiệm vụ là tìm min của ε.

Viết lại (2.4) ta có:

$$\varepsilon = \sum_{i=1}^{L} \int_{t_1}^{t_{i+1}} (u - r_i)^2 P_u(u)du \qquad i=0, 1,\ldots, L-1 \tag{2.5}$$

Để tính r_k, ta cần giải hệ phương trình (nhận được khi lấy vi phân 2.5):

$$\begin{cases} (t_k - r_{k-1})^2 P_u(t_k) - (t_k - r_k)^2 P_u(t_k) = 0 \\ 2\int_{t_k}^{t_{k+1}} (u - r_k)P_u(u)du = 0 \end{cases}$$

Lưu ý rằng $t_k \geq t_{k-1}$, do đó giá trị của t_k và r_k cho bởi:

$$t_k = (r_k - r_{k-1})/2 \qquad k = 1, 2, \ldots L \tag{2.6}$$

và

$$r_k = \frac{\int_{t_k}^{t_{k+1}} up_u(u)du}{\int_{t_k}^{t_{k+1}} p_u(u)du} \qquad k = 0, 1, \ldots, L-1 \tag{2.7}$$

Thông thường hệ phương trình (2.6), (2.7) là không tuyến tính.

Kết quả trên chứng tỏ rằng mức dịch chuyển tối ưu nằm trên nửa đường của các mức khôi phục lại. Các mức khôi phục lại tối ưu nằm tại trọng tâm của phân bố mật độ giữa các mức dịch chuyển.

Giải hệ phương trình (2.6) và (2.7) ta thu được các cận t_1 và t_{L+1}. Trong thực tế, người ta hay áp dụng phương pháp Newton để giải phương trình trên. Khi số mức lượng hoá lớn, người ta dùng phương pháp xấp xỉ mật độ xác suất như một hàm hằng *khôn ngoan* (picewise) $p_u(u) = p_u(v_i)$ với $v_i = (t_i + t_{i+1})$; $t_i \quad u < t_{i+1}$. Thay giá trị mới của $p_u(u)$ vào 2.5 và tính cực tiểu hoá, ta có lời giải xấp xỉ cho mức quyết định t_{i+1}[1]:

$$t_{i+1} = \frac{A\int_{t_i}^{t_{k+1}} [p_u(u)]^{-1/3} du}{\int_{t_i}^{t_{i+1}} [p_u(u)]^{-1/3} du} - t_i \qquad (2.8)$$

với $A = t_{L+1} - t_1$ và $r_k = (k/L)A$, k=1,2,...,L. Từ đó ta dễ dàng tính được giá trị của sai số ε. (*)

Các hàm mật độ thường dùng là hàm Gauss và hàm Laplace.

Hàm Gauss có dạng:

$$P_u(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(u-\mu)^2}{2\sigma^2}\right) \qquad (2.9)$$

Hàm Laplace có dạng: $P_u(u) = \alpha/2 * \exp(-\alpha \ u-\mu \ )$ (2.10)

trong đó:

- μ là kỳ vọng toán học.
- σ^2 là hiệp biến với biến ngẫu nhiên u đối với hàm Gauss.

Hiệp biến Laplace được tính bởi $\sigma^2 = 2/\alpha$.

Trường hợp đặc biệt, nếu phân bố là đều thì hệ phương trình (2.6) và (2.7) là tuyến tính và sẽ cho ta các khoảng đều nhau giữa các mức dịch chuyển và mức khôi phục lại. Do vậy, phép lượng hoá này có tên là lượng hoá tuyến tính.

Giả sử hàm mật độ cho bởi công thức:

$$p_u(u) = \begin{cases} 1/(t_{L+1} - t_1) & \text{nếu } t_i \le u \le t_{i+1} \\ 0 & \text{khác đi} \end{cases}$$

Từ phương trình (2.7) ta có:

$$r_k = \frac{t_{2k+1} - t_{2k})}{2(t_{k+1} - t_k)} = \frac{t_{k+1} - t_k}{2} \qquad (2.11)$$

do đó $t_k = (t_{k+1} - t_k)/2 \Rightarrow t_k = t_{k-1} = t_{k+1} - t_k = \text{const} = q$.

Cuối cùng ta có $q = (t_{L+1} - t_1)/L$; $t_k = t_{k-1} + q$; $r_k = t_k - q/2$ (2.12)

Như vậy, mọi mức dịch chuyển và mức khôi phục lại đều cách đều. Sai số của phép lượng hoá là u - u* sẽ phân phối đều trên khoảng (-q/2 , q/2). Sai số trung bình bình phương sẽ là:

$$\varepsilon = \frac{1}{q}\int_{-q/2}^{q/2} u^2 du = \frac{q^2}{12} \qquad (2.13)^*$$

Lượng hoá đều như trên khá thuận tiện cho cài đặt. Tuy nhiên, trong thực tế ta còn gặp nhiều loại phân bố không đều của các biến ngẫu nhiên. Độc giả quan tâm đến các biến đổi này cũng như so sánh kết quả giữa một số phương pháp xin tham khảo tài liệu [1].

2.3. MỘT SỐ PHƯƠNG PHÁP BIỂU DIỄN ẢNH (IMAGE REPRESENTATION)

Sau bước số hoá, ảnh sẽ được lưu trữ hay chuyển sang giai đoạn phân tích. Trước khi đề cập đến vấn đề lưu trữ ảnh, ta cần xem xét ảnh sẽ được biểu diễn ra sao trong bộ nhớ máy tính. Phần trên cũng đã nói đến các mô hình toán học để biểu diễn ảnh. Nếu lưu trữ trực tiếp ảnh thô theo kiểu bản đồ ảnh, dung lượng sẽ khá lớn, tốn kém mà nhiều khi không hiệu quả theo quan điểm ứng dụng. Thường người ta không biểu diễn toàn bộ ảnh thô mà tập trung đặc tả các đặc trưng của ảnh như: biên ảnh (Boundary) hay các vùng ảnh (Region). Các kỹ thuật phát hiện biên hay phân vùng ảnh sẽ được giới thiệu kỹ trong chương 5 và 6. Dưới đây giới thiệu một số phương pháp biểu diễn. Thường người ta dùng:

- Biểu diễn mã loạt dài (Run - Length Code).
- Biểu diễn mã xích (Chaine Code).
- Biểu diễn mã tứ phân (Quad Tree Code).

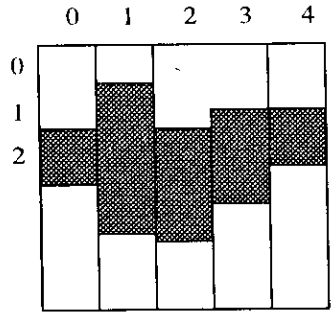
Ngoài ra cũng dùng mô hình thống kê .

2.3.1. Mã loạt dài

Phương pháp này hay dùng biểu diễn cho vùng ảnh hay ảnh nhị phân. Một vùng ảnh $\mathcal{R}$ có thể biểu diễn đơn giản nhờ một ma trận nhị phân:

$$u(m,n) = \begin{cases} 1 & \text{nếu } (m,n) \in \mathcal{R} \\ 0 & \text{nếu không} \end{cases}$$

Với cách biểu diễn trên, một vùng ảnh hay ảnh nhị phân được xem như gồm các chuỗi 0 hay 1 đan xen. Các chuỗi này gọi là một mạch (run). Theo phương pháp này, mỗi mạch sẽ được biểu diễn bởi địa chỉ bắt đầu của mạch và chiều dài mạch theo dạng: (<hàng,cột>, chiều dài).



Ảnh được biểu diễn

(1,1) 1, (1,3) 2

(2,0) 4, (3,1) 2

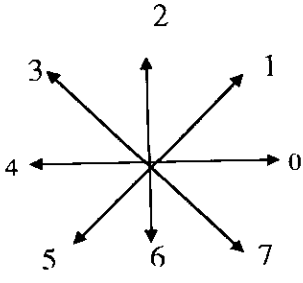
Hình 2.8. Ảnh nhị phân và các biểu diễn mã loạt dài tương ứng.

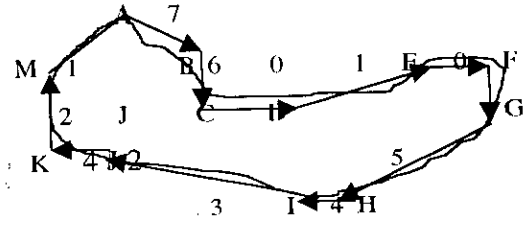
Nhiều dạng biến thể khác nhau của phương pháp này sẽ đề cập chi tiết trong chương 8 (8.2.1).

2.3.2. Mã xích

Mã xích thường được dùng để biểu diễn biên của ảnh. Thay vì lưu trữ toàn bộ ảnh, người ta lưu trữ dãy các điểm ảnh như A, B,..., M (hình 2.9). Theo phương pháp này, 8 hướng của véctơ nối 2 điểm biên liên tục được mã hoá. Khi đó ảnh được biểu diễn qua điểm ảnh bắt đầu A cùng với chuỗi các từ mã. Điều này được minh hoạ trong hình 2-9 dưới đây.

Một biến thể của phương pháp này là tăng số hướng. Với biên cũng còn nhiều phương pháp khác (Chương 5).





Hình 2.9. Hướng các điểm biên và mã tương ứng.

A 111 110 000 001 000 110 101 110 101 010 100 010

2.3.3. Mã tứ phân

Theo phương pháp mã tứ phân, một vùng của ảnh coi như bao kín bởi 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 đen và trắng gọi là vùng xám lại tiếp tục được chia làm 4 vùng con tiếp. Quá trình chia dừng lại khi không thể chia tiếp được nữa, có nghĩa là vùng con chỉ chứa thuần nhất điểm đen hay trắng. Như vậy, cây biểu diễn gồm một chuỗi các ký hiệu b(black), w (white) và g(grey) kèm theo ký hiệu mã hoá 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 khá khó.

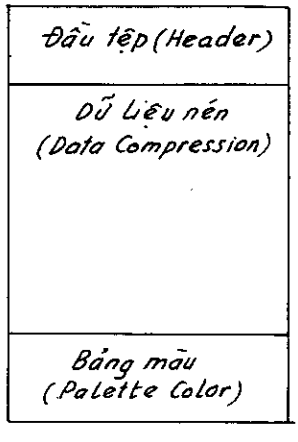
2.4. CÁC ĐỊNH DẠNG ẢNH CƠ BẢN TRONG XỬ LÝ ẢNH

2.4.1. Khái niệm chung

Ảnh thu được sau quá trình số hoá thường được lưu lại cho các quá trình xử lý tiếp theo hay truyền đi. Trong quá trình phát triển của kỹ thuật xử lý ảnh, tồn tại nhiều định dạng khác nhau từ ảnh đen trắng IMG, ảnh đa cấp xám/ảnh màu: BMP, GIF, JPEG, ... Tuy các định dạng này là khá khác nhau, song chúng cũng tuân theo một cấu trúc chung nhất. Để trong sáng cách trình bày, định dạng chi tiết của các File ảnh sẽ được mô tả trong phần phụ lục. Dưới đây chỉ trình bày cấu trúc chung và ý nghĩa của các phần cấu trúc đó.

Nhìn chung, một tệp ảnh gồm 3 phần:

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



a)Đầu tệp: Chứa các thông tin về kiểu ảnh, kích thước, độ phân giải, số bit dùng cho 1 pixel, cách mã hoá, vị trí bảng màu,.... Kích thước phần header rất khác nhau, phụ thuộc kiểu định dạng ảnh.

b)Dữ liệu nén: Số liệu ảnh đã được mã hoá bởi kiểu mã hoá chỉ ra trong phần Header.

c)Bảng màu: Không nhất thiết phải có. Nếu có, nó chỉ ra số màu dùng trong ảnh và sử dụng để hiện ảnh.

2.4.2. Qui trình đọc 1 tệp ảnh

Để xử lý được ảnh, ta 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 các byte. Để đọc đúng, ta cần hiểu ý nghĩa các phần trong cấu trúc của tệp ảnh như đã nêu trên. Trước tiên, ta cần đọc Header để lấy các thông tin điều khiển. Việc đọc này sẽ dừng ngay khi ta không gặp được chữ ký mong muốn (thường là 6 bytes đầu). Dựa vào thông tin điều khiển này, ta xác định được vị trí bảng màu (nếu có) và đọc nó vào bộ nhớ. Cuối cùng, ta đọc phần dữ liệu nén.

Sau khi đã đọc xong các khối vào bộ nhớ, ta tiến hành giải nén số liệu ảnh. Căn cứ vào phương pháp nén chỉ ra trong phần Header (có thể có các bảng phụ) 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 màn hình.

2.5. CÁC KỸ THUẬT TÁI HIỆN ẢNH

2.5.1. Kỹ thuật chụp ảnh (photography hardcopy techniques)

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

1) Đặt camera trong phòng tối, cách màn hình khoảng 10 feet (1 feet =0,3048 m).

2) Mở ống kính để làm phẳng mặt cong màn hình do vậy ảnh sẽ dàn đều hơn.

3) Tắt phím phản chiếu (brightness) và phím tương phản (constrast) của màn hình để tạo nên 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.

Với ống kính tốc độ 1/4 giây, bắt đầu với f và dừng ở f/8.

2.5.2. Kỹ thuật in ảnh (Printer Hardcopy techniques)

Trước khi đi sâu vào kỹ thuật in ảnh, ta xem xét các ảnh được thể hiện trên sách, báo ảnh và tạp chí như thế nào. Nhìn chung, người ta dùng kỹ thuật nửa cường độ (*halftone*). Theo kỹ thuật này, một ảnh tạo nên bởi một chuỗi các điểm in trên giấy. Thực chất, mỗi pixel gồm một hình vuông trắng bao quanh một chấm đen (black dot). Do vậy, nếu chấm đen càng lớn ảnh sẽ càng xẫm màu. Màu xám có thể coi như chấm đen chiếm

nửa vùng trắng. Vùng trắng là vùng gồm một chùm các các pixel gồm rất ít hoặc không có chấm đen.

Do sự 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 là 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. Thường báo ảnh tạo ảnh nửa cường độ với độ phân giải từ 60 đến 80 dpi, sách có thể in đến 150 dpi.

Tuy nhiên, các máy in của máy tính không có khả năng sắp xếp các chấm đen của các kích thước khác nhau của ảnh. Do đó, người ta phải dùng một số kỹ thuật biến đổi như: phân ngưỡng, chọn mẫu, dithering. Chúng ta lần lượt xem xét dưới đây.

- **Phân ngưỡng (Thresholding)**

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:

- Hiển thị 2 màu: dùng cho 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. Và như vậy:

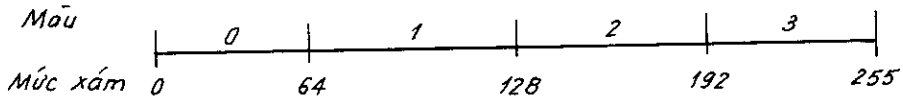
$$u(m,n) = \begin{cases} 1 \text{ cho hiện (đen) nếu } u(m,n) < 127 \\ 0 \text{ (hay hiện trắng) nếu ngược lại} \end{cases}$$

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

- Kỹ thuật hiện 4 màu: để khắc phục nhược điểm của cách hiện 2 màu, người ta qui định cách hiện 4 màu như sau:

Màu	Màn hình monochrome(đơn sắc)	Màn hình màu
0	đen	đen
1	xám đậm	đỏ
2	xám nhạt	xanh
3	trắng	vàng

Ta có thể hình dung cách phân ngưỡng 4 màu theo sơ đồ sau:



- Dùng ngưỡng ngẫu nhiên 4 màu: theo phương pháp này, ta chia không gian mức xám thành các miền hiển thị và đánh số là 0, 1, 2 và 3. Tiếp sau, định nghĩa các miền mà các cặp hiển thị (bật, tắt) tương ứng với: (0,1), (1,2) và (2,3). Khác với ngưỡng cố định, ở đây ngưỡng được reo ngẫu nhiên . Quá trình thực hiện được mô tả trong thuật toán :

```
for each pixel I(x,y) do
 Begin
     if I(x,y) < 85 then
     Begin
            . Khởi tạo 1 số ngẫu nhiên r trong khoảng [0,84]
            . if I(x,y) >r then màu =1 else màu =0
     End
     else if  I(x,y)<170 then
          Begin
             . Khởi tạo 1 số ngẫu nhiên r trong khoảng [85,169]
             . if I(x,y) >r then màu =1 else màu =0
          End
          Else
             Begin
              . Khởi tạo một số ngẫu nhiên r trong khoảng [170,255]
              . if I(x,y) >r then màu =1 else màu =0
             End
  end
```

- **Kỹ thuật chọn theo mẫu (patterning)**

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 × 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:

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 đi 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 640 x 480 khi sử dụng nhóm có kích thước 4 x 4 sẽ chỉ còn 160 x 120.

- **Kỹ thuật Dithering**

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.

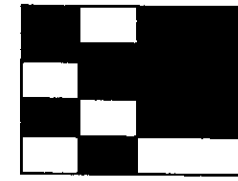
Để tạo ảnh, mỗi phần tử của ảnh gốc sẽ được so sánh với phần tử tương ứng của ma trận Dither. Nếu lớn hơn, phần tử ở đầu ra sẽ sáng và ngược lại. Ma trận Dither cấp 2n sẽ được tính như sau:

8	7	8	15
6	7	15	13
7	5	13	12
15	13	12	12

a) ảnh gốc

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

b) ma trận Dither - D^4



c) ảnh kết quả

Hình 2.11. Tạo ảnh theo phương pháp Dithering

$$D^{2n} = \begin{bmatrix} 4D^n + D^2_{00}U^n & 4D^n + D^2_{01}U^n \\ 4D^n + D^2_{10} & 4D^n + D^2_{11}U^n \end{bmatrix} \text{ với } D^2 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} D^2_{00} & D^2_{01} \\ D^2_{10} & D^2_{11} \end{bmatrix}$$

D^n là ma trận Dither cấp n ;U^n là ma trận cấp n (các phần tử đều là 1) dạng:

$$\begin{bmatrix} 1 & . & 1 \\ . & & . \\ 1 & . & 1 \end{bmatrix}$$

Thí dụ, với D^2 như trên, ta tính D^4 như sau:

$$D^4 = \begin{bmatrix} 4D^2 + D_{00}^2U^2 & 4D^2 + D_{01}^2U^2 \\ 4D^2 + D_{10}^2 & 4D^2 + D_{11}^2U^2 \end{bmatrix} \text{ và } D^4 = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

Một cách tương tự, ta tính được D^{16}. Với D^{16}, ta thấy tất cả các giá trị từ 0 đến 255 đều có mặt. Khác với phương pháp ngưỡng ngẫu nhiên chỉ dựa vào một ngưỡng biến thiên, ở đây ngưỡng được xác định một cách rõ ràng. Cách dùng ma trận ngưỡng có thể hình dung như sau:

$$\begin{bmatrix}
0 & 128 & 32 & 160 & 8 & 136 & 40 & 168 & 2 & 130 & 34 & 162 & 10 & 138 & 42 & 170 \\
192 & 64 & 224 & 96 & 200 & 72 & 232 & 104 & 194 & 66 & 226 & 98 & 202 & 74 & 234 & 106 \\
48 & 176 & 16 & 144 & 56 & 184 & 24 & 152 & 50 & 178 & 18 & 146 & 58 & 186 & 26 & 154 \\
240 & 112 & 208 & 80 & 248 & 120 & 216 & 88 & 242 & 114 & 210 & 82 & 250 & 122 & 218 & 90 \\
12 & 140 & 44 & 172 & 4 & 132 & 36 & 164 & 14 & 142 & 46 & 174 & 6 & 134 & 38 & 166 \\
204 & 76 & 236 & 108 & 196 & 68 & 228 & 100 & 206 & 78 & 238 & 110 & 198 & 70 & 230 & 102 \\
60 & 188 & 28 & 156 & 52 & 180 & 20 & 148 & 62 & 192 & 30 & 158 & 54 & 182 & 22 & 150 \\
252 & 124 & 220 & 92 & 244 & 116 & 212 & 84 & 254 & 126 & 222 & 94 & 246 & 118 & 214 & 86 \\
3 & 131 & 35 & 163 & 11 & 139 & 43 & 171 & 1 & 129 & 33 & 161 & 9 & 137 & 41 & 169 \\
195 & 67 & 227 & 99 & 203 & 75 & 235 & 107 & 193 & 65 & 225 & 97 & 201 & 73 & 233 & 105 \\
51 & 179 & 19 & 147 & 59 & 187 & 27 & 155 & 49 & 177 & 17 & 145 & 57 & 185 & 25 & 153 \\
243 & 115 & 211 & 83 & 251 & 123 & 219 & 91 & 241 & 113 & 209 & 81 & 249 & 121 & 217 & 89 \\
15 & 143 & 47 & 175 & 7 & 135 & 39 & 167 & 13 & 141 & 45 & 173 & 5 & 133 & 37 & 165 \\
207 & 79 & 239 & 111 & 199 & 71 & 231 & 103 & 205 & 77 & 237 & 109 & 197 & 69 & 229 & 101 \\
63 & 191 & 31 & 159 & 55 & 183 & 23 & 151 & 61 & 189 & 29 & 157 & 53 & 181 & 21 & 149 \\
255 & 127 & 223 & 95 & 247 & 119 & 215 & 87 & 253 & 125 & 221 & 93 & 245 & 117 & 213 & 85
\end{bmatrix}$$

Ma trận Dither cấp 16 - D^{16}

Giả sử I(x,y) là một điểm ảnh. Đặt x_0 = x mod 16 và y_0 = y mod 16 thì x_0,y_0 sẽ chỉ nhận một trong các giá trị từ 0 đến 15. Như vậy, nó sẽ xác định một phần tử của D^{16}. Gọi S = $D^{16}(x_0,y_0)$. S sẽ có giá trị trong khoảng từ 0 đến 255 và có thể dùng làm ngưỡng để hiện ảnh. Hơn nữa, nếu chỉ dịch chuyển theo một chiều (x chẳng hạn), ta thấy rằng cả 16 điểm ảnh sẽ rơi vào ngưỡng S thu được từ 16 điểm trước. Do vậy, ta có một cách thức chuẩn để bật hay tắt các điểm ảnh với trạng thái lưới.

Để sử dụng được 4 màu, ta cũng sử dụng theo kỹ thuật ngưỡng 4 màu nhưng chỉ cần 3 vùng, mỗi vùng 85 mức. Ở đây cần có sự lựa chọn giữa D^{16} và D^{8}. Nếu chọn D^{16} thì sẽ thừa, nên ta chọn D8 với chuẩn hoá theo cách thức chỉ dùng 63 mức giá trị.

vùng 1 $$p = \frac{I(x,y)*63}{84} = \frac{3*I(x,y)}{4}$$

vùng 2 $$p = \frac{(I(x,y)-85)*3}{4}$$

vùng 3 $$p = \frac{(I(x,y)-170)*3}{4}$$

Thuật toán phân ngưỡng dùng ma trận ngưỡng được mô tả như sau:

```
for each I[i,j] do
 if I[i,j] < 84 then { vùng 1}
 Begin  I[i,j] := I[i,j] * 3/4
        x0 := (i mod 8) +1; y0 := (j mod 8)+1;
        nguong := D[x0,y0];
        If  I[i,j] < nguong then  Hien(j+x+1, i+y+1,0)
        Else Hien(j+x+1, i+y+1,0)
end
Else  If I[i,j] < 169 then {vùng 2}
          Begin  I[i,j] := (I[i,j] * 3-85)/4
                 x0 := (i mod 8) +1; y0 := (j mod 8)+1;
                 nguong := D[x0,y0];
                 If  I[i,j] < nguong then  Hien(j+x+1, i+y+1,1)
                 Else Hien(j+x+1, i+y+1,2)
          end
      Else {vùng 3}
          Begin  I(i,j) := (I[i,j] * 3-170)/4
                 x0 := (i mod 8) +1; y0 := (j mod 8)+1;
                 nguong := D[x0,y0];
```

```
                If I[i,j] < nguong then  Hien(j+x+1, i+y+1,2)
                Else Hien(j+x+1, i+y+1,3)
            end
        End
```

2.6. KHÁI NIỆM ẢNH ĐEN TRẮNG, ĐA CẤP XÁM, ẢNH MÀU

Có xu hướng hay nhầm lẫn giữa các khái niệm ảnh nhị phân, ảnh đen trắng cũng như ảnh đơn sắc (một màu — monochrome) với ảnh màu. Thực tế, ảnh đen trắng bao gồm ảnh nhị phân và ảnh đa cấp xám. Khái niệm cấp xám hay mức xám (grey level) đã đề cập trong phần 1.2.1 chương 1. Dưới đây sẽ làm rõ thêm khái niệm này cũng như cách biểu diễn và lưu trữ các loại ảnh này.

2.6.1. Ảnh đen trắng

Ảnh đen trắng chỉ bao gồm 2 màu: màu đen và màu trắng. Người ta phân sự biến đổi đó thành L mức. Nếu L bằng 2, nghĩa là chỉ có 2 mức: mức 0 và mức 1 và 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. Việc xác định số mức là phụ thuộc vào tiêu chí lượng hoá. L thường chọn 32, 64, 128 và 256. Ảnh 256 mức là ảnh có chất lượng cao và thường được sử dụng.

Với ảnh nhị phân, mỗi pixel mã hoá trên 1 bit; còn với ảnh 256 mức, mỗi pixel mã hoá trên 8 bit. Ví dụ với một ảnh 256 mức xám, kích thước 512 x 512 cần không gian lưu trữ là 512 x 512 bytes hay 256 Kbytes.

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

2.6.2. Ảnh màu

Ảnh màu là ảnh tổ hợp từ 3 màu cơ bản: đỏ (R), lục (Green) và lơ (Blue) và thường thu nhận trên các dải băng tần khác nhau. Mỗi pixel ảnh màu gồm 3 thành phần như nêu trong phần 2.1.4. Mỗi màu cũng phân thành L cấp khác nhau (L thường là 256). Do vậy, để lưu trữ ảnh màu, người ta có thể lưu trữ từng mặt 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 thước.

Bài tập chương 2

1. Viết thủ tục hiện ảnh theo kỹ thuật 4 màu với ngưỡng ngẫu nhiên.

2. Với ma trận Dither cấp 4 - D^4 đã cho trong giáo trình, hãy viết chương trình tính ma trận D^8 và D^{16}.

3. Với ma trận D^8 vừa tính được, hãy áp dụng kỹ thuật Dithering để biến đổi ảnh ra và so sánh kết quả. Ảnh vào có thể tự tạo hay sử dụng ảnh đã tạo ra ở chương I.

4. Viết thủ tục thể hiện chức năng in ảnh từ màn hình ra máy in:

a) đọc một ảnh PCX hay BMP và hiện lên màn hình;

b) dùng kỹ thuật Dither để in ảnh .

5. Viết thủ tục đọc một ảnh PCX (giải nén) và lưu vào bảng 2 chiều.

6. Viết thủ tục lưu ảnh số biểu diễn bởi bảng lên tệp PCX.

7. Xây dựng giải thuật chuyển đổi định dạng PCX sang BMP và ngược lại.

8. Đọc và hiện ảnh với định dạng TIFF.

9. Đọc và hiện ảnh với định dạng GIF.

10. Một ảnh đa cấp xám kích thước 1024 x 1024, mỗi pixel mã hoá trên 12 bit.

a) Số mức lượng hoá tối đa của ảnh là bao nhiêu?

b) Không gian nhớ dùng lưu trữ cho ảnh ?

c) Nếu ảnh trên là ảnh màu thì ảnh đó có tối đa bao nhiêu màu?

3

CÔNG CỤ TRỢ GIÚP XỬ LÝ ẢNH SỐ

TOOLS FOR IMAGE PROCESSING

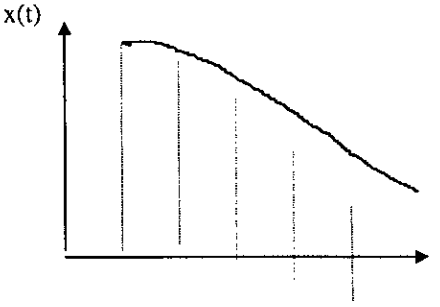
Thuật ngữ " xử lý ảnh số" thường dùng để chỉ các quá trình xử lý ảnh 2 chiều bằng máy tính. Ảnh số thường được biểu diễn bởi ma trận 2 chiều các số thực hay số phức gồm một số hữu hạn các bit. Để có thể xử lý được trên máy tính, ảnh đã cho (ảnh, giấy phim hay đồ thị) đầu tiên phải được số hoá (digitalized) và lưu dưới dạng ma trận 2 chiều các bit. Trong chương này chúng ta sẽ đề cập tới các công cụ và các kỹ thuật sử dụng trong xử lý ảnh số. Trước tiên là giới thiệu tổng quan về xử lý ảnh số (tín hiệu trong không gian). Tiếp theo, giới thiệu một số khái niệm như : toán tử tuyến tính, tích chập (convolution product) và lọc số (filtering) - các công cụ cơ bản và ứng dụng của chúng trong xử lý ảnh. Kế đó trình bày về một số biến đổi hay dùng như biến đổi Fourier, biến đổi Karhumen Loeve. Các công cụ xử lý điểm ảnh được trình bày chi tiết về nguyên tắc cũng như công cụ lược đồ xám (histogram) và các phép biến đổi lược đồ. Cuối cùng là một số kỹ thuật khác trong mô hình thống kê.

3.1. TỔNG QUAN VỀ XỬ LÝ ẢNH TRONG KHÔNG GIAN

3.1.1. Tín hiệu số và biểu diễn ảnh số

Như đã nêu trong chương 1, một hàm hai biến thực hoặc phức có thể coi như một ảnh. Một ảnh trong không gian 2 chiều có thể biểu diễn bởi một tập hợp các ma trận cơ sở gọi là ảnh cơ sở. Như vậy một tín hiệu 2 chiều liên tục trong không gian, theo khái niệm trên gọi là ảnh liên tục trong không gian số thực và ký hiệu là f(x,y): giá trị của f(x,y) là liên tục trong khoảng $(-\infty,\infty)$.

Các tín hiệu liên tục theo thời gian qua quá trình số hoá ta thu được tín hiệu rời rạc (tín hiệu số).



Hình 3.1. Tín hiệu số rời rạc.

Ảnh số chính là ảnh xử lý bằng máy tính thu được từ ảnh liên tục bởi quá trình số hoá (lấy mẫu và lượng hoá), thường được ký hiệu là I[m,n]. Giá trị I[x,y] biểu diễn cường độ sáng được mã hoá của mỗi điểm ảnh (x,y). Giá trị đó còn gọi là mức xám (grey level). Vậy I[x,y] có giá trị rời rạc và để tiện xử lý, ta coi giá trị của I[x,y] là nguyên: I[x,y] ∈ {0, 1, ..., L-1} với L là mức xám tối đa dùng để biểu diễn.

Để giảm độ phức tạp tính toán, các giá trị của (m,n) thường chọn là hữu hạn và thường chọn là 512; còn L chọn là 256. Ảnh có nhiều mức xám gọi là ảnh đa cấp xám. Ảnh chỉ có 2 mức xám 0 và 1 gọi là ảnh nhị phân.

Với cách biểu diễn trên, ảnh số chính là một một phần của tín hiệu số trong không gian 2 chiều. Và cách biểu diễn ảnh số thông dụng nhất là dùng bảng 2 chiều mà thuật ngữ thường gọi là ma trận ảnh hay bản đồ ảnh.

3.1.2. Khái quát về hệ thống xử lý tín hiệu số

Hệ thống số là một hệ thống tiếp nhận tín hiệu số ở đầu vào, xử lý tín hiệu theo một qui trình nào đấy và đưa ra cũng là một tín hiệu số. Vì ảnh số là một phần của tín hiệu số, nên hệ thống xử lý ảnh số có đặc thù như hệ thống số cộng thêm một số tính chất riêng.

Nếu gọi tín hiệu số đầu vào là X(m,n), tín hiệu số đầu ra là Y(m,n), đặc trưng của hệ thống là H, ta có thể biểu diễn hệ thống số một cách hình thức như sau:

$$Y(m,n) = H\,[X(m,n)]$$

Phần lớn các các hệ thống số là tuyến tính và bất biến. Khái niệm tuyến tính và bất biến sẽ trình bày trong phần 3.2. Trong xử lý tín hiệu số, thường có 2 cách tiếp cận khác nhau:

- Biên độ của tín hiệu được lấy mẫu, lượng hoá theo một qui chuẩn và có thể biểu diễn bởi một hàm liên tục theo thời gian. Đây là cách tiếp cận theo không gian thực.

- Cách tiếp cận thứ hai là tiếp cận theo miền tần số của tín hiệu. Trong cách tiếp cận này, trước tiên tín hiệu được biến đổi chẳng hạn như phép biến đổi Fourrier, sau đó, tiến hành xử lý trên miền tần số. Cuối cùng dùng biến đổi ngược để đưa tín hiệu đã xử lý về miền số thực.

Thí dụ như tín hiệu thu nhận là tiếng còi ô tô. Ta có thể tiếp cận theo 2 cách khác nhau:

- Lấy mẫu biên độ tín hiệu nhiều lần trong một chu kỳ và được một xấp xỉ của tín hiệu là một hàm liên tục theo thời gian.

- Phân tích tín hiệu theo độ cao của âm thanh hay tần số của âm thanh và lưu trữ biên độ của mỗi tần số.

Hai cách tiếp cận trên cho ta 2 kỹ thuật cơ bản được dùng trong xử lý ảnh (đề cập trong các phần sau):

-Tác động trực tiếp lên điểm ảnh: tích chập, lọc số và các toán tử điểm.

- Biểu diễn ảnh sang một không gian khác bằng các biến đổi, xử lý và biến đổi ngược lại.

3.2. CÁC TOÁN TỬ KHÔNG GIAN (SPATIAL OPERATORS)

Các toán tử không gian (KG) thường dùng là các toán tử tuyến tính, tích chập và lọc. Mục đích chính của các toán tử này là làm cho ảnh "tốt hơn" và thuận tiện cho việc biến đổi và xử lý ảnh về sau như: tăng cường và nâng cao chất lượng ảnh, dò biên, trích chọn đặc tính v.v.

a) Toán tử tuyến tính

Phần lớn các hệ thống xử lý ảnh có thể mô hình hoá như một hệ thống tuyến tính hai chiều. Giả sử x(m,n) và y(m,n) biểu diễn các tín hiệu vào và ra tương ứng của hệ thống. Hệ thống hai chiều được biểu diễn bởi:

$$y(m,n) = H[x(m,n)] \tag{3.1}$$

Hệ thống này gọi là tuyến tính *khi và chỉ khi:* tổ hợp tuyến tính của 2 tín hiệu vào $x_1(m,n)$, $x_2(m,n)$ cũng tạo nên chính tổ hợp tuyến tính tương ứng của đầu ra $y_1(m,n)$, $y_2(m,n)$, nghĩa là: với 2 hằng số bất kì α và ß, ta có:

$$\begin{aligned} H[\alpha x_1(m,n) + \beta x_2(m,n)] &= \alpha H[x_1(m,n)] + \beta H[x_2(m,n)] \\ &= [\alpha y_1(m,n)] + [\beta y_2(m,n)] \end{aligned} \tag{3.2}$$

Phương trình 3.2 gọi là *chồng tuyến tính* của 2 tín hiệu. Ý nghĩa quan trọng của hệ tuyến tính là: khi có nhiều tín hiệu vào, hệ thống có thể xử lý độc lập từng tín hiệu sau đó tổ hợp kết quả lại

Khi tín hiệu vào là hàm đenta Kronecker 2 chiều δ (xung đơn vị) tại vị trí (m',n'), tín hiệu ra ở vị trí (m,n) được định nghĩa:

$$h(m,n\ ;\ m',n') = H[\delta(m-m';\ n-n')] \tag{3.3}$$

Dấu ";" trong các công thức trên để phân biệt toạ độ vào và toạ độ ra.

Hàm đenta δ(m,n) có dạng:

$$\delta(m,n) = \begin{cases} 1 & \text{nếu } m = n \\ 0 & \text{nếu } m \neq n \end{cases}$$

b) Tích chập

Trước khi đề cập đến khái niệm này, ta xét một khái niệm có liên quan, đó là khái niệm bất biến trượt (shift invariance). Một hệ thống gọi là bất biến trượt nếu dịch chuyển đầu vào thì cũng tạo nên một dịch chuyển tương ứng của đầu ra. Theo phương trình 3.3, nếu xung xảy ra ở gốc toạ độ, ta có:

$$H[\delta(m-n)] = h[m,n\ ;\ 0,0] \tag{3.4}$$

$$\Rightarrow \quad h(m,n\ ;\ m',n') = h(m-m'\ ;\ n-n') \tag{3.5}$$

Theo định nghĩa này, tín hiệu ra có dạng:

$$y(m,n) = \sum_{m,n=-\infty}^{\infty} h(m-m';n-n')x(m',n') \tag{3.6}$$

Phương trình 3.6 gọi là chập của đầu vào x(m',n') với đáp ứng xung (impulse response) h(m,n).

Hình 3.2 minh hoạ toán tử chập. Ma trận đáp ứng xung quay quanh gốc 180° và trượt một khoảng (m,n) rồi chồng lên ma trận tín hiệu vào x(m',n').

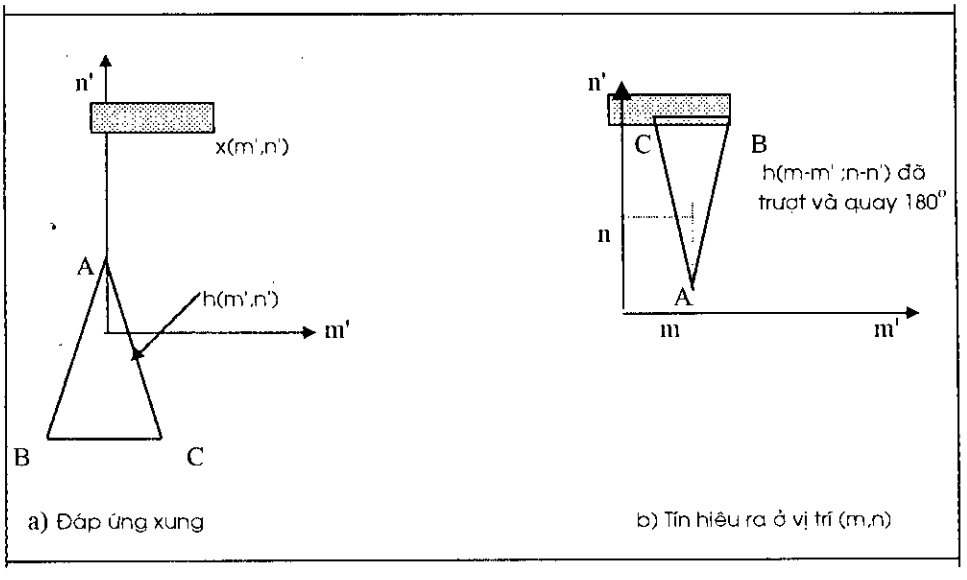
Toán tử tích chập được định nghĩa như sau:

+ trường hợp liên tục

$$g(x,y) = h(x,y) \otimes f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x-x',y-y')f(x',y')dx'dy' \tag{3.7}$$

+ trường hợp rời rạc

$$y(m,n) = h(m,n) \otimes x(m,n) = \sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty} h(m-m',n-n')x(m',n') \quad (3.8)$$



Hình 3.2. Một biểu diễn của toán tử chập.

Để tiện theo dõi, ta xét ví dụ sau:

- ma trận tín hiệu $x_{2\times 3}$

$$\begin{bmatrix} 1 & 4 & 1 \\ 2 & 5 & 3 \end{bmatrix}$$

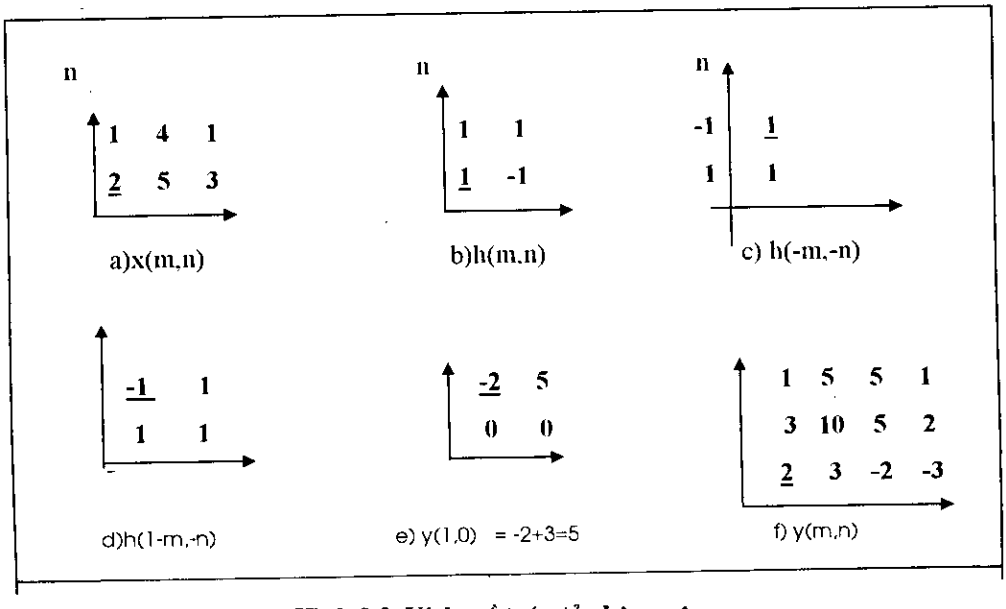
- ma trận đáp ứng xung $h_{2\times 2}$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Ma trận thu được bởi tích chập của 2 ma trận h và x là một ma trận 4 x 3. Nói chung, chập của 2 ma trận số (M_1 x N_1) và (M_2 x N_2) là một ma trận cỡ ($M_1 + M_2 - 1$, $N_1 + N_2 - 1$). Hình 3.3 dưới đây mô tả các bước thực hiện chập của 2 ma trận h và x ở trên. Các số gạch dưới là điểm bắt đầu thực hiện qua mỗi bước.

Theo công thức 3.8 , tích chập $H \otimes X$ có độ phức tạp tính toán rất cao. Để giảm độ phức tạp tính toán người ta thường dùng nhân chập H_{KxL} có kích thước hữu hạn và nhỏ: Nhân chập này thường chọn có kích thước lẻ và các giá trị hay dùng là: K = L = 3, 5, 7. Trong các phần sau, ta thấy đa số các nhân chập được sử dụng trong tích chập, lọc số là

nhân chập vuông, đôi khi là nhân chập chữ thập. Thực ra nhân chập chữ thập là nhân chập vuông, song một số phần tử của nó có giá trị 0 nên ta coi như không có.



Hình 3.3. Ví dụ về toán tử chập cuộn.

Với cách chọn nhân chập như trên, hai công thức tính nhân chập sau đây thường được sử dụng:

- Xếp chồng tại biên

$$Y(m,n) = \sum_{k=0}^{L-1}\sum_{l=0}^{L-1} H(k,l)^* X(m-k,n-l) \tag{3.9}$$

Theo công thức này, nếu K=L=3, nhân chập H có thể viết:

$$H(k.l) = \begin{bmatrix} H_{00} & H_{01} & H_{02} \\ H_{10} & H_{11} & H_{12} \\ H_{20} & H_{21} & H_{22} \end{bmatrix}$$

Với nhân chập H như trên, công thức (3.9) được viết lại một cách tường minh như sau:

$$Y(m,n) = H_{00}X_{m,n} + H_{01}X_{m,n-1} + H_{02}X_{m,n-2} + H_{10}X_{m-1,n} + H_{11}X_{m-1,n-1} + H_{12}X_{m-1,n-2} +$$
$$+ H_{20}X_{m-2,n} + H_{21}X_{m-2,n-1} + H_{22}X_{m-2,n-2}.$$

với $m \in [1, M]$ và $n \in [1, N]$. Các chỉ số được viết theo ký pháp toán cho dễ quan sát.

Như vậy, việc tính Y(m,n) như được thực hiện bởi xoay nhân chập H 90° rồi xếp chồng với lân cận của điểm (m,n) sao cho H_{00} trùng với điểm lấy ra. Điều này lý giải cho tên gọi của phương pháp "*xếp chồng tại biên*".

- *Xếp chồng tại trung tâm*

$$Y(m,n) = \sum_{k=1}^{L}\sum_{l=1}^{L} H(k,l)* X(m-k+L_c,n-l+L_c) \text{ với } L_c = \frac{L+1}{2} \qquad (3.10)$$

Theo công thức này, nếu K=L=3, nhân chập H có thể viết:

$$H(k,l) = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

Và công thức (3.10) được viết lại một cách tường minh như sau:

$$Y(m,n) = H_{11}X_{m+1,n+1} + H_{12}X_{m+1,n} + H_{23}X_{m+1,n+1} + H_{21}X_{m,n+1} + H_{22}X_{m,n} + H_{23}X_{m,n-1} + H_{31}X_{m-1,n+1} + H_{32}X_{m-1,n} + H_{33}X_{m-1,n-1}.$$

với $m \in [1, M]$ và $n \in [1, N]$. Các chỉ số được viết theo ký pháp toán cho dễ quan sát.

Như vậy, việc tính Y(m,n) như được thực hiện bởi xoay nhân chập H 90° rồi xếp chồng với lân cận của điểm (m,n) sao cho H_{22} trùng với điểm lấy ra (điểm lấy ra trùng với tâm nhân chập). Điều này cũng lý giải cho tên gọi của phương pháp "*xếp chồng tại trung tâm*".

Thực tế, công thức này có thể áp dụng cho cả 2 trường hợp. Nếu áp dụng để tính cho điểm ở biên, ta coi các điểm ngoài biên có giá trị 0. Thí dụ, cho ảnh số I sau:

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

và nhân chập H:

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

tích chập H ⊗ I tính theo công thức 3.10 được:

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

Để hiểu rõ cách tính nhân chập, ta tính một vài phần tử. Các phần tử còn lại coi như một bài tập nhỏ tự làm.

$$Y(1,1) = \frac{1}{9}(1*7 + 1*5 + 1*7 + 1*4) = \frac{1}{9}*23$$

$$Y(4,2) = \frac{1}{9}(1*6 + 1*7 + 1*5 + 1*5 + 1*7 + 1*5 + 1*1 + 1*6 + 1*6) = \frac{1}{9}*48$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 \\ 0 & \boxed{1} & 7 \\ 0 & 5 & 7 \end{bmatrix} \qquad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 6 & 6 & 1 \\ 5 & \boxed{1} & 5 \\ 5 & 7 & 6 \end{bmatrix}$$

nhân chập các lân cận nhân chập các lân cận

Vị trí điểm ảnh đầu ra là điểm được in đậm

a) Tính Y(1,1) **b) Tính Y(4,2)**

Tích chập là một khái niệm rất quan trọng trong xử lý ảnh, đặc biệt là tính chất của nó có liên quan đến biến đổi Fourier: biến đổi Fourier của một tích chập bằng tích đơn giản các biến đổi Fourier của các tín hiệu đó:

$$F[H(x,y) \otimes I(x,y)] = F[H(x,y)].\ F[I(x,y)] \tag{3.11}$$

Trong kỹ thuật, người ta gọi H là nhân chập hay nhân cuộn và cũng còn gọi là mặt nạ (mask); I [x,y] trong công thức trên là ảnh đối tượng.

Dưới đây, đưa ra một thuật toán tổng quát để tính nhân chập dùng cho mọi trường hợp. Để sử dụng thuật toán này chỉ cần thay đổi 2 thông số: ma trận biểu diễn ảnh số cần xử lý và ma trận biểu diễn nhân chập. Thuật toán được mô phỏng dưới dạng Pascal:

NhanChap(ImagIn,ImagOut: ảnh;H: Nhân chập;N:kích thước ảnh;w:kích thước nhân chập)

/* Vào: ImagIn

```
                    Nhân chập H
            Ra:  ImagOut      */
Begin
      For i:=1 to N do
      For j:=1 to N do
         Begin   Sum :=0; Lc:=(w+1) div 2;
For k:=1 to w do
                  For l:=1 to w do
                     Begin   Col:=i-k+Lc;Row:=j+l+Lc
                             If (Col<>0)and (Col <=N) then
                              If (Row<>0)and (Row <=N) then
                                  Sum:= Sum + ImagIn[Col,Row] * H[k,l];
                     End;
                  ImagOut[i,j]:=Sum
         End;
End;
```

c) Kỹ thuật lọc số

Trong nhiều lĩnh vực kỹ thuật, nhiễu đóng vai trò chủ yếu gây nên những khó khăn khi ta cần phân tích một tín hiệu nào đó, cũng không loại trừ tín hiệu ảnh. Giữa một ảnh thực và ảnh số hoá thu nhận được khác nhau khá nhiều vì có nhiều quá trình can thiệp vào. Nguyên nhân là do nhiễu điện tử của máy thu hay chất lượng kém của bộ số hoá. Ta xem xét để biết nhiễu thể hiện trên ảnh thế nào. Giả sử ảnh là một miền có mức xám đồng nhất. Như vậy các phần tử của ma trận biểu diễn ảnh sau quá trình số hoá phải có cùng giá trị. Nhưng thực tế quan sát, ta thấy: gần giá trị trung bình của mức xám có những phần tử trội lên khá nhiều. Đó chính là hiện tượng nhiễu. Như vậy, nhiễu trong ảnh số được xem như sự dịch chuyển nhanh của tín hiệu thu nhận (tín hiệu ảnh I[m,n]) trên một khoảng cách ngắn. Xem xét một cách tương đương trong không gian tần số, nhiễu ứng với các thành phần tần số cao trong ảnh. Do vậy, người ta nghĩ đến việc biến đổi có tính đến ảnh hưởng của các phần tử lân cận bằng cách lấy "tổ hợp " các điểm lân cận này (trong không gian thực) hay lọc các thành phần tần số cao (trong không gian tần số). Đây chính là *kỹ thuật lọc*

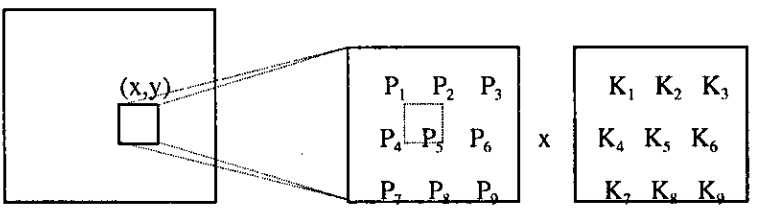
(filtering). *Cơ sở lý thuyết* của kỹ thuật lọc số là dựa trên tính dư thừa thông tin không gian: các pixel lân cận có thể có cùng hoặc gần cùng một số đặc tính. Hơn nữa, nhiễu có thể coi như sự đột biến của một điểm ảnh so với các điểm lân cận.

Trong kỹ thuật này, người ta sử dụng một mặt nạ và di chuyển khắp ảnh gốc. Tuỳ theo cách tổ hợp điểm đang xét với các điểm lân cận mà ta có kỹ thuật lọc tuyến tính hay phi tuyến. Điểm ảnh chịu tác động của biến đổi là điểm ở tâm mặt nạ.

- **Lọc tuyến tính**

Trong kỹ thuật lọc tuyến tính, ảnh thu được sẽ là tổng trọng số hay là trung bình trọng số các điểm lân cận với nhân cuộn hay mặt nạ. Nguyên tắc lọc theo tổng trọng số được minh hoạ qua hình 3.4. Thí dụ tâm mặt nạ là điểm P_5, thì điểm P_5 mới sẽ được tính theo công thức sau:

$$P_5 = P_1K_1 + P_2K_2 + P_3K_3 + P_4K_4 + P_5K_5 + P_6K_6 + P_7K_7 + P_8K_8 + P_9K_9$$



8 lân cận của P_5 Nhân cuộn 3 * 3

Hình 3.4. Lấy tổ hợp các điểm ảnh lân cận.

Nói chung, người ta sử dụng nhiều kiểu mặt nạ khác nhau:

$$H_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_2 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad H_3 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Mặt nạ H_1 là mặt nạ dùng để tính trung bình không trọng số (không ưu tiên theo hướng nào). Mặt nạ H_2 cho trọng số lớn nhất với điểm ở tâm. Còn mặt nạ H_3 ưu tiên cho 2 hướng x, y.

Giả sử I_i là ảnh đang xét, I_f là ảnh thu được và cả 2 ảnh đều có cùng kích thước p x p. Với mặt nạ trên, mỗi điểm ảnh thu được $I_f(x,y)$ sẽ được tính bởi:

$$I_f = \frac{1}{9}\{ I_i(x-1,y-1) + I_i(x-1,y) + I_i(x-1,y+1) + I_i(x,y-1) + I_i(x,y) + I_i(x,y+1)$$

$$+ I_i(x+1,y-1) + I_i(x,y) + I_i(x+1,y+1) \}$$

$$= \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} H_1(i+1,j+1)\, I_i(x+i,y+j) \qquad (3.12)$$

Nếu H là bộ lọc kích thước (n+1) x (n+1), n chẵn và tổng các hệ số là K, I_f sẽ được tính bởi:

$$I_f = \frac{1}{K}\sum_{i=-n/2}^{n/2}\sum_{j=-n/2}^{n/2} H_1(i+n/2,j+n/2)\, I_i(x+i,y+j) \qquad (3.13)$$

Công thức trên chính là tích chập giữa mặt nạ H và ảnh gốc I: $I_f = H \otimes I_i$. Trên thực tế, kỹ thuật lọc tuyến tính được xây dựng trên nền phép nhân chập. Phụ thuộc vào đầu ra mà chọn các nhân chập có ý nghĩa thích hợp (xem chương 4 và chương 5).

Chú ý rằng vừa rồi ta chưa xét đến biên của ảnh khi sử dụng kỹ thuật lọc. Giả sử ta áp mặt nạ H vào điểm tại gốc toạ độ (0,0), rõ ràng là điều này không thể được. Do vậy, chỉ có thể hoặc lọc phần trong của ảnh từ n/2 đến p-n/2 và trong trường hợp này ta thu được ảnh cỡ (p+1-n) x (p+1-n) hoặc là tạo thêm một nửa cỡ n/2 bằng cách sao.

Ngoài các bộ lọc trên, người ta cũng hay dùng bộ lọc Gauss. Bộ lọc này có ưu điểm là dễ cài đặt và cho chất lượng cao. Bộ lọc Gauss gồm tích chập của một ảnh I_f với mặt nạ Gauss $G(x,y,\sigma)$: $I_f = G \otimes I_i$ với

$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

G là mặt nạ hình vuông mà các hệ số của nó là các phần tử rời rạc của phân bố Gauss. Vì mặt nạ có kích thước (n+1) x (n+1) hữu hạn, còn đường cong G định nghĩa trên toàn miền thực, do vậy ta cần chọn một khoảng hữu hạn. Thường người ta chọn khoảng là 4σ (95%) hay 6σ (99.9%).

Người ta cũng chứng minh được rằng với mặt nạ N x N cần N^2 phép nhân và N^2-1 phép cộng. Các phương pháp lọc nói trên, nhìn chung làm giảm mức nhiễu trắng đi N_w lần, với N_w là số phần tử của mặt nạ và hạn chế nhoè sau khi lọc.

- **Lọc phi tuyến**

Khác với lọc tuyến tính, kỹ thuật lọc phi tuyến coi một điểm ảnh kết quả không phải

là tổ hợp tuyến tính của các điểm lân cận. Bộ lọc phi tuyến thường dùng là lọc trung vị (median filtering) mang tên Tuckey.

Trong trường hợp một chiều, trung vị x_a của một chuỗi n phần tử $\{x_n\}$ được định nghĩa:

- Nếu n lẻ: có (n-1)/2 phần tử lớn hơn x_a và (n-1)/2 nhỏ hơn hay bằng x_a.

- Nếu n chẵn: x_a là trung bình cộng của 2 phần tử x_i và $x_j \ni \{x_n\}$ sao cho có (n-2)/2 phần tử nhỏ hơn hay bằng x_i và (n-2)/2 phần tử lớn hơn hay bằng x_j.

Kỹ thuật lọc trung vị được dùng để lọc nhiễu bằng cách trượt trên mặt phẳng ảnh, mỗi lần trượt di chuyển một cột điểm. Những phần tử trong cửa sổ được xem như là 1 chuỗi $\{x_n\}$ và điểm quan tâm được thay thế bởi giá trị x_a của chuỗi. Thí dụ như chuỗi {1,2,9,5,4}, điểm trung tâm sẽ được thay thế bởi giá trị 4 được tính theo nguyên tắc ở trên. Rõ ràng trong ví dụ này, gía trị 9 có thể là nhiễu nhọn trong dãy tăng dần.

Lọc trung vị thường sử dụng cửa sổ kích thước 3. Tuy nhiên, nếu không có dấu hiệu quan trọng nào bị mất, kích thước cửa sổ có thể tăng lên 5, 7, v.v. và sẽ kết thúc khi quá trình lọc không làm thay đổi kết quả.

Khái niệm lọc trung vị dễ dàng mở rộng cho trường hợp hai chiều. Giả sử đầu vào là X(m,n) và đầu ra bộ lọc là Y(m,n). Lọc trung vị hai chiều được định nghĩa:

$$Y(m,n) = \text{Median}(X(m-k,n-l), \text{ với } k,l \in [1, L]$$

Lưu ý rằng công thức $L_c = (L+1)/2$ còn gọi là bán kính bộ lọc. Do vậy, ta có cách viết khác tương đương $(k,l) \in (-r,r)$ với $2r + 1 = L$.

Khi đó trung vị của cửa sổ vuông n x n có thể được tính như những phần tử của chuỗi một chiều. Ta tiến hành sắp xếp dãy đó rồi thay thế phần tử tâm cửa sổ bằng trung vị của dãy vừa tìm được.

Thuật toán được minh hoạ như sau:

Giả sử ta dùng nhân chập 3 x3 và các phần tử trong cửa sổ có dạng: n

Điểm xét X(m,n) = 78 (nhiễu)

Dãy lấy ra và sắp lại ta có:

15	15	16	17	~~17~~	17	18	20	78	
1	2	3	4	5	6	7	8	9	

m

	15	17	18	
	16	~~78~~	17	
	17	15	20	

Trung vị của dãy là phần tử ở vị trí số 5 và có giá trị là 17.

Giá trị mới này được thay cho phần tử tại tâm (78).

Như vậy là nhiễu đã bị khử.

Với cách thức như vậy, ta lần lượt rê cửa sổ lọc đi khắp ảnh và tiến hành lọc. Lưu ý rằng các ảnh mới phải lưu trữ khác với ảnh gốc.

Với lọc trung vị, số lượng tính toán khá lớn (có thể bằng số mũ của kích thước cửa sổ lọc). Vì vậy, để khắc phục nhược điểm này, người ta dùng một phương pháp khác: lọc giả trung vị (Pseudo-Median Filter). Thí dụ với dãy 5 số: a, b, c, d, e, lọc giả trung vị được định nghĩa như sau:

$$\text{PseudoMedian}(a,b,c,d,e) = \frac{1}{2}\left\{\begin{matrix}\text{MAX}[\text{Min}(a,b,c),\text{Min}(b,c,d),\text{Min}(c,d,e)] \\ +\text{MIN}[\text{Max}(a,b,c),\text{Max}(b,c,d),\text{Max}(c,d,e)]\end{matrix}\right\}$$

Rõ ràng là với phương pháp này, ta chỉ phải dùng 3 chuỗi con thay vì dùng 10 chuỗi như lọc trung vị.

Một cách tổng quát, ta có thuật toán sau:

b1. Lấy các phần tử trong cửa sổ ra mảng một chiều (L phần tử).

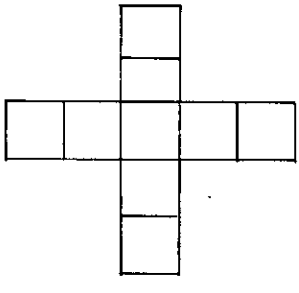
b2. Tìm min của lần lượt các chuỗi con rồi lấy max: gọi m1 là giá trị này.

b3. Tìm max của lần lượt các chuỗi con rồi lấy min: gọi m2 là giá trị tìm được.

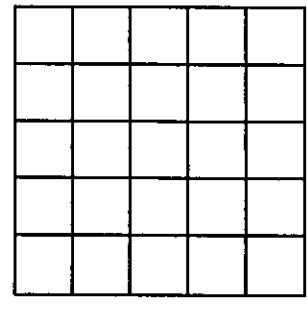
b4. Gán giá trị điểm đang xét là trung bình cộng của m1 và m2.

Lọc giả trung vị có nhiều điểm giống như lọc trung vị. Dãy lấy ra không cần sắp xếp và giá trị gọi là trung vị lại được tính theo trung bình cộng Max của min và Min của max.

Hai loại mặt nạ hay dùng là mặt nạ vuông và mặt nạ chữ thập. Thực tế, người ta thích loại mặt nạ vuông hơn vì nó không làm biến dạng ảnh mà lại hiệu quả. Tuy nhiên trong lọc giả trung vị, người ta lại thấy dùng cửa sổ chữ thập cho kết quả khả quan hơn nhiều.



a) mặt nạ chữ thập



b) mặt nạ vuông 5 x 5

Hình 3.5. Mặt nạ vuông và mặt nạ chữ thập.

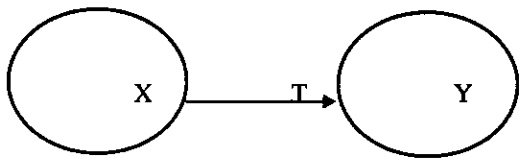
Ngoài cách phân loại dựa theo tính chất tổ hợp, trên quan điểm tần số, người ta phân các bộ lọc theo đặc trưng tần số: lọc thông thấp, thông cao, chắn dải, ... Các kỹ thuật lọc thông thấp được dùng để lọc nhiễu. Lọc thông cao dùng để làm nổi bật các chi tiết có tần số không gian cao (thí dụ như các điểm biên, các điểm gờ) mà không ảnh hưởng đến các chi tiết có tần số thấp. Do vậy, các phần tử có tần số không gian cao sẽ được cải thiện (sáng hơn), còn các phần tử có tần số không gian thấp sẽ đen đi. Kỹ thuật lọc thông cao cũng được thực hiện nhờ thao tác nhân chập. Các nhân chập thông cao hay được sử dụng:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ 1 & -1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

3.3. CÁC BIẾN ĐỔI KHÔNG GIAN: BIẾN ĐỔI FOURIER VÀ BIẾN ĐỔI KL (SPATIAL TRANSFORMS)

Các phép biến đổi là cách tiếp cận thứ hai được áp dụng trong tín hiệu số nói chung và trong xử lý ảnh số nói riêng. Phép biến đổi (transform) là thuật ngữ dùng để chỉ việc chuyển đổi sự biểu diễn của một đối tượng từ không gian này sang một không gian khác. Thí dụ, X là một đối tượng trong không gian $\mathscr{X}$, phép biến đổi T biểu diễn bởi ma trận A sẽ chuyển biểu diễn X sang Y trong không gian $\mathscr{Y}$ như sau:

Y = AX



Không gian $\mathscr{X}$ Không gian $\mathscr{Y}$

Như vậy, biến đổi ảnh (Image Transform) nhằm chuyển đổi sự biểu diễn ảnh từ một không gian ban đầu sang một không gian khác sao cho việc xử lý được tiện lợi hơn.

Để theo dõi một cách có hệ thống, trước tiên ta xem xét khái niệm chung về biến đổi ảnh trong ngữ cảnh của xử lý ảnh. Ta nói khai triển chuỗi trực giao tổng quát của một ảnh số u(m,n) , kích thước NxN là một cặp biến đổi có dạng:

$$v(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)\, a_{k,l}(m,n) \quad \text{với } k,l = 0, 1,\ldots,N-1 \tag{3.14}$$

$$u(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)\, a^*_{k,l}(m,n) \quad \text{với } k,l = 0, 1,\ldots,N-1 \tag{3.15}$$

Trong đó $\{a_{k,l}(m,n)\}$ gọi là một biến đổi ảnh. Đó chính là tập các hàm cơ sở (trong xử lý ảnh gọi là các ảnh cơ sở) .

Theo định nghĩa, một biến đổi tương ứng với A là unita và tách được (separable unitary transforms) nếu:

$AA^{*T} = A^TA^* = I$ với A là ma trận biến đổi; A^{*T} là ma trận chuyển vị của A.

Nhìn chung, trong xử lý ảnh số, ta hay dùng biến đổi đơn vị trực giao và tách được. Trong ngữ cảnh này, viết dưới dạng ma trận ta có:

$$v(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a(k,m)\, u(m,n) a(l,n) \leftrightarrow V = AUA^T \tag{3.16}$$

$$u(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} a^*(k,m) v(k,l) a^*(l,n) \leftrightarrow U = A^{*T}VA^* \tag{3.17}$$

Thí dụ, cho A là ma trận của biến đổi trực giao và U là một ảnh:

$$A = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Theo công thức trên, ta có:

$$V = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 5 & -1 \\ -2 & 0 \end{bmatrix}$$

và

$$U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 5 & -1 \\ -2 & 0 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Có rất nhiều phép biến đổi được dùng trong xử lý ảnh như biến đổi Fourrier, biến đổi Cosin, Karhuman-Loeve,... Tuy nhiên, để trong sáng cách trình bày, trong phần dưới đây ta chỉ xét 2 biến đổi quan trọng là biến đổi Fourrier TF (Fourrier Transform) và biến đổi

KL(Karhuman-Loeve). Biến đổi Cosin rất hữu ích trong nén ảnh sẽ được đề cập đến trong phần nén ảnh (chương 8).

3.3.1. Biến đổi Fourier

Trước tiên ta xem xét các khái niệm và bản chất của biến đổi TF cho tín hiệu số một chiều và hai chiều. Vì ảnh số chỉ là một phần của tín hiệu số nên phải dùng một dạng khác của biến đổi TF đó là biến đổi Fourrier rời rạc DFT(Discrete Fourrier Transform). Cuối cùng, sẽ trình bày sẽ trình bày thuật toán biến đổi nhanh FFT(Fast Fourrier Transform) để tính các DFT.

3.3.1.1. Biến đổi Four ier-TF: khái niệm và công thức

Biến đổi Fourrier cho một tín hiệu có thể hình dung như sau:

$$x(t) \xleftrightarrow{\quad TF \quad} X(f)$$

Miền thời gian **Miền tần số**

Một số ứng dụng cần miền phức, người ta dùng biến đổi phức (biến đổi z) :

$$x(n) \xleftrightarrow{\quad TZ \quad} X(z) \quad \text{với z là biến phức}$$

Biến đổi Fourrier cho một tín hiệu một chiều gồm một cặp biến đổi:

- Biến đổi thuận: chuyển sự biểu diễn từ không gian thực sang không gian tần số (phổ và pha). Các thành phần tần số này được gọi là cách biểu diễn trong không gian Fourrier của tín hiệu.

- Biến đổi ngược: chuyển đổi sự biểu diễn của đối tượng từ không gian Fourrier sang không gian thực.

a) Không gian một chiều

Cho một hàm f(x) liên tục. Biến đổi Fourrier của f(x), kí hiệu F(u), u biểu diễn tần số không gian, được định nghĩa:

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi ixu}dx \qquad (3.18)$$

trong đó:

f(x): biểu diễn biên độ tín hiệu;

$e^{-2\pi ixu}$: biểu diễn pha.

Biến đổi ngược của F(u) cho f(x) được định nghĩa:

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{2\pi ixu}du \tag{3.19}$$

b) Không gian hai chiều

Cho f(x,y) hàm biểu diễn ảnh liên tục trong không gian 2 chiều, cặp biến đổi Fourier cho f(x,y) được định nghĩa:

- Biến đổi thuận $F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(xu+yv)}dxdy$ (3.20)

u,v biểu diễn tần số không gian.

- Biến đổi ngược $f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{2\pi i(xu+yv)}dudv$ (3.21)

3.3.1.2. Biến đổi Four ier rời rạc - DFT

Biến đổi DFT được phát triển dựa trên biến đổi Fourrier cho ảnh số. Ở đây, ta dùng tổng thay cho tích phân. Biến đổi DFT tính các giá trị của biến đổi Fourrier cho một tập các giá trị trong không gian tần số được cách đều.

a) DFT cho tín hiệu một chiều

Với tín hiệu một chiều, người ta biểu diễn bởi một chuỗi trực giao các hàm cơ sở. Với các hàm liên tục, khai triển chuỗi trực giao sẽ cung cấp chuỗi các hệ số dùng trong nhiều quá trình khác nhau hay trong phân tích hàm. Khai triển Fourrier rời rạc DFT cho một dãy {u(n), n = 0, 1, ..., N-1} định nghĩa bởi:

$$v(k) = \sum_{n=0}^{N-1} u(n)W_N^{kn} \quad \text{với } k = 0, 1, ..., N-1 \tag{3.22}$$

với $W_N = e^{-j2\pi/N}$

và biến đổi ngược $u(n) = \frac{1}{N}\sum_{k=0}^{N-1} v(k)W_N^{-kn}$, n = 0, 1, ..., N-1 (3.23)

Thực tế trong xử lý ảnh người ta hay dùng DFT đơn vị:

$$v(k) = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} u(n)W_N^{kn} \quad , k = 0 , 1, ..., N-1 \tag{3.24}$$

$$u(n) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} v(k)\ W_N^{kn} \quad , n= 0, 1, ..., N-1 \tag{3.25}$$

Các DFT và DFT đơn vị có tính đối xứng. Hơn nữa khai triển DFT và DFT đơn vị của một chuỗi và biến đổi ngược lại của nó có tính chu kỳ và chu kỳ N.

b) DFT cho tín hiệu hai chiều (ảnh số)

DFT hai chiều của một ảnh M x N : {u(m,n) } là một biến đổi tách được và được định nghĩa :

$$v(k,l) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)\ W_N^{km}W_N^{ln} \quad 0 \quad l, k \quad N-1 \tag{3.26}$$

và biến đổi ngược:

$$u(m,n) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)\ \ W_N^{-km}W_N^{-ln} \quad 0 \quad m, n \quad N-1 \tag{3.27}$$

Cặp DFT đơn vị hai chiều được định nghĩa:

$$v(k,l) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)\ W_N^{km}W_N^{ln} \quad 0 \quad l, k \quad N-1 \tag{3.28}$$

$$u(m,n) = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)\ W_N^{-km}W_N^{-ln} \quad 0 \quad m, n \quad N-1 \tag{3.29}$$

Viết lại công thức 3.27 và 3.28, ta có:

$$v(k,l) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(m,n)W_N^{(km+ln)} \quad 0 \quad l, k \quad N-1 \tag{3.30}$$

$$u(m,n) = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} v(k,l)W_N^{-(km+ln)} \quad 0 \quad m, n \quad N-1 \tag{3.31}$$

Ở đây, $W_N^{(km+ln)}$ là ma trận ảnh cơ sở. Nhắc lại rằng $e^{j\alpha} = \cos(\alpha) + j\sin(\alpha)$ (công thức Ơle).

Do vậy:

$$W_N^{(km+ln)} = e^{-j2\pi(km+ln)/N} = \cos(2\pi(km+ln)/N) - j \sin(2\pi(km+ln)/N).$$

Như vậy, các hàm cơ sở trong ma trận ảnh cơ sở của biến đổi Fourier là các hàm cosine và hàm sine. Theo tính toán trên, ta thấy biến đổi Fourier biểu diễn ảnh trong không gian mới theo các hàm sine và cosine.

3.3.1.3. Một số tính chất và áp dụng

a) Tính chất

- **Đối xứng và đơn vị**

$$F^T = F, \quad F^{-1} = F^*$$

- **Chu kỳ**

$$v(k + N, l + N) = v(k,l) \quad \forall\, k, l \tag{3.32}$$

$$u(k + N, l + N) = u(k,l) \quad \forall\, k, l \tag{3.33}$$

Việc chứng minh tính có chu kỳ của biến đổi Fourier coi như một bài tập nhỏ. Bạn đọc tự kiểm nghiệm lại.

- **Phổ Fourier mẫu hoá**

$$\text{nếu } \overline{U}(m,n) = \begin{cases} U(m,n) & 0 \le m,n \le N-1 \\ 0 & \text{nếu không} \end{cases}$$

thì $\overline{U}(2k/N, 2l/N) = DFT\{u(m,n)\} = v(k,l)$ với $\overline{U}(\omega_1,\omega_2)$ là biến đổi Fourier của u(m,n).

- **Biến đổi nhanh**

Vì DFT hai chiều là tách được, do đó biến đổi V = FUF tương đương với DFT đơn vị 1chiều 2N.

- **Liên hiệp đối xứng:**

DFT và DFT đơn vị của một ảnh thực có tính đối xứng liên hợp:

$$v(N/2 \pm k, N/2 \pm l) = v^*(N/2 \pm k, N/2 \pm l) \quad 0 \le l \le N/2-1 \tag{3.34}$$

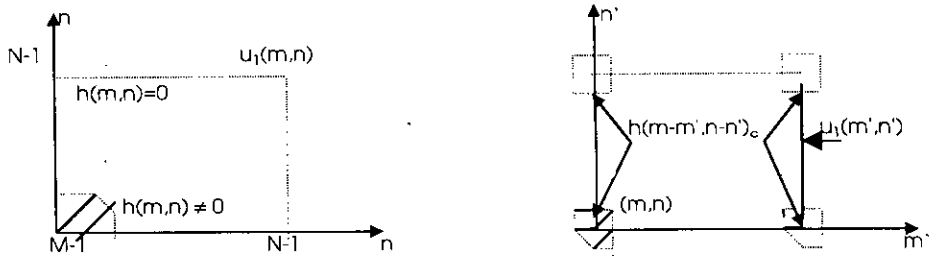
hay $$v(k,l) = v^*(N-k, N-l) \quad \text{với} \quad 0 \le l \le N/2-1 \tag{3.35}$$

b) Định lý chập cuộn 2 chiều

DFT của chập cuộn hai chiều của hai ma trận bằng tích DFT của chúng:

$$u(m,n) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} h(m-m', n-n')_c u_1(m',n') \quad 0 \le m,n \le N-1 \tag{3.36}$$

Với h(m,n), u1(m,n) là ma trận NxN và $h(m,n)_c = h(m \bmod N, n \bmod N)$. Hình 3.6 cho thấy ý nghĩa của chập tròn. Chúng là như nhau khi chu kỳ mở rộng của h(m,n) là chập trên miền NxN với $u_1(m,n)$.



a) ma trận h(m,n)

b) chập tròn h(m,n) với $u_1(m,n)$ trên miền N x N

Hình 3.6. Chập cuộn tròn.

c) Thuật toán biến đổi nhanh -FFT(Fast Fourier Transform)

- Trường hợp 1 chiều

Từ công thức $v(k) = \frac{1}{N}\sum_{n=0}^{N-1} u(n)\, W_N^{kn}$ với k=0, 1,...,N-1, ta nhận thấy:

với mỗi giá trị k ta cần N phép nhân và N phép cộng. Suy ra rằng để tính N giá trị của v(k) ta cần N^2 phép nhân. Để tính toán một cách hiệu quả , người ta dùng thuật toán tính nhanh gọi là FFT với độ phức tạp tính toán là $O(N\log_2 N)$.

Thuật toán tính nhanh có thể tóm tắt như sau:

- giả sử $N = 2^n$

- giả sử W_N là nghiệm thứ N của đơn vị: $W_N = e^{-2j\pi/N}$ và $M = \frac{N}{2}$ ta có:

$$v(k) = \frac{1}{2M}\sum_{n=0}^{2M-1} u(n)\, W_{2N}^{mk}$$

- Khai triển công thức trên ta được:

$$v(k) = \left(\frac{1}{M}\sum_{n=0}^{M-1} u(2n)\, W_{2M}^{2nk} + \frac{1}{M}\sum_{n=0}^{M-1} u(2n+1)\, W_{2M}^{(2n+1)k}\right)/2 \qquad (3.37)$$

vì $W_{2M}^{2nk} = W_{2M}^{nk}$, do đó:

$$v(k) = \frac{1}{2}\,[u_{chẵn}(n) + u_{lẻ}(n)]$$

Chú ý rằng v(k) với k = [0, M-1] là một DFT trên M = N/2. Thực chất thuật toán FFT là dùng nguyên tắc chia đôi và tính chu kỳ để tính DFT. Với k = [0, M-1] ta dùng công

thức 3.37; với k = [M, 2M-1] ta dùng phép trừ trong công thức 3.37. Có thể dùng thuật toán này có sửa đổi một chút để tính DFT ngược. Bạn đọc coi như một bài tập.

- Trường hợp 2 chiều

Do DFT 2 chiều là tách được nên từ công thức (3.29), ta có:

$$v(k,l) = \frac{1}{N}\sum_{m=0}^{N-1} W_N^{km} \sum_{n=0}^{N-1} u(m,n)\, W_N^{ln} \qquad (3.38)$$

Từ công thức 3.38, ta có cách tính DFT hai chiều như sau:

- Tính DFT 1 chiều với mỗi giá trị của x (theo cột).

- Tính DFT 1 chiều theo hướng ngược lại (theo hàng) với giá trị thu được ở trên.

3.3.2 Biến đổi KL

Biến đổi KL có nguồn gốc từ khai triển chuỗi của các quá trình ngẫu nhiên liên tục. Biến đổi KL cũng còn được gọi là biến đổi Hotelling hay phương pháp thành phần chính. Để tiện theo dõi ta cũng cần nhắc lại một số khái niệm và định nghĩa trong xử lý thống kê.

3.3.2.1 Một số định nghĩa và khái niệm

X là một biến véctơ ngẫu nhiên gồm n thành phần x_i, i = 1, 2,..., n. Mỗi thành phần x_i là giá trị ngẫu nhiên. Người ta định nghĩa:

- Kỳ vọng toán học (*Trung bình số học*) $E[x] = \int_{-\infty}^{\infty} xP(x)dx$ (3.39)

với P(x) là hàm mật độ xác suất và x là biến ngẫu nhiên liên tục.

- Mômen toán học

$$m_k = \int_{-\infty}^{\infty} x^k P(x)dx = E[x^k] \qquad (3.40)$$

m_k gọi là mô men bậc k của x.

- Tính tương quan: một tín hiệu phụ thuộc vào thời gian.
- Hàm tự tương quan của 1 tín hiệu x[t] được định nghĩa:

$$\Psi_{xx} = E[x(t).x(t+\tau)] \qquad (3.41)$$

- Hàm tương quan của 2 tín hiệu:

$$\Psi_{xy} = E[x(t).y(t+\tau)] \qquad (3.42)$$

- Cho tập các đối tượng X, *ma trận tương quan* của tập các đối tượng ký hiệu là R và được định nghĩa $R = E[X X^T] = \langle XX^T \rangle$. Viết dưới dạng ma trận ta có:

$$R = \begin{bmatrix} E[x_{11}] & E[x_{12}] & \dots & E[x_{1n}] \\ E[x_{21}] & E[x_{22}] & \dots & E[x_{2n}] \\ \dots & \dots & \dots & \dots \\ E[x_{n1}] & E[x_{n2}] & \dots & E[x_{nn}] \end{bmatrix} \quad (3.43)$$

- *Ma trận hiệp biến*, ký hiệu $A = E[(X-M)(X-M^T)]$

$$= \langle (X-M)(X-M)^T \rangle \quad (3.44)$$

Trong một số trường hợp $A = \langle XX^T \rangle - \langle MM^T \rangle = R - \langle MM^T \rangle$ (*). Nếu đối tượng không tương quan (độc lập) lúc đó ma trận A là ma trận đường chéo. Có nghĩa là:

$$a_{i,i} = x_i^2 - m_i^2 \neq 0 \text{ còn } a_{i,j} = 0 \text{ với } i \neq j.$$

3.3.2.2 Cơ sở lý thuyết của biến đổi KL

Đây là phép biến đổi không gian n chiều thành không gian m chiều, với m < n. Mỗi thành phần của véctơ miêu tả một đặc tính của đối tượng. Nếu ta biến đổi được từ không gian n chiều về không gian m chiều, như vậy ta sẽ làm giảm được thông tin dư thừa (theo thuật ngữ trong xử lý ảnh hay nhận dạng gọi là *giảm thứ nguyên*).

Mục đích của biến đổi KL là chuyển từ không gian n chiều sang không gian trực giao m chiều sao cho sai số bình phuương là nhỏ nhất. Gọi U là tập các véctơ cơ sở trong không gian trực giao $U = \{u_1, u_2, \dots, u_n\}$,

$$u_j = \begin{cases} u_{1j} \\ u_{2j} \\ \dots \\ u_{nj} \end{cases}$$

với $j = 1, 2, \dots, n$ và

$$u_i . u_k = \begin{cases} 0 & \text{nếu } i \neq k \\ 1 & \text{nếu } i = k. \end{cases}$$

Mọi véctơ y trong không gian trực giao có thể viết:

$y = \varphi_1 u_1 + \varphi_2 u_2 + \dots + \varphi_n u_n = \Phi U$ với $\Phi = \varphi_1, \varphi_2, \dots, \varphi_n$

$\Rightarrow \Phi = U^T y.$

Gọi $\overline{X}$ là kết quả thu được trong không gian m chiều và $\overline{X} = \varphi_1 u_1 + \varphi_2 u_2 + \dots + \varphi_m u_m \approx X$.

Sai số trong phép biến đổi $\varepsilon = X - X = \sum_{i=1}^{n} \varphi_i u_i - \sum_{i=1}^{m} \varphi_i u_i = \sum_{i=m+1}^{n} \varphi_i u_i$ (3.45)

Sai số trung bình bình phương $\zeta = E[\varepsilon^2] = E[(\overline{X} - X)^T(\overline{X} - X)]$ (3.46)

$$= <(\overline{X} - X)^T(\overline{X} - X)>$$

$$= <\sum_{i=m+1}^{n} \varphi_i^2>$$

$$= \sum_{i=m+1}^{n} <\varphi_i^2> \quad (3.47)$$

mà $\Phi = U^T X$, do đó $\zeta = \sum_{i=m+1}^{n} (u_i^T X)(u_i X)^T = \sum_{i=m+1}^{n} u_i^T <XX^T> u_i$ (3.48)

Theo định nghĩa của R, phương trình 3.48 trở thành $\zeta = \sum_{i=m+1}^{n} u_i^T R u_i$ (3.49)

ζ đạt min khi (3.49) đạt min.

Đặt $\eta = \zeta + \sum_{i=m+1}^{n} \lambda_i (1 - u_i^T u_i)$. (3.50)

Như vậy η đạt min khi (3.50) min. Để tìm min của (3.50) ta dùng phương pháp đạo hàm và dẫn đến việc giải phương trình:

$$(R - \lambda I) u_i = 0 \quad (3.51)$$

Phương trình (3.51) gọi là phương trình đặc trưng của R với λ_i là các trị riêng và u_i là các véctơ riêng tương ứng. Đây chính là cơ sở lý thuyết của biến đổi KL.

3.3.2.3. Biến đổi KL

Định nghĩa và khái niệm

Cho u là một véctơ các số thực ngẫu nhiên; véctơ cơ sở của biến đổi KL là các véctơ riêng trực giao của ma trận hiệp biến R(định nghĩa trong phần 3.3.2.1) cho bởi phương trình: $R\varphi_k = \lambda_k \varphi_k$; $0 \le k \le N-1$

Biến đổi KL của u là $v = \Phi^{*T} u$ (3.52)

và biến đổi ngược $u = \Phi v = \sum_{k=0}^{N-1} v(k)\, \Phi_k$ (3.53)

u là véctơ cột, v là véctơ hàng và Φ_k là cột thứ k của ma trận Φ.

Biến đổi Φ đưa R về dạng đường chéo :

$$\Phi^{*T} R\Phi = \lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_N \end{bmatrix}$$

Thường người ta hay làm việc với ma trận A hơn.

Biến đổi KL của ảnh

Nếu một ảnh u(m,n) NxN được biểu diễn bởi trường ngẫu nhiên, ma trận A cho bởi:

$$E[u(m,n)u(m',n')] = r(m,n;m',n') \quad 0 \le m,m',n,n' \le N-1 \tag{3.54}$$

thì ảnh cơ sở của biến đổi KL là các hàm riêng, chuẩn và trực giao $\phi_{k,l}$ là lời giải của phương trình:

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} r(m,n;m',n')\,\phi_{k,l} = \lambda_{k,l}\,\phi_{k,l} \tag{3.55}$$

Theo ký pháp ma trận ta có: $R\Psi_i = \lambda_i\,\phi_i \qquad i = 0, 1, ..., N^2-1$ (3.56)

với ϕ_i là véctơ N^2 x 1 biểu diễn của $\Psi_{k,l}$ và R là ma trận N^2 x N^2 ánh xạ vào véctơ u, R = E[uu].

Nếu R là tách được thì ma trận $\Psi_{N2 \times N2}$ mà các cột là Ψ_i sẽ tách được:

$$\phi_{k,l}(m,n) = \phi_1 \otimes \phi_2 \text{ hay } R = R_1 \otimes R_2 \tag{3.57}$$

Biến đổi KL của U là $V = \Psi^{*T}u = \phi_1^{*T} \otimes \phi_2^{*T}$

và biến đổi ngược $\qquad U = \phi_1 V \phi_2$ (3.58)

3.4. TOÁN TỬ XỬ LÝ ĐIỂM ẢNH

Ảnh thô có cấu trúc đơn giản, song lại rất phức tạp về nội dung. Như chúng ta biết, ảnh là một tập hợp các điểm ảnh, chứa một lượng thông tin khá lớn. Thường để xử lý ảnh, người ta hay biểu diễn ảnh dưới một dạng khác để có thể làm rõ một số tính chất của chúng. Xử lý điểm ảnh thực chất là dùng các ánh xạ nhằm biến đổi giá trị của một điểm chỉ dựa vào giá trị của chính nó mà không quan tâm tới các giá trị của các điểm ảnh khác. Một cách toán học, ánh xạ đó được định nghĩa như sau:

$$v(m,n) = f(u(m,n)L)$$

trong đó: - u(m,n) thể hiện giá trị cường độ sáng tại toạ độ (m,n);

- v(m,n) là giá trị cường độ sáng thu được sau phép biến đổi;

- f là hàm biến đổi. Nó có thể là hàm liên tục hay hàm rời rạc.

Chi tiết về các hàm này và cách vận dụng được trình bày kỹ trong chương 4 (4.1.1).

Xử lý điểm ảnh là một trong các phép xử lý cơ bản và đơn giản. Có 2 cách tiếp cận trong cách xử lý này: dùng một hàm thích hợp tuỳ theo mục đích cải thiện ảnh để biến đổi giá trị của điểm ảnh (mức xám) sang một giá trị khác (mức xám mới). Cách thứ hai là dựa vào kỹ thuật biến đổi lược đồ xám (histogram).

3.4.1. Xử lý điểm ảnh bằng ánh xạ biến đổi

Bản chất của xử lý điểm ảnh như đã nói trên là nhằm biến đổi giá trị của một điểm ảnh bằng một hàm tuyến tính hay phi tuyến (hàm mũ, hàm lôgarít). Các phép xử lý này là cơ sở cho biến đổi độ tương phản của ảnh: co giãn, tăng giảm và biến đổi độ tương phản vì độ tương phản trên một ảnh chỉ phụ thuộc vào độ sáng của mỗi điểm ảnh. Giả sử ta dùng một hàm phi tuyến dạng f = alog():

$$Y[m,n] = alog(X[m,n]).$$

Nếu ảnh có kích thước 512 x 512 ta cần 512^2 phép biến đổi. Một cách tổng quát, nếu ảnh có kích thước NxN thì phép biến đổi sẽ có độ phức tạp tính toán là $O(N^2)$. Nếu chú ý rằng ảnh gồm NxN điểm song chỉ có L mức xám (L rất nhỏ so với N^2) và phép biến đổi chỉ nhằm biến đổi một mức xám $l \in L$ sang một mức xám $l' \in L'$ (mức xám kết quả) thì ta có thể thực hiện nhanh hơn. Do vậy, ta có cách tính sau:

- Tính L giá trị của hàm f và lưu vào một bảng: $y_i = f(x_i)$ với i=1,2,..., L.

- Duyệt toàn bộ ảnh, với mỗi điểm ảnh ta tra giá trị trong bảng (không cần tính) và thu được ảnh mới.

Kỹ thuật này có tên gọi là kỹ thuật bảng tra - LUT(Look Up Table). Để minh hoạ, xét thí dụ sau. Cho ảnh số X:

$$X = \begin{bmatrix} 2 & 1 & 2 & 3 \\ 3 & 1 & 2 & 5 \\ 2 & 2 & 3 & 4 \\ 3 & 2 & 3 & 2 \end{bmatrix}$$

Ảnh này có 16 điểm song chỉ có 5 mức xám. Hàm biến đổi là hàm alog(). Bảng tra có giá trị:

Mức xám	Bảng tra (LUT)
1	alog(1)
2	alog(2)
3	alog(3)
4	alog(4)
5	alog(5)

Ảnh thu được sau phép biến đổi:

$$X = \begin{bmatrix} a\log(2) & a\log(1) & a\log(2) & a\log(3) \\ a\log(3) & a\log(1) & a\log(2) & a\log(5) \\ a\log(2) & a\log(2) & a\log(3) & a\log(4) \\ a\log(3) & a\log(2) & a\log(3) & a\log(2) \end{bmatrix}$$

Thuật toán biến đổi được mô tả như sau:

{Bảng tra có tên là LUT và có L phần tử}

a) Tính bảng LUT

For k = 1 to L do LUT[k] := $f(x_k)$

b) Biến đổi

For each pixel X[i,j] do Y[i,j] :=LUT(X[i,j])

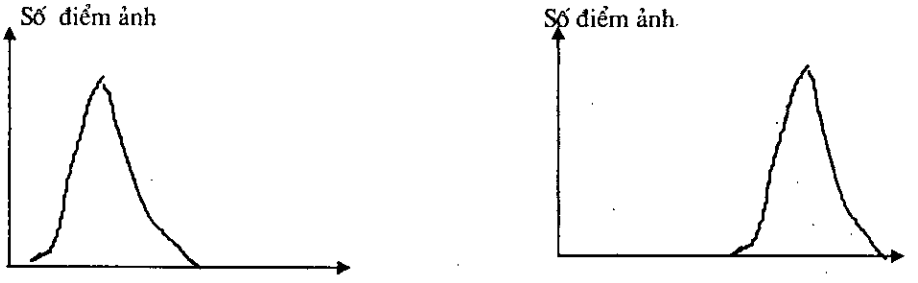
Như vậy, để có thể lập trình, phụ thuộc vào các hàm biến đổi khác nhau, ta chỉ cần viết hàm tính bảng tra (tham số là hàm) còn phép biến đổi là như nhau.

3.4.2 Lược đồ mức xám (histogram)

Lược đồ mức xám của một ảnh, từ nay về sau ta qui ước gọi là *lược đồ xám*, là một hàm cung cấp tần suất xuất hiện của mỗi mức xám (grey level).

Lược đồ xám được biểu diễn trong một hệ toạ độ vuông góc x,y. Trong hệ toạ độ này, trục hoành biểu diễn số mức xám từ 0 đến N, N là số mức xám (256 mức trong trường hợp chúng ta xét). Trục tung biểu diễn số điểm ảnh cho một mức xám (số điểm ảnh có cùng mức xám). Cũng có thể biểu diễn khác một chút: trục tung là tỷ lệ số điểm ảnh có cùng mức xám trên tổng số điểm ảnh.

Lược đồ xám cung cấp rất nhiều thông tin về phân bố mức xám của ảnh. Theo thuật ngữ của xử lý ảnh gọi là *tính động* của ảnh. Tính động của ảnh cho phép phân tích trong khoảng nào đó phân bố phần lớn các mức xám của ảnh: ảnh rất sáng hay ảnh rất đậm.



a) ảnh đậm **b) ảnh nhạt**

Hình 3.8. Lược đồ xám của ảnh

Nếu ảnh sáng, lược đồ xám nằm bên phải (mức xám cao), còn ảnh đậm lược đồ xám nằm bên trái (mức xám thấp).

Theo định nghĩa của lược đồ xám, việc xây dựng nó là khá đơn giản. Thuật toán xây dựng lược đồ xám có thể mô tả như sau:

Bắt đầu

H là bảng chứa lược đồ xám (là vectơ có N phần tử)

a. Khởi tạo bảng

For i:=1 to N do H[i] := 0;

b. Tạo bảng

Với mỗi điểm ảnh I(x,y) tính H[I(x,y)] = H[I(x,y)] + 1

c. Tính giá trị Max của bảng H. Sau đó hiện bảng trong khoảng từ 0 đến Max.

Kết thúc

Lược đồ xám là một công cụ hữu hiệu dùng trong nhiều công đoạn của xử lý ảnh như *tăng cường ảnh* (xem chương 4). Dưới đây ta xem xét một số biến đổi lược đồ xám hay dùng.

3.4.3. Biến đổi lược đồ xám

Trong tăng cường ảnh, các thao tác chủ yếu dựa vào phân tích lược đồ xám. Trước tiên ta xét bảng tra LUT(Look Up Table). Bảng tra LUT là một bảng chứa biến đổi một mức xám i sang mức xám j như đã nói trong phần 3.4.1. Một cách toán học, LUT được định nghĩa như sau:

- Cho G_I là tập các mức xám ban đầu $G_I = \{0, 1, ..., N_I\}$

- Cho G_F là tập các mức xám kết quả $G_F = \{0, 1, ..., N_F\}$

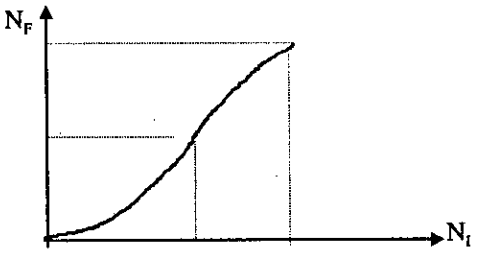
để cho tiện ta cho $N_I = N_F = 255$.

- f là ánh xạ từ G_I vào G_F: $\forall g_i \in G_i$ sẽ $\exists g_f \in G_F$ mà $g_f = f(g_i)$

Với mỗi giá trị của mức xám ban đầu ứng với một giá trị kết quả. Việc chuyển đổi một mức xám ban đầu về một mức xám kết quả tương ứng có thể dễ dàng thực hiện được nhờ một bảng tra.

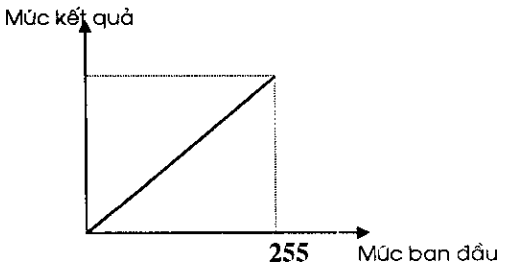
Khi đã xây dựng được bảng, việc sử dụng bảng là khá đơn giản. Người ta xem xét mức xám của mỗi điểm ảnh, nhờ bảng tra tính được mức xám kết quả. Gọi là bảng tra,

thực ra là một véctơ có $N_I + 1$ phần tử. Mỗi phần tử của bảng chứa một giá trị mức xám kết quả. Có hai kiểu bảng tra: bảng đồng nhất và bảng nghịch đảo. Với bảng đồng nhất, giá trị



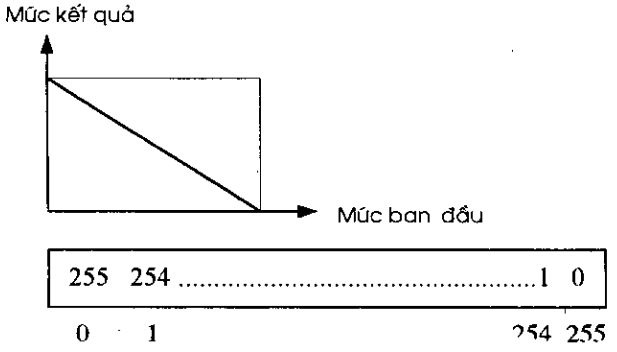
Hình 3.9. Ánh xạ biến đổi lược đồ xám.

mức xám ban đầu cũng chính là giá trị mức xám kết quả; còn với bảng nghịch đảo, nếu giá trị mức xám ban đầu là g_I thì giá trị mức xám kết quả là $255-g_I$.



0	1	2	. . . . .	254	255
0	1	2		254	255

Hình 3.10 a. LUT đồng nhất.

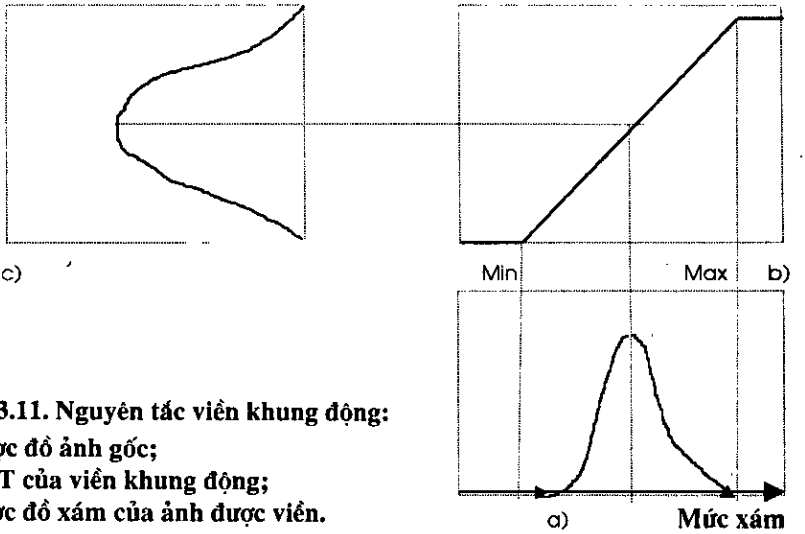


Hình 3.10 b.
LUT nghịch đảo.

Một trong những ứng dụng phổ biến của LUT là viền khung động. Một số ảnh ban đầu hoặc có thể là rất đậm hay rất nhạt, hoặc độ tương phản thấp. Điều này có thể là do trong ảnh ban đầu, các mức xám có thể vượt lên cao hoặc xuống dưới tỷ lệ, hay tập trung lại trong một vùng rất hẹp (trên lược đồ xám thể hiện rõ điều này).

Mục đích của LUT là phân bố lại mức xám để chúng có thể phủ trên toàn dải - đó chính là viền khung động. Việc chọn giá trị Min và Max là phụ thuộc vào từng ứng dụng.

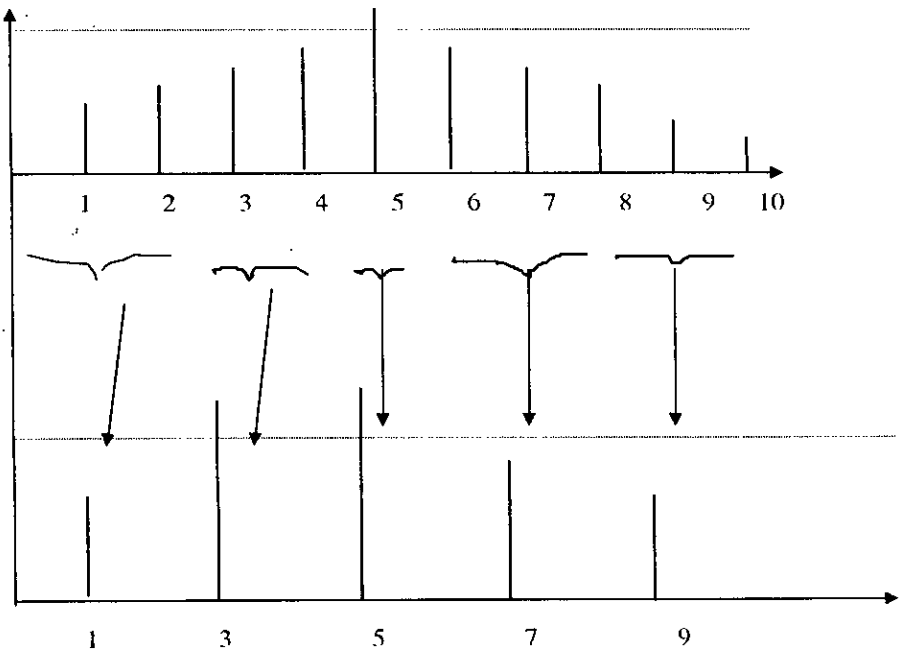
Một ứng dụng khác của LUT là làm nổi bật một số dải mức xám của ảnh. Điều này có thể thực hiện được nhờ viền khung động tại miền quan tâm, bên ngoài miền đặt giá trị là 0 hay nhị phân hoá ảnh (binarisation).



Hình 3.11. Nguyên tắc viền khung động:
a) Lược đồ ảnh gốc;
b) LUT của viền khung động;
c) Lược đồ xám của ảnh được viền.

Với một ảnh tự nhiên được lượng hoá một cách tuyến tính, phần lớn các điểm ảnh có giá trị thấp hơn độ sáng trung bình. Trong miền tối, ta khó có thể cảm nhận các chi tiết của ảnh. Thực tế cần phải khắc phục nhược điểm này bằng cách biến đổi lược đồ xám. Người ta biến đổi lược đồ sao cho tiến gần tới lược đồ định trước. Có nhiều phương pháp, trong đó phương pháp phổ dụng là *san bằng lược đồ* (histogram equalisation).

Nếu ảnh có kích thước pxp và ảnh kết quả được mã hoá trên N_F mức xám, thì số điểm ảnh cho 1 mức xám trong lược đồ cân bằng lý tưởng sẽ là hằng số và bằng p^2/N_F (N_F là số mức xám đầu ra). Trên thực tế, N_F thường nhỏ hơn N_I (số mức xám ban đầu). Nguyên tắc san bằng lược đồ được minh hoạ trong hình 3.12.



Hình 3.12. Cân bằng lược đồ.

Việc san bằng lược đồ được thực hiện theo thuật toán:

/*

Ima: ảnh gốc cần san bằng

Histo: lược đồ xám của ảnh

Transfo: bảng san bằng lược đồ

BatDau, KetThuc : điểm bắt đầu và điểm kết thúc mỗi dải xét.

Bande, CentreBande: độ rộng băng và trung điểm của dải

N_I: Số mức xám của ảnh gốc

N_F: Số mức xám của ảnh kết quả */

a. Khởi tạo

TBLituong <-- pxp/N_F;

Bande <-- N_I/N_F;

CentreBande <-- Bande/2;

BatDau, KetThuc <-- 0;

b. Tính và biến đổi lược đồ

```
While KetThuc < N_I  do
        Begin  Tong <-- 0;
                While (Tong < TBLituong) and(KetThuc < N_I ) do
                Begin Tong <-- Tong + Histo(KetThuc);
                        Inc(KetThuc)
                End;
        For  i := BatDau to KetThuc -1 do Trandfo[i] <-- CentreBande;
        CentreBande <- - CentreBande + Bande;
        ebut <-- KetThuc;
End
```

c. Tính ảnh kết quả

```
For  i := 1 to N  do
        For j :=1 to N do Begin
                        Pic <-- Ima[i,j];
                        Ima[i,j] <-- Transfo[Pic]
                End
```

Lưu ý rằng, N_F càng hạn chế thì việc tích hợp càng quan trọng vì giá trị p^2/N_F sẽ tăng. Trên thực tế, người ta hay dùng $N_F = N_I/2$ và lặp lại nhiều lần quá trình san bằng. Thí dụ với một ảnh 128 x 128 mã hoá trên 256 mức xám, nếu muốn lược đồ san bằng trên 64 mức xám, số lượng trung bình các điểm ảnh lý tưởng sẽ tiệm cận $128^2/64 = 256$.

3.5. MÔ HÌNH THỐNG KÊ

Mô hình thống kê có một ý nghĩa rất quan trọng trong biểu diễn ảnh cũng như trong nhiều quá trình của xử lý ảnh. Trong mô hình này, mỗi điểm ảnh được xem như một biến ngẫu nhiên u. Một ảnh là một hàm mẫu của một ma trận biến ngẫu nhiên còn gọi là trường ngẫu nhiên (random field). Thực tế, số biến ngẫu nhiên là rất lớn (262144 biến cho một ảnh 512 x 512). Đều này gây không ít khó khăn vì để đặc tả một hàm mật độ phải cần một khối lượng đo hay quan sát rất lớn. Vì vậy, người ta nghĩ đến sử dụng các đại luợng đặc trưng của phân bố xác suất như: kỳ vọng toán học, moment (đã nêu trong phần 3.3.3).

Những đặc trưng này rất có ích trong kỹ thuật xử lý ảnh không chỉ cho một ảnh mà là cho một lớp ảnh.

- **Mô hình hiệp biến (covariance model)**

Trong mô hình này, người ta chọn kỳ vọng toán học là hằng số μ, còn hiệp biến biểu diễn bởi mô hình mũ tách được hay không tách được (separable or nonseparable). Trong mô hình tách được, hiệp biến 2 chiều có thể biểu diễn bởi tích của hai hiệp biến một chiều:

$$r(m,n;m',n') = r_1(m,n)\, r_2(m',n') \tag{3.59}$$

$$r(m,n) = r_1(m)\, r_2(n) \tag{3.60}$$

Phương trình 3.59 biểu diễn mô hình không ổn định, còn 3.60 biểu diễn mô hình ổn định. Trong xử lý ảnh, người ta hay dùng hiệp biến ổn định tách được dưới dạng mũ:

$$r(m,n) = \sigma^2 p_1^{|m|} p_2^{|n|} \tag{3.61}$$

với $|p_1| < 1$, $|p_2| < 1$ và σ^2 là phương sai của trường ngẫu nhiên;

$p_1 = r(1,0)/\sigma^2$, $p_2 = r(0,1)/\sigma^2$.

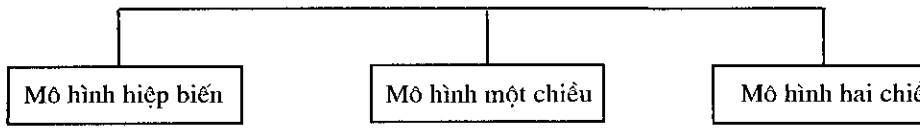
Hiệp biến không tách được cũng được biểu diễn dưới dạng mũ:

$$r(m,n) = \sigma^2 \sqrt{\alpha_1 m^2 + \alpha_2 n^2} \tag{3.62}$$

Khi $\alpha_1 = \alpha_2 = \alpha$, r(m,n) trở thành khoảng cách Euclide d và $r(m,n) = \alpha^2 p^2$, với: $p = \exp(-|\alpha|)$. Điều này lý giải tại sao lại gọi là mô hình mũ.

Mô hình hiệp biến tách được rất thuận tiện cho việc phân tích các thuật toán xử lý ảnh. Mô hình hiệp biến không tách được là một mô hình tốt hơn, tuy vậy nó không thuận tiện cho việc phân tích. Mô hình hiệp biến rất có ích trong việc biến đổi ảnh, khôi phục và nén ảnh.

Đối ngược với biểu diễn trường ngẫu nhiên dùng kỳ vọng toán học và hiệp biến, một cách khác là coi nó như đầu ra của một hệ thống tuyến tính mà đầu vào là trường ngẫu nhiên với một số tính chất thống kê đã biết (thí dụ như nhiễu trắng đầu vào). Hệ thống tuyến tính biểu diễn bởi phương trình vi phân và hữu ích trong việc phát triển các thuật toán xử lý ảnh với hiệp biến được biết đến với tên gọi " *phân tích phổ*". Hình 3.13 dưới đây tổng kết các mô hình thống kê trong xử lý ảnh:



Hình 3.13. Một số mô hình thống kê.

3.5.1. Mô hình 1 chiều nhân quả (one dimensional causal model)

Một cách đơn giản để đặc trưng một ảnh là xem xét tín hiệu một chiều xuất hiện ở đầu ra của một lưới quét hàng có nghĩa là một chuỗi hàng hay cột. Nếu sự phụ thuộc giữa hàng/cột là không tính đến, hệ thống tuyến tính một chiều tỏ ra rất có ích cho việc mô hình hoá các tín hiệu như vậy.

Giả sử u(n) là một chuỗi các số thực ngẫu nhiên với trung bình 0 và hiệp biến là r(n). Nếu U(n) được coi như đầu ra của một hệ thống tuyến tính bất biến ổn định H(z) mà đầu vào là một chuỗi ngẫu nhiên dừng trung bình 0: ε(n), hàm phân bố rời rạc có dạng:

$$S(z) = H(z)S_\varepsilon(f)H(z^{-1}) \tag{3.63}$$

với $S_\varepsilon(z)$ là hàm phân bố rời rạc của ε(n).

Chuỗi ngẫu nhiên u(n) trung bình 0 gọi là một quá trình tự điều chỉnh bậc p khi nó có thể được khởi tạo như đầu ra của hệ thống:

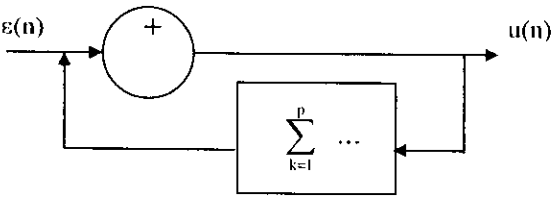
$$u(n) = \sum_{k=1}^{p} a(k)u(n-k) + \varepsilon(n) \qquad \forall n \tag{3.64}$$

$$E[\varepsilon(n)] = 0,\ E[\varepsilon(n)^2] = \beta^2,\ E[\varepsilon(n)u(m)] = 0 \quad m < n$$

Gọi $$\pi(n) = \sum_{k=1}^{p} a(k)u(n-k) \tag{3.65}$$

là dự đoán bình phương trung bình tuyến tính tốt nhất của u(n) dựa vào tất cả những cái trước song chỉ phụ thuộc duy nhất vào mẫu gần nhất. Nếu u(n) là chuỗi Gauss, điều đó có nghĩa là AR bậc p của một quá trình Markov và phương trình 3.64 trở thành:

$$u(n) = \pi(n) + \varepsilon(n) \tag{3.66}$$



Hình 3.14. Mô hình quay lui AR.

Điều đó có nghĩa là mẫu ở thời điểm n là tổng của các ước lượng dự đoán sai số tối thiểu ε(n). Thí dụ: hiệp biến của lưới quét hàng của một ảnh có thể thu được khi xem xét hiệp biến giữa 2 điểm trên cùng một hàng.

Giải phương trình:

$$\sigma^2\begin{pmatrix} 1 & p \\ p & 1 \end{pmatrix} \times \begin{pmatrix} a[1] \\ a[2] \end{pmatrix} = \sigma^2\begin{pmatrix} p \\ p^2 \end{pmatrix}$$

ta sẽ thu được nghiệm a[1] = p, a[2] = 0 và $\beta^2 = \delta^2 (1-p^2)$. Ta suy ra biểu diễn tương ứng của lưới quét hàng của một ảnh có điểm trung bình μ là một mô hình AR bậc nhất:

$$x(n) = px(n-1) + \varepsilon(n)$$

$$u(n) = x(n) - \mu$$

$$r(n) = \sigma^2(1-p^2)\,\delta(n) \tag{3.67}$$

Qua thí dụ này, chúng ta thấy mô hình AR rất hữu ích trong biểu diễn ảnh quét theo hàng. Còn nhiều biến đổi của mô hình AR, bạn đọc quan tâm xem trong Anin.K.Jain [1] như mô hình AR đồng nhất, MA (Moving Averge), ARMA, v.v.

3.5.2. Mô hình nhân quả hai chiều

Khái niệm nhân quả không mở rộng một cách tự nhiên cho mô hình hai chiều hay nhiều chiều. Kỹ thuật xử lý theo từng dòng áp dụng khá tốt cho mô hình một chiều như đã trình bày ở trên, song lại không áp dụng cho cấu trúc hai chiều khi sự phụ thuộc giữa các

hàng. Bởi vì quan hệ nhân quả không phải là cái chủ yếu trong cấu trúc hai chiều, do vậy ta phải tìm những cấu trúc khác đặc trưng cho mô hình hai hay nhiều chiều. Người ta đưa ra 3 dạng chính tắc: dạng nhân quả, dạng bán nhân quả và dạng không nhân quả.

Mỗi dạng đều có đặc trưng riêng. Thí dụ dạng nhân quả cho những thuật toán đệ qui trong nén dữ liệu bởi phương pháp mã hoá điều xung vi phân (DPCM).

Dạng bán nhân quả là nhân quả theo một chiều song lại không nhân quả theo chiều kia. Trong mô hình này, người ta dùng các thuật toán đệ qui cho chiều nhân quả, còn chiều khác có thể dùng biến đổi dải tương quan đơn vị.

Mô hình không nhân quả dẫn tới các thuật toán dựa vào các biến đổi như biến đổi KL (trình bày ở trên) hay mô hình MVR. Nhiều toán tử xử lý ảnh không gian là không nhân quả như kỹ thuật mặt nạ Gradient sẽ nói tới trong chương 5. Nhìn chung lý thuyết phần này khá phức tạp và phần nào vượt quá khuôn khổ giáo trình.

Bài tập chương 3

1. Cho ma trận hiệp biến R trong không gian 3 chiều:

$$R = \begin{bmatrix} -3 & -1 & 0 \\ -1 & 3 & 3 \\ 0 & 0 & 3 \end{bmatrix}$$

Hãy xác định các trị riêng và véctơ riêng tương ứng trong không gian trực giao 3 chiều.

2. Cho ảnh số:

a)

$$I = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 3 & 1 & 4 & 5 \\ 7 & 0 & 6 & 4 \\ 8 & 1 & 5 & 3 \end{bmatrix}$$

Hãy tính ảnh đầu ra với nhân chập H_3 trong tài liệu $Y = H_3 \otimes I$

b) Cho nhân chập H_x:

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

Tính tích chập $H_x \otimes I$

3. Viết thủ tục tính lược đồ xám của một ảnh đã cho và vẽ đồ thị biểu diễn lược đồ. Số mức xám nhập vào theo tham số.

4. Viết thủ tục xây dựng bảng tra LUT của một ảnh và biến đổi ảnh theo LUT.

5. Viết thủ tục thực hiện việc san bằng lược đồ xám của một ảnh dựa vào giải thuật san bằng lược đồ xám.

6. Dựa vào giải thuật lọc trung vị hãy viết một thủ tục thực hiện lọc trung vị cho một ảnh.

4

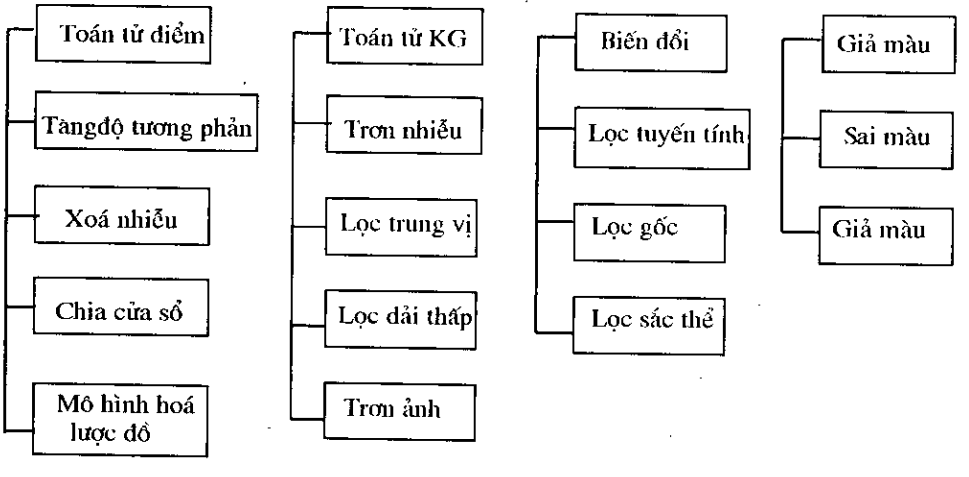
XỬ LÝ VÀ NÂNG CAO CHẤT LƯỢNG ẢNH

IMAGE ENHANCEMENT

Nâng cao chất lượng ảnh là một bước quan trọng, tạo tiền đề cho xử lý ảnh. Mục đích chính là nhằm làm nổi bật một số đặc tính của ảnh như thay đổi độ tương phản, lọc nhiễu, nổi biên, làm trơn biên ảnh, khuếch đại ảnh, ... Tăng cường ảnh và khôi phục ảnh là 2 quá trình khác nhau về mục đích. ***Tăng cường ảnh*** bao gồm một loạt các phương pháp nhằm hoàn thiện trạng thái quan sát của một ảnh. Tập hợp các kỹ thuật này tạo nên giai đoạn tiền xử lý ảnh. Trong khi đó, ***khôi phục ảnh*** nhằm khôi phục ảnh gần với ảnh thực nhất trước khi nó bị biến dạng do nhiều nguyên nhân khác nhau.

4.1. Các kỹ thuật tăng cường ảnh (Image Enhancement)

Nhiệm vụ của tăng cường ảnh không phải là làm tăng lượng thông tin vốn có trong ảnh mà làm nổi bật các đặc trưng đã chọn làm sao để có thể phát hiện tốt hơn, tạo thành quá trình tiền xử lý cho phân tích ảnh.



Hình 4.1. Các kỹ thuật cải thiện ảnh.

Tăng cường ảnh bao gồm: điều khiển mức xám, dãn độ tương phản, giảm nhiễu, làm trơn ảnh, nội suy, phóng đại, nổi biên v.v. Các kỹ thuật chủ yếu trong tăng cường ảnh được mô tả qua hình 4.1.

4.1.1. Cải thiện ảnh dùng toán tử điểm

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 $v \in [0,N]$: $v = f(u)$ (chương 3 — phần 3.4). Ứng dụng chính của toán tử điểm là nhằm biến đổi độ tương phản của ảnh. Ánh xạ f tuỳ theo các ứng dụng khác nhau sẽ có dạng khác nhau và được liệt kê trong bảng sau:

1) Tăng độ tương phản

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

Các độ dốc α, β, γ xác định độ tương phản tương đối. L là số mức xám cực đại.

2) Tách nhiễu và phân ngưỡng

$$f(u) = \begin{cases} 0 & 0 \le u < a \\ \alpha u & a \le u < b \\ L & u \ge b \end{cases}$$

Khi a = b = t gọi là phân ngưỡng

3) Biến đổi âm bản

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

4) Cắt theo mức

$$f(u) = \begin{cases} L & a \le u \le b \\ 0 & \text{khác đi} \end{cases}$$

5) Trích chọn bit

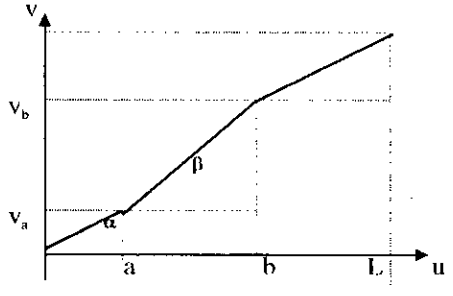
$$f(u) = (i_n - 2i_{n-1})L, \text{ với } i_n = \mathrm{Int}[it/2^{n-1}], \; n = 1, 2, \ldots, B$$

4.1.1.1. Tăng độ tương phản (stretching contrast)

Trước tiên cần làm rõ khái niệm độ tương phản. Ảnh số là tập hợp các điểm, mà 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 khác nhau. Như vậy, độ tương phản biểu diễn

sự thay đổi độ sáng của đối tượng so với nền. Một cách nôm na, độ tương phản là độ nổi của điểm ảnh hay vùng ảnh so với nền. Với định nghĩa này, nếu ảnh của ta có độ tương phản kém, ta có thể thay đổi tuỳ 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, ta 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 lôgarít). Khi dùng hàm tuyến tính các độ dốc α, β, γ phải chọn *lớn hơn một trong miền cần dãn*. Các tham số a và b (các cận) có thể chọn khi xem xét lược đồ xám của ảnh.



Hình 4.2. Dãn độ tương phản.

Chú ý, nếu dãn độ tương phản bằng hàm tuyến tính ta có:

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

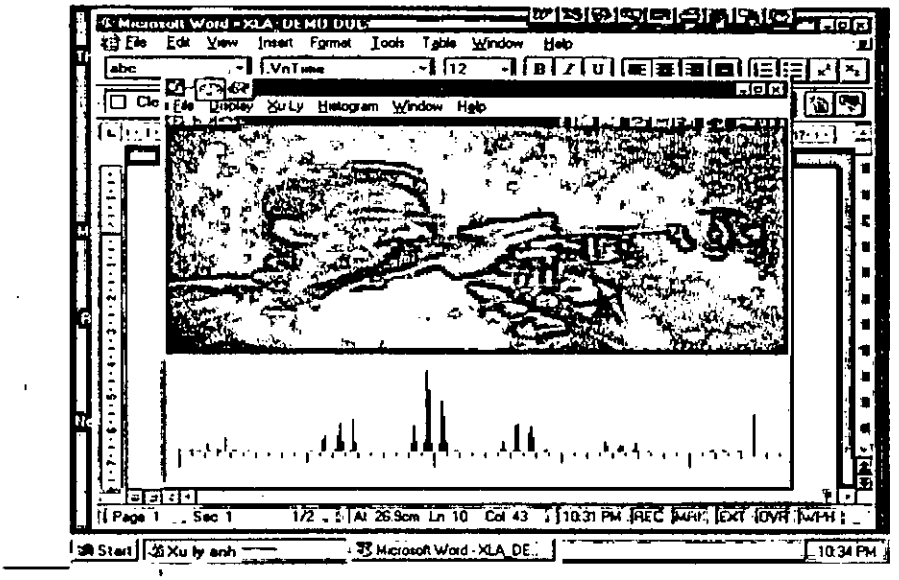
Hàm mũ hay dùng trong dãn độ tương phản có dạng:

$$f = (X[m,n])^p$$

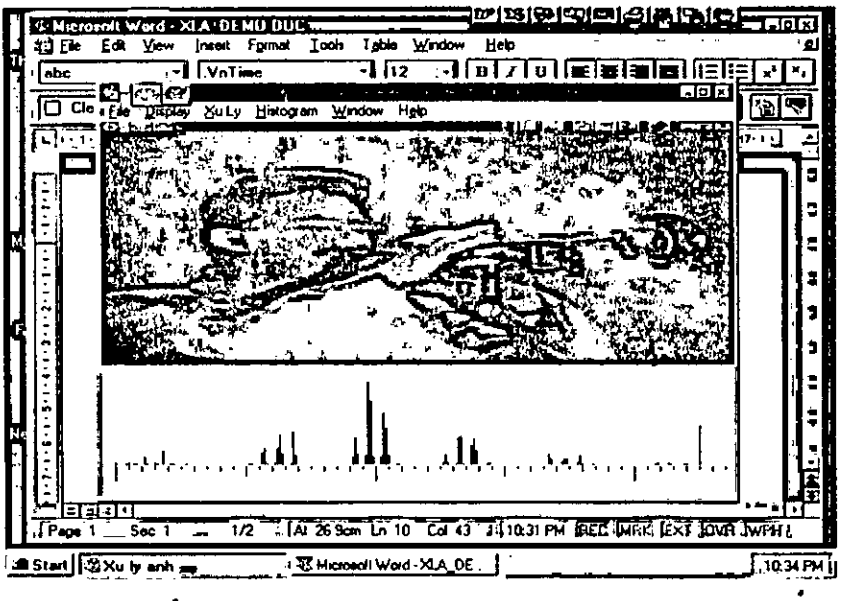
Với các ảnh hạng động nhỏ, p thường chọn bằng 2.

4.1.1.2. Tách nhiễu và phân ngưỡng

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

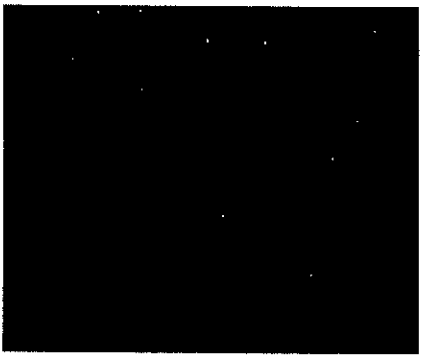


A) Ảnh nguồn cùng lược đồ xám. Chỉ số màu cao nhất là 97



b)Ảnh sau khi dãn độ tương phản với $\alpha = 3$, $\beta = 2$ và =1.

Hình 4.3. Ảnh gốc và ảnh kết quả sau khi dãn.

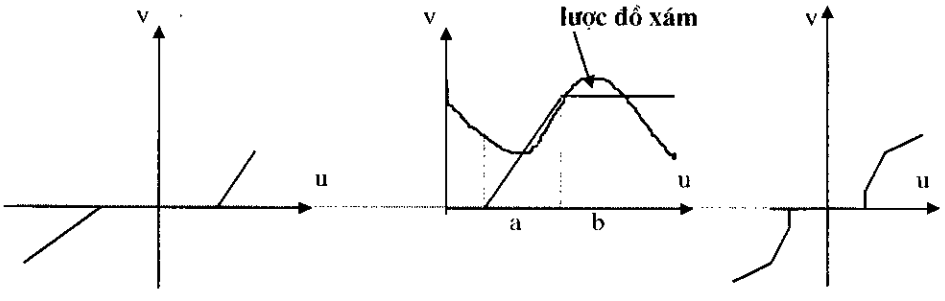


c) ảnh gốc(ảnh vệ tinh — TIFF)

d) ảnh sau khi dãn độ tương phản

Hình 4.3. Ảnh gốc và ảnh kết quả sau khi dãn.

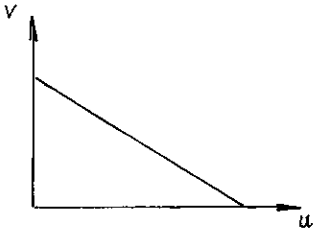
Phân ngưỡng là trường hợp đặc biệt của tách nhiễu khi a = b = const và rõ ràng trong trường hợp này, ảnh đầu ra là ảnh nhị phân (vì chỉ có 2 mức). Phân ngưỡng hay dùng trong kỹ thuật in ảnh 2 màu vì ảnh gần nhị phân không thể cho ra ảnh nhị phân khi quét ảnh bởi có sự xuất hiện của nhiễu do bộ cảm biến và sự biến đổi của nền. Thí dụ như trường hợp ảnh vân tay.



Hình 4.4. Tách nhiễu và phân ngưỡng.

4.1.1.3 Biến đổi âm bản (Digital Negative)

Biến đổi âm bản nhận được khi dùng phép biến đổi f(u) = 255 - u. Biến đổi âm bản rất có ích khi hiện các ảnh y học và trong quá trình tạo các ảnh âm bản.



Hình 4.5. Biến đổi âm bản.

4.1.1.4. Cắt theo mức (Intensity Level Slicing)

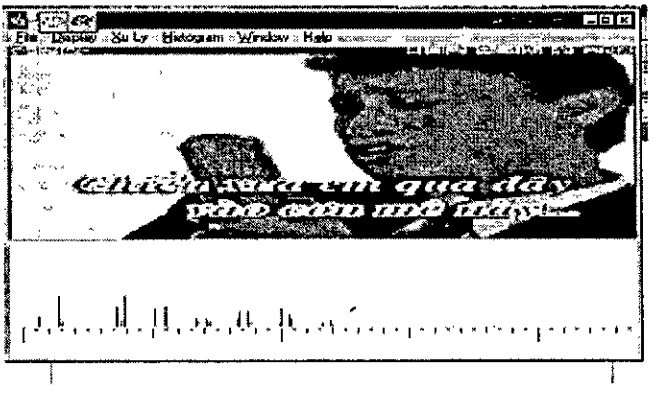
Kỹ thuật này dùng 2 phép ánh xạ khác nhau cho trường hợp có nền và không nền

- **Có nền**

$$f(u) = \begin{cases} L & \text{nếu } a \le u \le b \\ u & \text{khác đi} \end{cases}$$

- **Không nền**

$$f(u) = \begin{cases} L & \text{nếu } a \le u \le b \\ 0 & \text{khác đi} \end{cases}$$

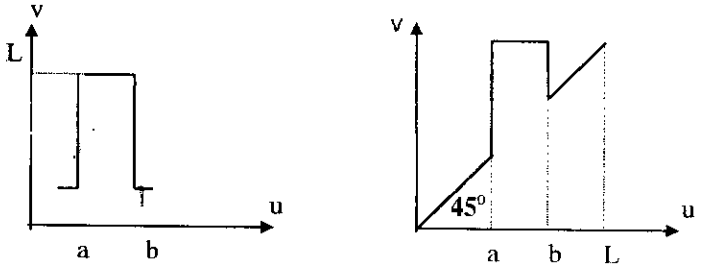


a)Ảnh màu cùng với lược đồ xám. Chỉ số màu cao nhất:243.

b)Ảnh âm bản cùng với lược đồ xám (ứng với phép biến đổi f(x) = L - x).

Chỉ số màu cao nhất:12

Hình 4.6. Ảnh gốc và ảnh âm bản.



a) không nền **b) có nền**

Hình 4.7. Kỹ thuật cắt theo mức.

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

4.1.1.5. Trích chọn bit (Bit Extraction)

Như đã trình bày trên, mỗi điểm ảnh thường được mã hoá 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 hoá 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 nhiều so với bit bậc thấp. Các bit bậc thấp thường biểu diễn nhiễu hay nền. Trong kỹ thuật này, ta có:

$$u = k_1 2^{B-1} + k_2 2^{B-2} + \ldots + k_{B-1} 2 + k_B$$

Nếu ta muốn trích chọn bit có nghĩa nhất: bit thứ n và hiện chúng, ta dùng biến đổi:

$$f(u) = \begin{cases} L & \text{nếu } k_n = 1 \\ 0 & \text{khác đi} \end{cases}$$

và dễ dàng thấy $k_n = i_n - 2\, i_{n-1}$ với i_n cho ở bảng trên.

4.1.1.6. 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.

4.1.1.7. 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 lôga sau:

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

với c là hằng số tỉ lệ, δ là rất nhỏ so với u(m,n). Thường δ chọn cỡ 10^{-3}.

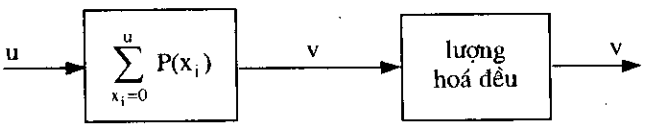
4.1.1.8. Mô hình hoá và biến đổi lược đồ xám

Về ý nghĩa của lược đồ xám và một số phép biến đổi lược đồ đã được trình bày trong chương 3 (phần 3.4). Ở đây, ta xét đến một số biến đổi hay dùng:

- $$f(u) = \sum_{x_i=0}^{u} p_u(x_i) \qquad (4.1)$$

$$\text{với } p_u(x_i) = \frac{h(x_i)}{\sum_{i=0}^{L-1} h(x_i)} \quad i = 0, 1, ..., L-1 \qquad (4.2)$$

$h(x_i)$ là lược đồ mức xám x_i: có nghĩa là số điểm ảnh có mức xám x_i. Trong biến đổi này, u là mức xám đầu vào; còn đầu ra sẽ được lượng hoá đều theo sơ đồ:



Biến đổi này được dùng trong sau bằng lược đồ.

• Ngoài biến đổi như 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:

$$f(u) = \frac{\sum_{x_i=0}^{u} p_u^{1/n}(x_i)}{\sum_{x_i=0}^{L-1} p_u^{1/n}(x_i)} \qquad \text{với } n=2, 3, ... \qquad (4.3)$$

$$f(u) = \log(1+u) \quad u \geq 0 \qquad (4.4)$$

$$(u) = u^{1/n} \qquad u \geq 0 \quad , n = 2, 3, ... \qquad (4.5)$$

sau đó đầu ra được lượng hoá đều. Ba phép biến đổi này được dùng trong lượng hoá ảnh.

Nhìn chung, các biến đổi lược đồ nhằm biến đổi lược đồ từ một đường không thuần nhất sang một đường đồng nhất để tiện cho việc phân tích ảnh.

4.1.2. 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 nhiễu cần phải loại bỏ nhiễu hay ảnh không sắc nét bị mờ hoặc cần làm rõ các chi tiết như biên. Các toán tử không gian dùng trong kỹ thuật tăng cường ảnh được phân theo 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). Do bản chất của nhiễu là ứng với tần số cao và cơ sở lý thuyết của lọc là bộ lọc chỉ cho tín hiệu có tần số nào đó thông qua (dải tần bộ lọc). Do vậy để lọc nhiễu ta 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, Laplace. Chi tiết và các cách áp dụng được trình bày dưới đây.

Trước khi xem xét chi tiết các kỹ thuật áp dụng, cũng hữu ích khi phân biệt một số các loại nhiễu hay can thiệp trong quá trình xử lý ảnh. Thực tế có 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 ta gọi ảnh quan sát (ảnh thu được) là X_{qs}, ảnh gốc là $X_{gốc}$ và nhiễu là η. Ảnh thu được có thể biểu diễn bởi:

$$X_{qs'} = X_{gốc} + \eta$$

- *Nhiễu nhân*: nhiễu nhân thường phân bố khắp ảnh. Nếu ta gọi ảnh quan sát (ảnh thu được) là X_{qs}, ảnh gốc là $X_{gốc}$ và nhiễu là η, ảnh thu được có thể biểu diễn bởi:

$$X_{qs'} = X_{gốc} \times \eta$$

- *Nhiễu xung*: nhiễu xung thường gây đột biến tại một số điểm của ảnh.

4.1.2.1. Làm trơn nhiễu bằng lọc tuyến tính: lọc trung bình và lọc dải thông thấp

Vì 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 (homomorphie); với nhiễu xung ta dùng lọc trung vị , giả trung vị, lọc ngoài (outlier).

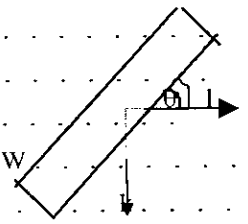
a) Lọc trung bình không gian

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:

$$v(m,n) = \sum_{(k,l)\in W}\sum a(k,l)y(m-k,n-l) \tag{4.6}$$

Nếu trong kỹ thuật lọc trên, ta dùng các trọng số như nhau, phương trình 4-6 trở thành:

$$v(m,n) = \frac{1}{N_w}\sum_{(k,l)\in W}\sum y(m-k,n-l) \tag{4.7}$$



Hình 4.8.

với - y(m,n) : ảnh đầu vào

- v(m,n) : ảnh đầu ra
- w(m,n) : là cửa sổ lọc
- a(k,l) : là trọng số lọc

với $a_{k,l} = \frac{1}{Nw}$ 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, đôi khi người ta ưu tiên cho các hướng để bảo vệ biên của ảnh khỏi bị mờ đi do làm trơn ảnh. Các kiểu mặt nạ như đã liệt kê trong chương trước được sử dụng tuỳ theo các trường hợp khác nhau. Các bộ lọc trên là bộ lọc tuyến tính theo nghĩa là điểm ảnh ở tâm cửa sổ sẽ được thay bởi thế bởi tổ hợp tuyến tính các điểm lân cận chập với mặt nạ.

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

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

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 và phương trình của bộ lọc có dạng:

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

Ở đây, nhân chập H là nhân chập 2*2 và mỗi điểm ảnh kết quả có giá trị bằng trung bình cộng của nó với trung bình cộng của 4 lân cận (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 nhiễu. Về nguyên lý giống như đã trình bày trên. Trong kỹ thuật này người ta hay dùng một số nhân chập sau:

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

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

Ta dễ dàng thấy khi b =1, H_b chính là nhân chập H_1 (lọc trung bình); còn khi b=2, H_b chính là nhân chập H_3 trong phần trước (mục 3.2 chương 3). Để hiểu rõ hơn bản chất khử nhiễu cộng của các bộ lọc này, ta viết lại phương trình thu nhận ảnh dưới dạng:

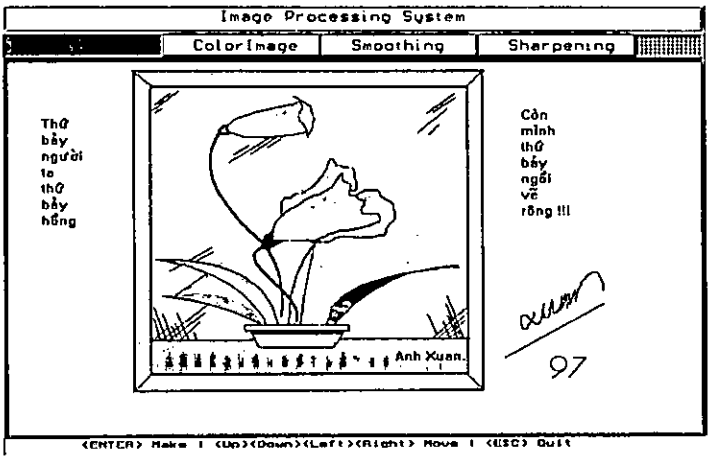
$$X_{qs}[m,n] = X_{gốc}[m,n] + \eta[m,n]$$

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

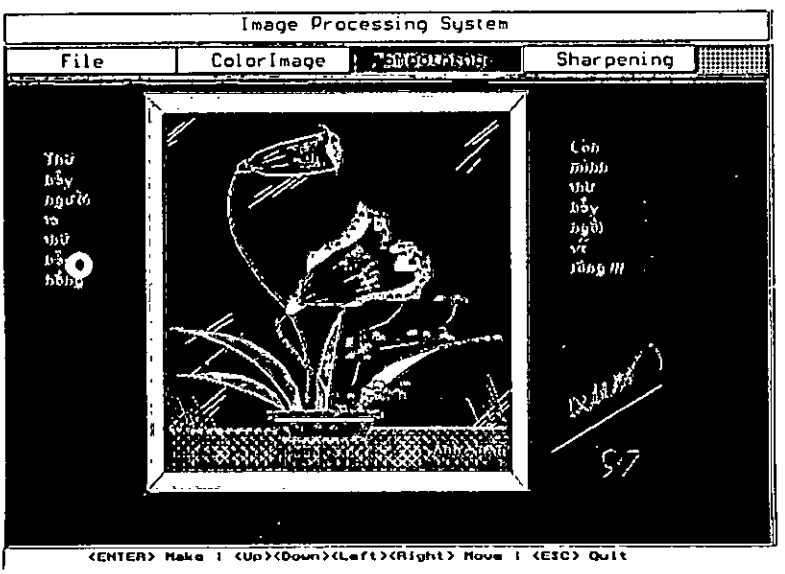
$$Y[m,n] = \frac{1}{N_w}\sum\sum_{(k,l)\in W} x_{gốc}(m-k, n-l) + \eta[\overline{m}, n] \qquad (4.8)$$

hay $$Y[m,n] = \frac{1}{N_w}\sum\sum_{(k,l)\in W} x_{m}(m-k,n-l) + \frac{\sigma_n^2}{N_w} \qquad (4.9)$$

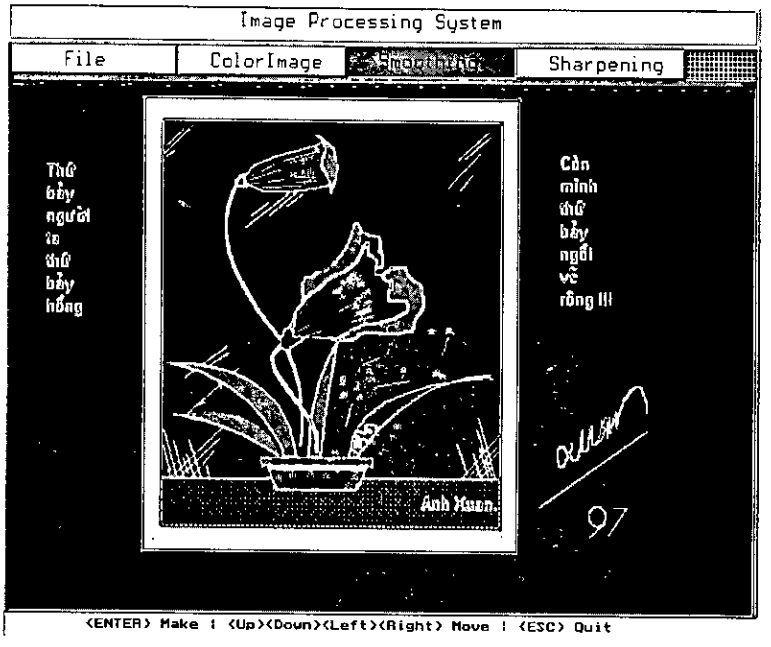
Như vậy nhiễu cộng trong ảnh đã *giảm đi N_w lần*. Hình 4.9 minh hoạ tác dụng cải thiện ảnh bằng lọc thông thấp.



a) Ảnh gốc (chuyển đổi từ ảnh màu sang ảnh mức xám).



b) Ảnh qua lọc trung bình.



c) Ảnh thu được qua lọc thông thấp.

Hình 4.9 Ảnh gốc và ảnh kết quả.

c) Lọc đồng hình (Homomorphie filter)

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

$$X(m,n) = \overline{X}(m,n) \ . \ \eta(m,n)$$

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

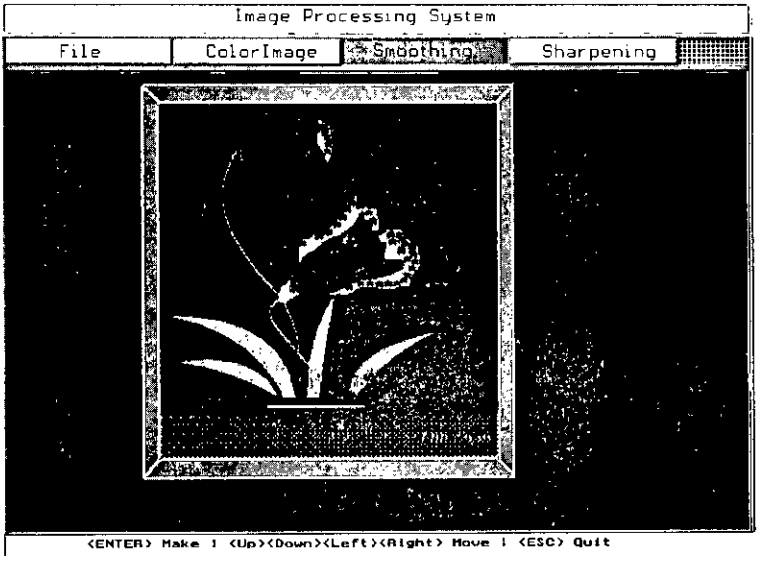
$$\log(X(m,n)) = \log(\overline{X}(m,n)) + \log(\eta(m,n))$$

Rõ ràng là nhiễu nhân có trong ảnh sẽ bị giảm. Sau quá trình lọc tuyến tính ta lại chuyển về ảnh cũ bằng phép biến đổi hàm e mũ. Ảnh thu được qua lọc đồng hình sẽ tốt hơn ảnh gốc.

4.1.2.2. Làm trơn nhiễu bằng lọc phi tuyến

Các bộ lọc phi tuyến cũng hay được dùng trong 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).



Hình 4.9. d) Ảnh qua bằng lọc Homomorphie.

a) Lọc trung vị.

Nhắc lại rằng khái niệm "trung vị" đã nêu trong chương 3 và được viết:

$$v(m,n) = \text{Trungvi}(y(m-k,n-l) \text{ với } (k,l) \in W \tag{4.8}$$

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 3x3, 5x5 hay 7x7. Thí dụ:

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

Thực vậy: mỗi lần ta so sánh một dãy 3 điểm ảnh đầu vào với trung vị, không kể điểm biên. Do đó:

v[0] = 2 < giá trị biên> v[1] = Trungvi(2,3,8) = 3

v[2] = Trungvi(3,8,4) = 4 v[3] = Trungvi(8,4,2) = 4

v[4] = 2 <giá trị biên>

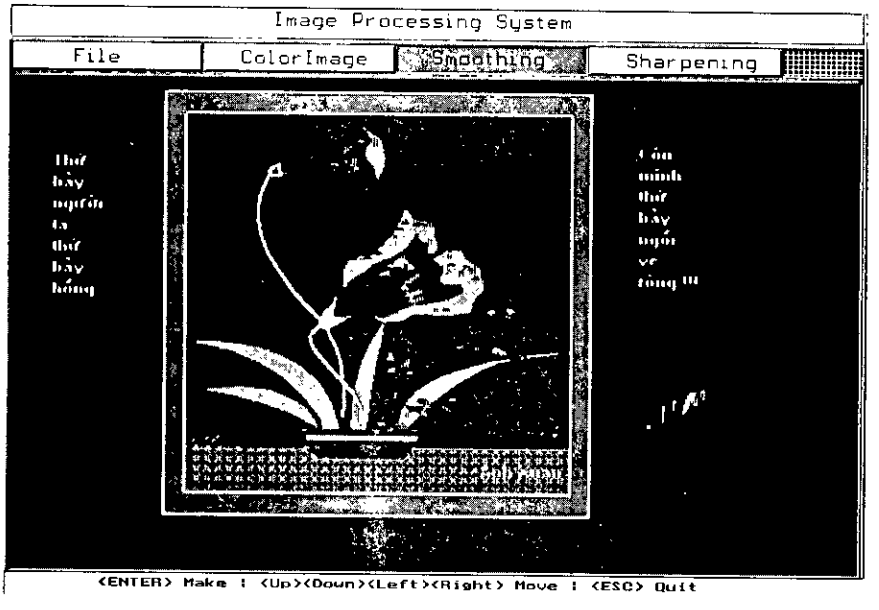
Tính chất của lọc trung vị:

Lọc trung vị là phi tuyến vì:

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

- Hữu ích cho việc loại bỏ các điểm ảnh hay các hàng mà vẫn bảo toàn độ phân giải.

- Hiệu quả giảm khi số điểm nhiễu trong cửa sổ lớn hơn 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, có nghĩa là người ta tiến hành lọc trung vị cho cột tiếp theo cho hàng.



Hình 4.10. Ảnh thu được qua lọc trung vị với ảnh gốc trong 4.9a.

b) Lọc ngoài (Outlier Filter)

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

Bộ lọc ngoài có thể biểu diễn bằng công thức sau:

$$Y(m,n) = \begin{cases} \alpha(w) & \text{nếu} \left|u(m,n) - \alpha(w)\right\rangle \delta\right| \\ u(m,n) & \text{nếu khác} \end{cases}$$

Với $\alpha(w)$ là trung bình cộng các điểm trong lân cận w, δ là ngưỡng ngoài.

Các cửa sổ tính toán thường 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.

4.1.2.3. Mặt nạ gờ sai phân và làm nhấn (Unharp Masking and Crispering)

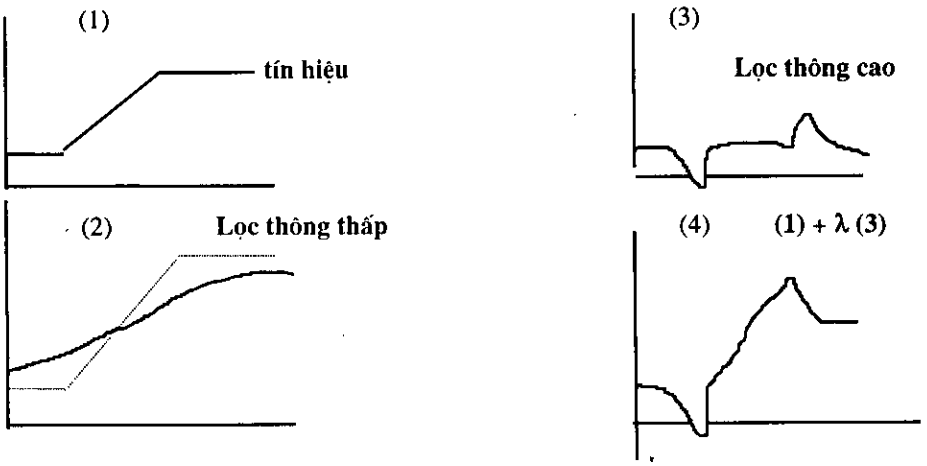
Mặt nạ gờ sai phân dùng khá phổ biến trong công nghệ in ảnh để làm đẹp ảnh. Với kỹ thuật này, tín hiệu đầu ra thu được bằng tín hiệu ra của bộ lọc gradient hay lọc dải cao bổ sung thêm đầu vào:

$$v(m,n) = u(m,n) + \lambda g(m,n) \tag{4.9}$$

với $\lambda > 0$, g(m,n) là gradient tại điểm (m,n). Hàm gradient dùng là hàm Laplace(sẽ trình bày trong chương 5)

$$g(m,n) = u(m,n) - \{u(m-1,n) + u(m+1,n) + u(m,n+1)\}/2 \tag{4.10}$$

Đây chính là mặt nạ chữ thập đã nói trong chương 3.



Hình 4.11. Các toán tử gờ sai phân.

4.1.2.4 Lọc thông thấp, thông cao và lọc dải thông

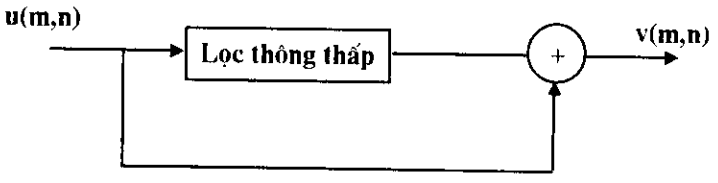
Toàn tử trung bình không gian nói tới trong mục 4.1.2.1 là lọc thông thấp. Nếu

$h_{LP}(m,n)$ biểu diễn bộ lọc thông thấp FIR (Finite Impulse Response) thì bộ lọc thông cao $h_{HP}(m,n)$ có thể được định nghĩa:

$$h_{HP}(m,n) = \delta(m,n) - h_{LP}(m,n) \tag{4.11}$$

Như vậy, bộ lọc thông cao có thể cài đặt một cách đơn giản như trên hình 4.12.

Bộ lọc dải thông có thể định nghĩa như sau: $h_{BP} = h_{L1}(m,n) - h_{L2}(m,n)$ với h_{L1}, h_{L2} là các bộ lọc thông thấp.



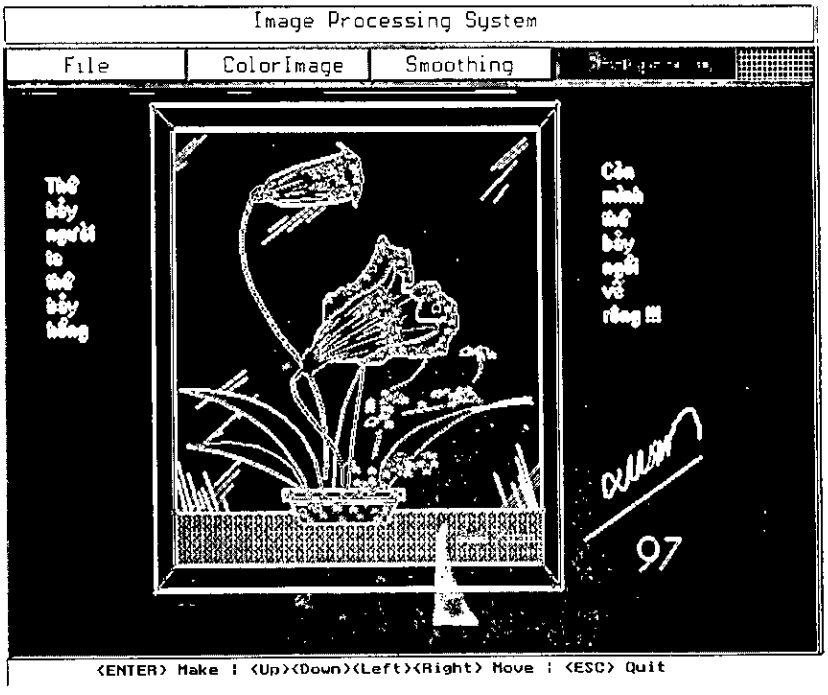
Hình 4.12. Sơ đồ bộ lọc thông cao.

Bộ lọc thông thấp thường dùng làm trơn nhiễu và nội suy. Bộ lọc thông cao dùng 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 chương 5. Tuy nhiên, dễ dàng nhận thấy rằng 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. Do vậy, ta có thể dùng bộ lọc thông cao để cải thiện: lọc các thành phần tần số thấp và chỉ giữ lại 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 4.13. 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).



Hình 4.14. Ảnh qua lọcthông cao (ảnh gốc hình 4.9a).

4.1.2.5. 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

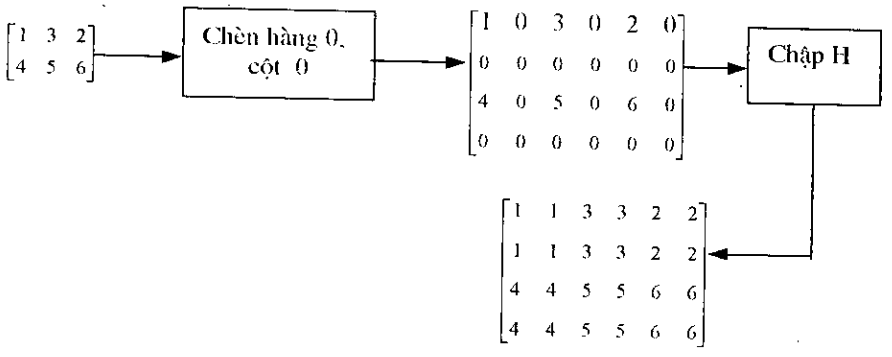
Người ta lấy một vùng của ảnh kích thước M x N và quét theo hàng. Mỗi điểm ảnh nằm trên đường quét sẽ được lặp lại 1 lần và hàng quét cũng được lặp lại 1 lần nữa. Như vậy ta sẽ thu được ảnh với kích thước 2N x 2N. Điều này tương đương với chèn thêm một hàng 0 và một cột 0 rồi chập với mặt nạ H.

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

Kết quả thu được $v(m,n) = u(k,l)$ với $k = \lfloor m/2 \rfloor$ và $l = \lfloor n/2 \rfloor$ (4.13)

Ở đây phép toán $\lfloor . \rfloor$ là phép toán lấy phần nguyên của một số.

Hình 4.15 dưới đây minh hoạ nội suy theo phương pháp lặp:



Hình 4.15. Khuếch đại bởi lặp 2 x 2.

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 khuếch đại 2x2, nội suy tuyến tính theo hàng sẽ tính theo công thức:

$$v_1(m,n) = u(m,n)$$

$$v_1(m,2n+1) = u(m,n) + u(m,n+1) \qquad (4.14)$$

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

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

$$v_1(2m,n) = v_1(m,n)$$

$$v_1(2m+1,n) = v_1(m,n) + v_1(m+1,n) \qquad (4.15)$$

với $0 \le m \le M-1$, $0 \le n \le N-1$.

Nếu dùng mặt nạ:

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

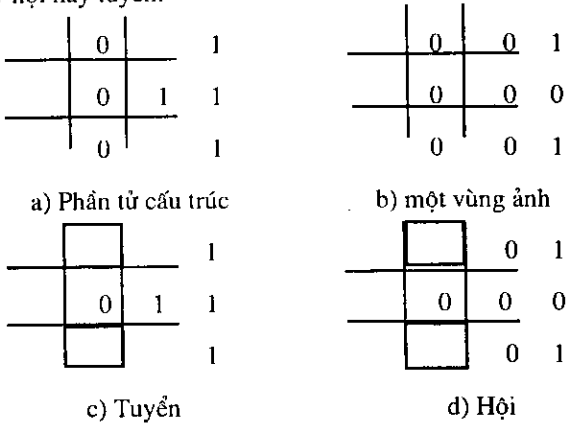
ta cũng thu được kết quả trên.

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 các số 0, rồi p cột các số 0. Cuối cùng, tiến hành nhân chập p lần ảnh với mặt nạ H ở trên [1].

4.1.3. 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 coi một phần tử ảnh như một phần tử lôgíc và có thể áp dụng các toán tử hình học (morphology operators) dựa trên khái niệm biến đổi hình học của một ảnh bởi một phần tử cấu trúc (structural element).

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 môtíp của mặt nạ theo cách lấy hội hay lấy tuyển. Hình 4.16 dưới đây , chỉ ra một phần tử cấu trúc và cách lấy hội hay tuyển:



Hình 4.16. Cải thiện ảnh nhị phân.

Dựa vào nguyên tắc trên, người ta sử dụng 2 kỹ thuật: dãn ảnh (dilatation) và co ảnh (erosion).

4.1.3.1. Dãn ảnh

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

For all pixels I(x,y) do

Begin

. Tính $F_{OR}(x,y)$ {tính or lô gíc }

- if $F_{OR}(x,y)$ then ImaOut(x,y) <--1

else ImaOut(x,y) <- ImaIn(x,y)

End

4.1.3.2. Co ảnh

Co ảnh là thao tác đối ngẫu của dãn ảnh nhằm loại bỏ điểm trắng bị vây bởi các điểm đen. Trong kỹ thuật này, một cửa sổ (N+1) x (N+1) được rê đi khắp ảnh và thực hiện sánh một pixel của ảnh với $(N+1)^2$ -1 điểm lân cận. Sánh ở đây thực hiện bởi phép hội lôgíc. Thuật toán biến đổi được tóm tắt như sau:

For all pixels I(x,y) do

Begin

. Tính $F_{AND}(x,y)$ {Tính và lô gíc}

- if $F_{AND}(x,y)$ then ImaOut(x,y) <--1

else ImaOut(x,y) <- ImaIn(x,y)

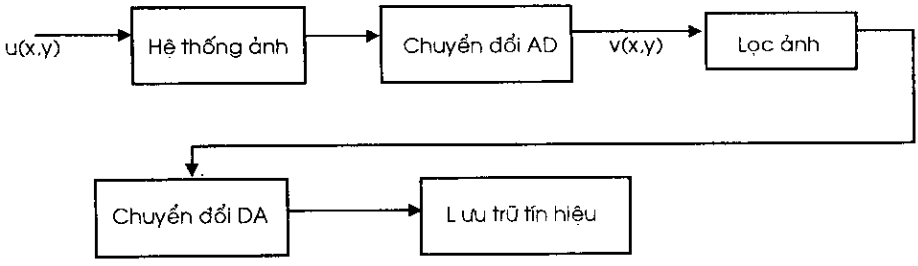
End

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

4.2. KHÔI PHỤC ẢNH (IMAGE RESTORATION)

Khôi phục ảnh đề cập tới các kỹ thuật loại bỏ hay tối thiểu hoá các ảnh hưởng của môi trường bên ngoài hay các hệ thống thu nhận, phát hiện và lưu trữ ảnh đến ảnh thu nhận được. Ở đây, ta có thể liệt kê nguyên nhân các biến dạng (degradations): do nhiễu bộ cảm nhận tín hiệu, ảnh mờ do camera, nhiễu ngẫu nhiên của khí quyển, v...v. Khôi phục ảnh bao gồm nhiều quá trình như: lọc ảnh, khử nhiễu nhằm làm giảm các biến dạng để có thể khôi phục lại ảnh gần giống ảnh gốc tuỳ theo các nguyên nhân gây ra biến dạng. Một hệ thống khôi phục ảnh số có thể minh hoạ như hình 4.17.

Về nguyên tắc, khôi phục ảnh nhằm xác định mô hình toán học của quá trình đã gây ra biến dạng, tiếp theo là dùng ánh xạ ngược để xác định lại ảnh. Việc xác định mô hình có thể thực hiện theo 2 hướng: trước và sau.



Hình 4.17. Hệ thống khôi phục ảnh số.

Theo hướng thứ nhất, một mô hình sẽ được xây dựng từ các ảnh kiểm nghiệm để xác định đáp ứng xung của hệ thống nhiễu.

Theo hướng thứ hai, người ta thực hiện các phép đo trên ảnh. Nói chung là mô hình không biết trước. Các mô hình toán học dùng cho cả hai phương pháp là rất phức tạp.

Mô hình khôi phục ảnh	Lọc tuyến tính	Các phương pháp khác
. Mô hình tạo ảnh	. Lọc ngược	. Entropy cực đại
. Mô hình gây nhiễu	. Đáp ứng xung hữu hạn FIR	. Mô hình Bayesian
. Mô hình quan sát mẫu		. Giải chập ,...

Hình 4.18. Các kỹ thuật khôi phục ảnh.

4.2.1. Các mô hình quan sát và tạo ảnh

Như đã nêu trên, quá trình gây ra biến dạng ảnh gốc phụ thuộc vào hệ thống quan sát và tạo ảnh. Do vậy, trước hết ta cần xem ảnh quan sát được biểu diễn thế nào, trên cơ sở đó mô hình hoá nhiễu sinh ra. Tiếp theo là dùng biến đổi ngược - chính là lọc ngược để khử nhiễu và thu lấy ảnh gốc. Đó là cơ sở lý thuyết của kỹ thuật khôi phục ảnh.

Lưu ý rằng đây là quá trình ngược: Từ tín hiệu quan sát được gồm tín hiệu vào (ảnh gốc) và các biến dạng (nhiễu). Nếu biết tín hiệu ra thường là ảnh thu nhận được qua hệ thống ảnh (xem chương 2), biết các loại tác động (phụ thuộc vào hệ thống và thiết bị) ta suy ra ảnh gốc.

Nếu gọi:

- v(x,y) là ảnh thu nhận được,

- $\eta(x,y)$ là nhiễu,

- $u(x,y)$ ảnh gốc chưa biết,

- f, g là các biến đổi nói chung là phi tuyến đặc trưng cho cơ chế phát hiện và lưu ảnh, ta có mô hình sau:

$$v(x,y) = g[w(x,y)] + \eta(x,y) \tag{4.16}$$

$$w(x,y) = \iint_{-\infty}^{\infty} h(x,y;x',y')u(x',y')dx'dy' \tag{4.17}$$

$$\eta(x,y) = f[g(w(x,y))]\ \eta_1(x,y) + \eta_2(x,y) \tag{4.18}$$

với:

- $h(x,y)$ là đáp ứng xung tại điểm (x,y) như đã nêu trong chương 3,

- $w(x,y)$ là tín hiệu đầu ra của hệ thống tuyến tính với đáp ứng xung h.

Nhiễu gồm 2 thành phần:

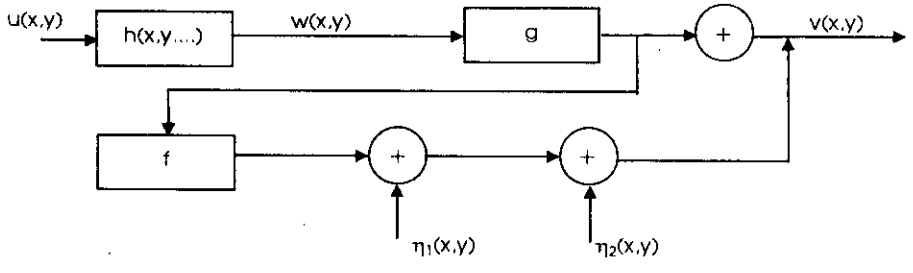
- Thành phần nhiễu phụ thuộc kiểu thiết bị quan sát và tạo ảnh $\eta_1(x,y)$,

- Thành phần nhiễu ngẫu nhiên độc lập $\eta_2(x,y)$.

Mô hình quan sát ảnh trên được thể hiện trên hình 4-19.

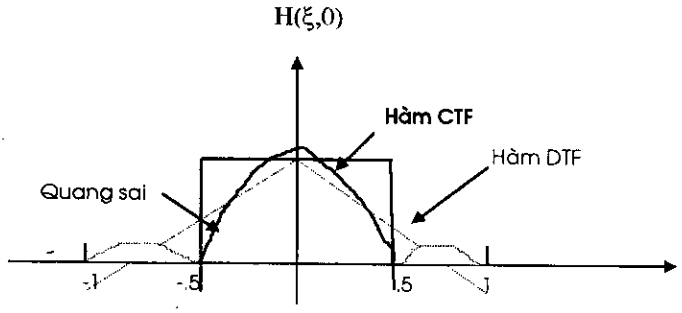
Tuỳ theo hệ thống, người ta có thể liệt kê một số biến dạng trong quá trình thu nhận ảnh:

- Sự vặn pha trong hàm truyền CTF (Coherent Transfer Function) và gọi là *quang sai* (aberration).



Hình 4.19. Mô hình hệ thống quan sát ảnh.

Hình 4.20 dưới đây cho ta thấy sự quang sai của một hệ thống quang học với ống kính vuông.



Hình 4.20. Sự biến dạng do nhiễu loạn và quang sai.

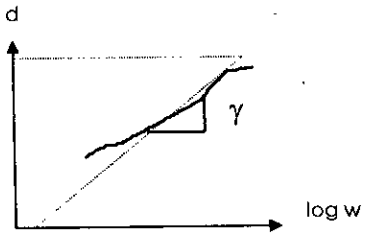
- Rung động mờ xảy ra khi có rung động tương đối giữa đối tượng và ống kính thu trong quá trình thu nhận ảnh.

- Sự nhiễu loạn ngẫu nhiên của môi trường xung quanh đối tượng và hệ thống ảnh (đối tượng ảnh thiên văn).

Chúng ta biết rằng, sự đáp ứng của hệ thống phát hiện và lưu ảnh thường là không tuyến tính. Trong phim ảnh, máy quét ảnh hay thiết bị hiện ảnh, sự đáp ứng được biểu diễn bởi công thức :

$$g = \alpha w^{\beta} \tag{4.19}$$

trong đó α, β là các hằng số phụ thuộc thiết bị; w là biến đầu vào. Thí dụ trong trường hợp máy ảnh, người ta hay dùng mô hình $d = \gamma \log w - d_0$ (4.20)



Hình 4.21. Mô hình đáp ứng tín hiệu ảnh

với γ là hệ số phim, w biểu diễn độ sáng tối và d gọi là mật độ quang học.

Mô hình nhiễu

Phương trình 4-18 cho ta một mô hình chung của nhiễu xuất hiện trong nhiều tình huống. Tuy nhiên, trong một số hệ thống ta có thể biểu diễn nó một cách tường minh hơn. Thí dụ, với một hệ thống quang điện, nhiễu trong chùm tia điện tử thường được biểu diễn bởi:

$$\eta(x,y) = \sqrt{g(x,y)}\,\eta_1(x,y) + \eta_2(x,y) \tag{4.21}$$

trong đó: g cho bởi 4-19; η_1 và η_2 là nhiễu trắng Gauss độc lập tương hỗ với trung bình 0.

Thành phần nhiễu phụ thuộc thiết bị η_1 tăng lên là do quá trình phát hiện và lưu ảnh kéo theo sự truyền điện tử ngẫu nhiên. Sự truyền điện tử ngẫu nhiên này có thể biểu diễn bằng phân bố Poisson có trung bình g. Trong một số trường hợp phân bố này tiệm cận đến phân bố Gauss. Vì phân bố Poisson có trung bình và độ sai lệch là như nhau, do đó thành phần phụ thuộc có phương sai là $\sqrt{g}$ nếu η_1 có độ sai lệch là đơn vị. Thành phần độc lập η_2 biểu diễn nhiễu do nhiệt và có thể mô hình hoá theo kiểu nhiễu trắng. Trong một số hệ thống không có nhiễu do nhiệt như hệ thống phim, mô hình nhiễu có thể viết:

$$\eta(x,y) = \sqrt{g(x,y)}\,\eta_1(x,y) \tag{4.22}$$

Một mô hình khác dành cho nhiễu hạt trong phim là:

$$\eta(x,y) = \varepsilon(g(x,y))^2\eta_1(x,y) \tag{4.23}$$

với ε là hệ số chuẩn hoá, $v \in [1/3, 1/2]$.

Nói chung, thành phần nhiễu phụ thuộc $\eta_1(x,y)$ gây rất nhiều khó khăn cho các thuật toán khôi phục ảnh. Do vậy người ta hay dùng trung bình không gian μ_w thay cho w trong $f[g(x,y)]$ và 4-18 trở thành:

$$\eta(x,y) = f[g(\mu_w)]\eta_1(x,y) + \eta_2(x,y) \tag{4.24}$$

và $\eta(x,y)$ trở thành mô hình nhiễu trắng Gauss. Nếu sự phát hiện ảnh thực hiện trên một miền tuyến tính với thiết bị quang điện, mô hình quan sát tuyến tính có dạng:

$$v(x,y) = w(x,y) + \sqrt{\mu_w}\,\eta_1(x,y) + \eta_2(x,y) \tag{4.25}$$

với máy ảnh $(\gamma=-1)$ ta có $v(x,y) = -\log w + \alpha\eta_1(x,y)$ (4.26)

Ngoài các biểu diễn trên, trong các hệ thống ảnh kết cố còn xuất hiện một loại nhiễu khác gọi là nhiễu đốm (specke noise). Với các đối tượng có độ phân giải thấp nó tăng lên gấp bội và xảy ra nếu bề mặt đối tượng có độ lồi lõm bậc bước sóng:

$$v(x,y) = u(x,y)s(x,y) + \eta(x,y) \tag{4.27}$$

trong đó: s(x,y) là cường độ nhiễu đốm - nó là trường nhiễu trắng ngẫu nhiên có mật độ hàm mũ:

$$s(x,y) = \begin{cases} \frac{1}{\sigma^2}\exp(\frac{-\xi}{\sigma^2}) & \text{với } \xi \geq 0 \\ p_1(\xi) = 0 & \text{khác đi} \end{cases} \quad (4.28)$$

Trong một số trường hợp mẫu hoá đều, mô hình cho bởi 4-16 đến 4-18 có thể thành một xấp xỉ rời rạc:

$$v(x,y) = g[w(x,y) + \eta(x,y) \quad (4.29)$$

$$w(x,y) = \sum_{(k,l)\in W} h(m,n;k,l)u(k,l) \quad (4.30)$$

$$\eta(x,y) = f[g(w(x,y))]\ \eta_1(x,y) + \eta_2(x,y) \quad (4.31)$$

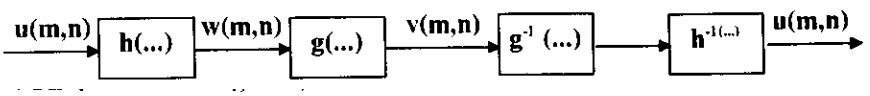
Sau khi đã nghiên cứu các mô hình thu nhận ảnh để xác định các biến dạng, tiếp theo ta dùng các bộ lọc ngược để khôi phục ảnh gốc. Có hai kỹ thuật chính là lọc tuyến tính và lọc phi tuyến. Các kỹ thuật lọc này đã trình bày kỹ trong chương 3. Sau đây ta chỉ xem xét riêng một số kỹ thuật lọc dùng trong khôi phục ảnh.

4.2.2. Kỹ thuật lọc tuyến tính

Kỹ thuật lọc tuyến tính gồm nhiều loại: lọc ngược, lọc giả ngược, lọc Wiener, làm trơn bằng kỹ thuật Spline, lọc sai số bình phương nhỏ nhất có điều kiện, v.v. Các kỹ thuật này sẽ được mô tả dưới đây.

4.2.2.1. Kỹ thuật lọc ngược (Inverse filter)

Lọc ngược là kỹ thuật lọc khôi phục đầu vào của một hệ thống khi biết đầu ra (ảnh thu được hay ảnh quan sát). Để đơn giản, ta giả thiết rằng hệ thống không có nhiễu và việc khôi phục u(x,y) được dựa vào v(x,y). Quá trình đó được mô hình hoá như sau:



Hình 4.22. Mô hình lọc ngược.

Với một hệ thống như thế ta có:

$$g^T(x) = g^{-1}(x); \qquad g^{-1}[g(x)] = x \quad (4.32)$$

và

$$h^T(x,y;k,l) = h^{-1}(x,y;k,l) \quad (4.33)$$

nghĩa là:

$$\sum_{k,l=-\infty}^{\infty} h^T(x,y;k',l')\, h(k',l';k,l) = \delta(x-k,y-l) \tag{4.34}$$

Lọc ngược rất có ích cho quá trình tiền hiệu chỉnh tín hiệu vào trước những biến dạng gây nên bởi hệ thống. Việc thiết kế một bộ lọc ngược là rất khó khăn vì nó không ổn định, do vậy ta có thể dùng biến đổi Fourier 2 vế của 4.34:

$$H^T(w1,w2)H(w1,w2) = 1 \tag{4.35}$$

do đó $H^T(w1,w2) = 1/H(w1,w2)$

Chú ý rằng $H^T(w1,w2)$ không phải luôn luôn tồn tại vì H(w1,w2) có thể nhận giá trị 0. Đây chính là nhược điểm lớn của kỹ thuật lọc ngược.

4.2.2.2. Lọc giả ngược (Pseudoinverse Filter)

Do nhược điểm của lọc ngược là không ổn định (vì H^T có thể không tồn tại), người ta nghĩ đến cách cải tiến nó. Điều đơn giản là làm sao cho H^T luôn tồn tại. Bộ lọc giả ngược được định nghĩa:

$$H^T(w1,w2) = \begin{cases} 1/H(w1,w2) & \text{nếu } H \neq 0 \\ 0 & H=0 \end{cases} \tag{4.36}$$

Trong thực tế, người ta coi H^T là 0 khi $|H|$ nhỏ hơn một lượng ε cho trước ($\varepsilon>0$).

4.2.2.3. Lọc Wiener

Lọc ngược và giả ngược có một yếu điểm là nhậy cảm với nhiễu. Vì thế khi áp dụng kiểu lọc này ta giả định là hệ thống lý tưởng không có nhiễu. Song trên thực tế điều này là không có. Do vậy, người ta nghĩ đến dùng kỹ thuật khác dùng cho các hệ thống có nhiễu gọi là lọc Wiener.

Gọi u(m,n) và v(m,n) là các chuỗi ngẫu nhiên bất kỳ, có trung bình 0. Người ta muốn tìm một xấp xỉ û (m,n) của u(m,n) sao cho sai số trung bình bình phương là cực tiểu.

Gọi $$\sigma_e^2 = E\{[u(m,n) - \hat{u}(m,n)]^2\} \tag{4.37}$$

là sai số trung bình bình phương khi xấp xỉ u(m,n) bởi û(m,n). Giá trị tốt nhất của xấp xỉ û(m,n) được biết khi trung bình có điều kiện của u(m,n) cho bởi v(m,n) với mỗi cặp (m,n), có nghĩa là:

$$\hat{u}(m,n) = E\{[u(m,n)/v(k,l)]\forall\, k,l\} \tag{4.38}$$

Hình 4.23. Lọc ngược và giả ngược.

Nhìn chung 4-37 là rất khó giải vì không tuyến tính. Người ta nghĩ đến sử dụng một dạng tuyến tính khác của xấp xỉ û:

$$\hat{u}(m,n) = \sum_{k,l=-\infty}^{\infty} g(m,n;k,l))v(k,l) \tag{4.39}$$

với g là đáp ứng xung được xác định sao cho sai số trung bình bình phương của 4-37 là cực tiểu.

Nếu giả thiết thêm rằng u, v là các chuỗi Gauss cùng nhau, thì lời giải của 4-37 là tuyến tính. Việc cực tiểu hoá 4-37 yêu cầu điều kiện:

$$\forall\ (m,n),(m',n')\quad E\{[u(m,n) - \hat{u}(m,n]^2\}\ v(m',n') \tag{4.40}$$

Sử dụng định nghĩa của hiệp biến chéo (cross-covariance):

$$r_{a,b}(m,n;k,l) = E[a(m,n)b(k,l)] \tag{4.41}$$

cho 2 chuỗi ngẫu nhiên bất kỳ và cho 4-39, ràng buộc 4-40 trở thành:

$$\sum\sum_{k,l=-\infty}^{\infty} g(m,n;k,l)r_{uv}(k,l;m',n') = r_{uv}(m,n;m',n') \tag{4.42}$$

Phương trình 4-37 và 4-42 là phương trình của *bộ lọc Wiener*.

Nếu u và v là dừng cùng nhau thì: $r_{uv}(m,n;m',n') = r_{uv}(m-m';n-,n')$ (4.43)

Điều này cho phép đơn giản hoá g thành bộ lọc bất biến không gian và nếu ký hiệu bởi g(m-k,n-l) thì 4-42 trở thành:

$$\sum\sum_{k,l=-\infty}^{\infty} g(m-k,n-l)r_{vv}(k,l) = r_{uv}(m,n) \quad (4.44)$$

Biến đổi Fourier cho 2 vế của 4-44 ta có:

$$G(w1,w2) = S_{uv}(w1,w2)S_{vv}^{-1}(w1,w2) \quad (4.45)$$

với G là biến đổi Fourier của g, S_{uv} là biến đổi của r_{uv}, và S_{vv} là biến đổi của r_{vv}. Phương trình trên gọi là đáp ứng tần số của bộ lọc Wiener và phương trình lọc trở thành:

$$\hat{u}(m,n) = \sum_{k,l=-\infty}^{\infty} g(m,n;k,l))v(k,l) \quad (4.46)$$

$$\hat{u}(m,n) = G(w1,w2)V(w1,w2) \quad (4.47)$$

và

$$v(m,n) = \sum\sum_{k,l=-\infty}^{\infty} h(m-k,n-l)u(k,l) + \eta(x,y) \quad (4.48)$$

4.2.2.4. Lọc Wiener với đáp ứng xung hữu hạn FIR(Finite Impulse Response)

Về lý thuyết, bộ lọc Wiener có đáp ứng xung vô hạn và do đó đòi hỏi DFT có kích thước lớn. Tuy nhiên, đáp ứng xung có hiệu quả chỉ là một phần nhỏ của kích thước đối tượng.

Nói chung việc thiết kế một FIR tối ưu là khá phức tạp. Người ta cài đặt một FIR như là tích chập của một bộ lọc có trọng số g, làm cực tiểu sai số trung bình bình phương E với v(m,n):

$$\hat{u}(m,n) = \sum_{k,l\in W} g(k,l)v(m-k,n-l) \quad (4.49)$$

W là cửa sổ từ -M đến M: $(k,l) \in W$ có nghĩa là $-M \le (k,l) \le M$

Ràng buộc trực giao của 4-48 định nghĩa bởi:

$$\forall\ (k,l) \in W:\ E[u(m,n) - \hat{u}(m,n)]v(m-k,n-l) = 0 \quad (4.50)$$

sẽ làm giảm số phương trình xuống còn $(2M+1)^2$

$$\forall\ (k,l) \in W:\ r_{vv}(m,n) - \sum_{k,l \in W} g(k,l) r_{uv}(m-k,n-l) = 0 \tag{4.51}$$

(kết quả này suy ra từ 4-40, 4-41 và 4-42).

Áp dụng 4-47 và giả thiết thêm rằng $\eta(m,n)$ là nhiễu trắng có trung bình 0 và độ lệch σ^2, ta có:

$$r_{vv}(k,l) = r_{uu}(k,l) \otimes u(k,l) + \sigma^2 \alpha(k,l) \tag{4.52}$$

$$\alpha(k,l) = h(k,l) * h(k,l) = \sum_{i,j=-\infty}^{\infty} \sum h(i,j)h(i+k,j+l) \tag{4.53}$$

$$r_{uv}(k,l) = h(k,l) * r_{uu}(k,l) = \sum_{i,j=-\infty}^{\infty} \sum h(i,j) r_{uu}(i+k,j+l) \tag{4.54}$$

Định nghĩa hàm tương quan:

$$r_0(k,l) = \frac{r_{uu}(k,l)}{r_{uu}(0,0)} = \frac{r_{uu}(k,l)}{\sigma^2} \tag{4.55}$$

Tuỳ theo các hệ thống và nhu cầu khôi phục ảnh, người ta còn sử dụng nhiều biến đổi khác cho bộ lọc Wiener [Anil.K.Jain trang 236-294].

4.2.2.5. Kỹ thuật làm trơn spline và nội suy

Kỹ thuật spline là dùng một đường cong để xấp xỉ một hàm liên tục từ các giá trị mẫu (giá trị quan sát được) trên một lưới. Trong các quá trình xử lý ảnh, hàm spline được dùng để khuếch đại ảnh, làm trơn nhiễu. Trước tiên, ta xử lý mỗi điểm ảnh theo hàng ngang và khuếch đại ngang cho hàng, tiếp theo áp dụng chính thủ tục này cho cột. Như vậy, ảnh sẽ làm trơn và nội suy bởi một hàm tách được (theo nghĩa thực hiện riêng cho từng chiều).

Gọi y_i là một chuỗi các giá trị quan sát được của một hàm liên tục mẫu đều trên khoảng [0,N], với i =0, 1, 2, . . ., N:

$$x_i = x_0 + ih \quad h > 0$$

và

$$y_i = f(x_i) + \eta(x_i) \tag{4.56}$$

với $\eta(x_i)$ biểu diễn sai số của quá trình quan sát.

Đường spline điều chỉnh một hàm trơn g(x) trên toàn bộ tập giá trị quan sát được mà độ gồ ghề của nó do bởi phổ năng lượng trên khoảng [0,N] sao cho sai số là cực tiểu. Hơn nữa, nếu sai số bình phương nhỏ nhất tại các điểm quan sát là bị chặn, có nghĩa là:

$$g_i = g(x_i) \text{ và } F = \sum_{i=0}^{N} \frac{g_i - y_i}{\sigma^n} \le S \tag{4.57}$$

Nếu S=0, có nghĩa là đường spline trùng hoàn toàn với đường cong biểu diễn các điểm quan sát được. Đặc biệt, nếu σ^2 là giá trị trung bình bình phương của nhiễu, S được chọn nằm trong khoảng (N-1) = $\sqrt{2}$ (N+1). Khoảng này gọi là *khoảng tin cậy* của S. Phụ thuộc vào giá trị của S, ta có 2 lời giải:

- Nếu S là khá lớn và 4-57 thoả (xấp xỉ) bởi 1 đường thẳng:

$$g(x) = a + bx \qquad x_0 \le x \le x_N$$

$$b = \frac{\mu_{xy} - \mu_x\mu_y}{\mu_{xx} - \mu_x^2}, \qquad a = \mu_x - \mu_y \tag{4.58}$$

μ biểu diễn giá trị trung bình của mẫu, thí dụ $\mu_x = \dfrac{\sum_{i=0}^{N} x_i}{(N+1)}$

- Ràng buộc 4-57 là chặt chẽ và chỉ có một ràng buộc đẳng thức có thể thoả mãn. Lời giải sẽ là một đường spline bậc 3 định nghĩa bởi:

$$g(x) = a_i + b_i(x-x_i) - c_i(x-x_i)^2 + d_i(x-x_i)^3 \qquad x_i \le x \le x_{i+1} \tag{4.59}$$

Các hệ số a_i, b_i, c_i và d_i là nghiệm của phương trình:

$$\begin{cases} (P + \lambda Q)c = \lambda v \\ v = L^T y \quad c_0 = c_N = 0 \\ d_i = (c_{i+1} - c_i)/3h \qquad 0 \le i \le N-1,\ d_0 = d_N = 0 \\ b_i = \dfrac{a_{i+1} - a_i}{h} - hc_i - h^2 d_i \qquad b_N = 0,\ 0 \le i \le N-1 \end{cases} \tag{4.60}$$

với:

- a, y và d là các véctơ có N+1 phần tử;
- c là véctơ có N-1 phần tử (i=1, 2, ..., N-1);

$$Q = h/3 \begin{bmatrix} 4 & 1 & & 0 \\ 1 & \ddots & \ddots & 1 \\ & \ddots & \ddots & 4 \\ 0 & & 1 & \end{bmatrix} \qquad L = 1/h \begin{bmatrix} 1 & & & 0 \\ -2 & \ddots & & \\ 1 & \ddots & \ddots & \\ 0 & & -2 & 1 \end{bmatrix}$$

- $P = \delta^2 L^T L$

- Q là ma trận 3 đường chéo Toeplitz: (N-1)x(N-1);

- L là ma trận tam giác dưới (N+1)x(N-1).

Thí dụ: Cho dãy giá trị quan sát $y_i = \{1, 3, 4, 2, 1\}$

- h=1 và δ=1

- N=4 và $x_0=0$, $x_n=4$ (gồm 5 điểm)

- nội suy tuyến tính là đường thẳng

- $\mu_x = 2$, $\mu_y = 2.2$, $\mu_{xy} = 4.2$ và $\mu_{xx} = 6$

Với các giá trị trên, ta tính được b = -1, a = 2,4 và g = 2,4-0,1x.

Sai số bình phương nhỏ nhất $\sum_{i=0}^{N} (y_i - g_i)^2 = 6,7$

Khoảng tin cậy cho S là [1,8; 8,16]. Tuy nhiên nếu chọn S =5 chúng ta sẽ có đường xấp xỉ là đường spline bậc 3. Với số liệu trên, ta dễ dàng tính được:

$$P = L^T L = \begin{pmatrix} 6 & -4 & 1 \\ -4 & 6 & -4 \\ 1 & -4 & 6 \end{pmatrix} \quad V = L^T y = \begin{pmatrix} -1 \\ -3 \\ 1 \end{pmatrix}$$

Giải phương trình F - S = 0 với S = 5 ta thu được x = 0,0274. Giải phương trình 4.60 ta thu được các véctơ a, b, c và d. Giá trị sai số nhỏ nhất kiểm nghiệm lại là 4,998 ≈ 5. Như vậy chấp nhận được.

$$a = \begin{pmatrix} 2,199 \\ 2,397 \\ 2,415 \\ 2,180 \\ 1,808 \end{pmatrix} \quad b = \begin{pmatrix} 0,198 \\ 0,157 \\ 0,194 \\ -0,357 \\ 0,000 \end{pmatrix} \quad c = \begin{pmatrix} 0,000 \\ -0,033 \\ -0,040 \\ -0,022 \\ 0,000 \end{pmatrix} \quad d = \begin{pmatrix} 0,000 \\ -0,005 \\ 0,009 \\ 0,007 \\ 0,000 \end{pmatrix}$$

4.2.3. Kỹ thuật lọc phi tuyến trong khôi phục ảnh

Một số kỹ thuật lọc phi tuyến đã được mô tả trong chương 3, nhưng ở đây sẽ đưa thêm kỹ thuật lọc đồng hình để khử nhiễu đốm, kỹ thuật Entropy cực đại, giải chập mù, mô hình Bayesian, v.v.

4.2.3.1. Lọc nhiễu đốm

Như đã nói ở trên, nhiễu đốm nảy sinh khi các tia đơn sắc được tán xạ từ bề mặt nhám mà độ gồ ghề bằng bước sóng. Trong không gian tự do, nhiễu đốm có thể coi như tổng vô hạn các pha đồng nhất, độc lập mà pha và biên độ là ngẫu nhiên. Như vậy ta có thể biểu diễn:

$$a(x,y) = a_R(x,y) + ja_I(x,y) \tag{4.61}$$

với a_R, a_I là các biến ngẫu nhiên độc lập theo phân bố Gauss, trung bình 0 cho mỗi cặp (x,y) với độ sai lệch σ^2. Trường cường độ S:

$$s(x,y) = |a(x,y)| = a_R^2 + a_I^2 \tag{4.62}$$

có phân bố mũ với lượng sai lệch $\sigma^2 = 2\sigma^2$ và trung bình $\mu_r = E[s] = \sigma^2$. Với một loại nhiễu đốm, tỉ lệ tương phản được định nghĩa:

$$\gamma = \text{phương sai } s/\text{ trung bình } s \tag{4.63}$$

Một phương pháp đơn giản để giảm nhiễu trắng là N-Look, có nghĩa là lấy ảnh ở N thời điểm khác nhau rồi tính trung bình các lần quan sát. Giả sử sự phát hiện nhiễu là thấp và ảnh lần thứ nhất được viết:

$$v_l(x,y) = u(x,y)s_l(x,y) \quad l = 1, 2, ..., N \tag{4.64}$$

Như vậy trung bình tức thời của N quan sát là:

$$\ddot{u}_N(x,y) = \frac{1}{N}\sum_{i=1}^{N} v_i(x,y) = u(x,y)\,\check{s}_N(x,y) \tag{4.65}$$

trong đó: $\check{s}_N(x,y)$ là trung bình N quan sát của trường nhiễu đốm. Đó cũng là ước lượng gần giống nhất của $v_i(x,y)$, i=1,...,N và $E[\ddot{u}_N] = \mu_u/N$; sai lệch = $u^2\mu_u^2/N$. Như vậy $\gamma = 1/N$ cho $\ddot{u}_N$.

Nếu số lượng nhìn N là nhỏ, để cho hiệu quả người ta thực hiện một số kiểu lọc không gian để giảm nhiễu đốm. Một kỹ thuật để lọc nhiễu đốm trong ra đa là tính trung bình giá trị cường độ của các điểm lân cận. Độ tương phản được cải thiện so với phương pháp N-quan sát. Do bản chất của nhiễu đốm, người ta dùng biến đổi logarit của 4.62 và thu được:

$$\log \ddot{u}_N(x,y) = \log u(x,y) + \log \check{s}_N(x,y) \tag{4.66}$$

Nếu ký hiệu $W_N = \log \ddot{u}_N(x,y)$, $Z = \log u(x,y)$ và $\eta_N = \log \dot{s}_N(x,y)$, ta có mô hình quan sát nhiễu phụ $W_N(x,y) = Z(x,y) + \eta_N(x,y)$ (4.67)

với $\eta_N(x,y)$ là nhiễu trắng dừng.

Với $N \geq 2\eta_N$,có thể mô hình hoá bởi trường ngẫu nhiên Gauss mà hàm mật độ phổ được định nghĩa bởi:

$$S_l(\xi_1,\xi_2) = \sigma^2 = \begin{cases} \pi^2/6 & \text{nếu } N = 1 \\ 1/N & \text{nếu } N > 1 \end{cases}$$

Bây giờ $Z(x,y)$ có thể ước lượng dễ dàng từ $W(x,y)$ bởi bộ lọc Wiener.

4.2.3.2. Kỹ thuật entropy cực đại

Người ta nhận thấy rằng: đầu vào, đầu ra và PSF của các hệ thống ảnh không kết cố thường là không âm. Các thuật toán khôi phục ảnh dựa vào trung bình bình phương hay bình phương cực tiểu không có lợi cho ảnh với các giá trị không âm. Phương pháp dựa vào entropy cực đại cho lời giải không âm. Cơ sở lập luận của phương pháp này là ở chỗ: entropy là số đo của một đại lượng không chắc chắn, do vậy nó đặt rất ít giả thiết về lời giải và tạo nên một sự tự do cực đại về các ràng buộc.

Với một ảnh quan sát được $v = \pi x$, với π là ma trận PSF; u, v là các ma trận biểu diễn đối tượng và quan sát. Kỹ thuật entropy cực đại nhằm cực đại:

$$g(u) = -\sum_{0}^{\infty} u(n)\log u(n) \tag{4.68}$$

với ràng buộc $$\frac{1}{2}\|v - \pi u\|^2 = \sigma_\varepsilon^2 > 0 \tag{4.69}$$

Vì $u(n)$ không âm, do vậy ta có thể chuẩn hoá cho $\sum u(n) = 1$. Như vậy ta có thể xử lý như phân bố xác suất mà Entropy là $\varepsilon(n)$. Dùng phương pháp tối ưu của Lagrange, lời giải của phương trình trên cho bởi:

$$\hat{u} = \exp\{-1-\lambda\pi^T(v-\pi\hat{u})\} \tag{4-70}$$

với λ là hằng số Lagrange, l là véctơ.

Một lời giải khác tốt hơn nếu ta định nghĩa các ràng buộc:

$$\begin{cases} u(n) \geq 0 \\ \sum_{j=0}^{N-1} h(m,j)u(j) = v(m), \qquad m = 0,...M-1 \end{cases} \tag{4.71}$$

và lời giải cho bởi:

$$\hat{u}(n) = \frac{1}{e} \exp\left[\sum_{i=0}^{M-1} h(l,n)\chi(l) \right] \quad n = 0, 1, ..., N-1 \tag{4.72}$$

với $\lambda(l)$ là hằng nhân Lagrange.

4.2.3.3. Phương pháp Bayesian

Trong nhiều tình huống ảnh, thí dụ như hệ thống ghi phim, mô hình quan sát là không tuyến tính và có dạng:

$$v = f(\pi u + \eta) \tag{4.73}$$

với f(x) là một hàm không tuyến tính của x và η biểu diễn nhiễu.

Công thức nổi tiếng của Bayes về xác suất có điều kiện cho bởi:

$$p(u \mid v) = p(u)\, p(v \mid u)/p(v) \tag{4.74}$$

Nó rất có ích để xác định nhiều kiểu ước lượng khác nhau cho một véctơ ngẫu nhiên u từ một véctơ quan sát v. Có một số kiểu ước lượng chính sau:

- MMSE: ước lượng trung bình bình phương cực tiểu của u.
- MAP : ước lượng xác suất có điều kiện cực đại $p(u \mid v)$.
- ML : ước lượng gần đúng nhất $p(v \mid u)$.

mà các đối tượng sử dụng là xác suất có điều kiện $p(v \mid u)$ hay $p(u \mid v)$.

Vì rất khó xác định p(v) ngay cả khi u và η là phân bố Gauss, nên người ta hay sử dụng MAP và ML vì nó không đòi hỏi p(v). Nếu giả thiết u và η là phân bố Gauss với hiệp biến R_u và R_v, các ước lượng ML, MAP có thể tính được khi giải các phương trình sau:

$$\hat{u}_{ML}: \quad \pi \mathcal{D} R^{-1}[v - f(\pi \hat{u}_{ML})] = 0 \tag{4.75}$$

với $\mathcal{D}$ là ma trận đường chéo = Diag $\{df(x)/dx$ với $x = wi\}$ (4.76)

w_i là các phần tử của $W = \bar{\pi}\, \hat{u}_{ML}$

và

$$\hat{u}_{MAP} = \mu_v + R_v \pi^T \mathcal{D} R_v^{-1} [v - f(\pi \hat{u}_{MAP})] \tag{4.77}$$

Nếu f(x) là tuyến tính, thí dụ f(x)=x, $R_v = \sigma_v^2$, thì $\hat{u}_{ML}$ là lời giả của phương trình:

$$\pi^T \pi \hat{u}_{ML} = \pi^T v \tag{4.78}$$

và

$$\hat{u}_{MAP} = \mu_v + G(v-\mu_u) \tag{4.79}$$

với

$$G = (R_v^{-1} + \pi^T R_v^{-1} \pi)^{-1} \pi^T R_v^{-1} \tag{4.80}$$

Trong thực tế, μ có thể lấy giá trị là trung bình cục bộ của v và $\mu_v \approx \pi^* f^{-1}(\mu_v)$. π^* là biến đổi ngược của π.

4.2.3.4. Giải chập mù (Blind deconvolution)

Việc khôi phục ảnh khi PSF không biết là một vấn đề khôi phục phi tuyến khó khăn. Với một hệ ảnh bất biến không gian, mật độ phổ năng lượng của ảnh quan sát tuân theo:

$$S_{vv}1,w2) = |H(w1,w2)|^2 S_{uu}(w1,w2) + S_{\eta\eta}(w1,w2)$$

$$S_{uv}(w1,w2) = H^*(w1,w2) S_{uu}(w1,W2)$$

và

$$\log |H|^2 = \log(S_{vv} - S_{\eta\eta}) - \log S_{uv} \tag{4.81}$$

Nếu nhiễu cộng là nhỏ, chúng ta có thể dùng ước lượng:

$$\log |H| = \frac{1}{2M}\sum_{k=1}^{M}\log|vk|2 - \log|uk|2 = \frac{1}{M}\sum_{k=1}^{M}\log|vk| - \log|uk| \tag{4.82}$$

với v_k và u_k là các khối của ma trận biểu diễn v(m,n) và u(m,n) trong biến đổi Fourier (k=1, 2,...,M).

Phương pháp này dựa vào ước lượng năng lượng phổ chưa biết nên có tên gọi là "*giải chập mù*". Trong một số tình huống, pha của H là 0 hay là 1 không quan trọng, lúc đó H biểu diễn trung bình nhiễu loạn môi trường, camera không hội tụ, trễ pha, v.v.

Trong khôi phục ảnh, các mô hình toán là rất nặng nề và phức tạp. Trên đây chỉ đề cập một phần cơ sở lý thuyết và một số kỹ thuật lọc trong khôi phục ảnh. Bạn đọc quan tâm xin tham khảo tài liệu [1].

Bài tập chương 4

Câu 1

Cho ảnh số và các nhân chập sau:

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

1. Ảnh trên có nhiễu không? Đó là loại nhiễu gì?

2. Minh hoạ 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).

Câu 2

1. Viết thủ tục dùng kỹ thuật lọc trung vị sử dụng bộ lọc chữ thập kích thước3 x3 và 5x5. Việc sắp xếp các điểm theo thuật toán tuỳ chọn (chọn đơn giản, chèn tuyến tính hay đổi chỗ).

2. Viết thủ tục cải thiện ảnh dùng kỹ thuật lọc theo mô hình Gauss.

3. Viết thủ tục thực hiện việc dãn ảnh bằng kỹ thuật Dialatation.

4. Viết thủ tục thực hiện việc ăn mòn ảnh bằng kỹ thuật Erosion.