

## CHƯƠNG 1.

### CƠ SỞ XỬ LÝ ẢNH.

#### v MỞ ĐẦU.

Xử lý ảnh số có nhiều ứng dụng thực tế. Một trong những ứng dụng sớm nhất là xử lý ảnh từ nhiệm vụ Ranger 7 tại phòng thí nghiệm Jet Propulsion vào những năm đầu của thập kỷ 60. Hệ thống chụp hình gắn trên tàu vũ trụ có một số hạn chế về kích thước và trọng lượng, do đó ảnh nhận được bị giảm chất lượng như bị mờ, méo hình học và nhiễu nền. Các ảnh đó được xử lý thành công nhờ máy tính số. Hình ảnh của mặt trăng và sao hoả mà chúng ta thấy trong tất cả các tạp chí đều được xử lý bằng những máy tính số.

Ứng dụng của xử lý ảnh có khả năng tác động mạnh mẽ nhất đến cuộc sống của chúng ta là trong lĩnh vực y tế. Soi chụp bằng máy tính dựa trên cơ sở định lý cắt lớp (projection\_slice) sẽ thảo luận trong phần 4.3, được dùng thường xuyên trong xét nghiệm lâm sàng, ví dụ như phát hiện và nhận dạng u não. Những ứng dụng y học khác của xử lý ảnh số gồm cải thiện ảnh X quang và nhận dạng đường biên mạch máu từ những ảnh chụp mạch bằng tia X (angiograms).

Ứng dụng khác, gần gũi hơn với cuộc sống gia đình là cải tiến ảnh tivi. Hình ảnh mà chúng ta thấy trên màn hình tivi có các khuyết tật là độ phân giải hạn chế, bị rung rinh, có ảnh ma, nhiễu nền và trượt hình do đan dòng ở những mức độ khác nhau.

Tivi số không còn xa với thực tế và xử lý ảnh số sẽ có tác động quyết định đến việc cải thiện chất lượng hình ảnh của những hệ truyền hình hiện tại và làm phát triển những hệ truyền hình mới như truyền hình có độ phân giải cao (HDTV).

Một vấn đề chính của truyền thông video như hội nghị video, điện thoại video là cần có có dải tần rộng. Việc mã hoá thẳng chương trình video chất lượng quảng bá yêu cầu đến 100 triệu bit/sec. Nếu hy sinh một phần chất lượng và dùng các sơ đồ mã hoá ảnh số thì có thể đưa ra thị trường những hệ truyền hình chất lượng đủ rõ với nhịp bit chỉ dưới 100 nghìn bit/sec.

Người máy càng ngày càng đóng vai trò quan trọng trong công nghiệp và gia đình. Chúng sẽ thực hiện những công việc rất nhàn chán hoặc nguy hiểm , và những công việc mà tốc độ và độ chính xác vượt quá khả năng của con người . Khi người máy trở nên tinh vi hơn , thị giác máy tính sẽ đóng vai trò ngày càng quan trọng. Người ta sẽ đòi hỏi người máy không những phát hiện và nhận dạng các bộ phận công nghiệp, mà còn “hiểu” được những gì chúng “thấy” và đưa ra những hành động phù hợp. Xử lý ảnh số sẽ có tác động lớn đến thị giác máy tính.

Ngoài những lĩnh vực ứng dụng mọi người đã biết, xử lý ảnh số còn có một số ứng dụng khác ít được nói đến hơn. Người thi hành luật pháp thường chụp hình trong những môi trường không thuận lợi ,và ảnh nhận được thường bị xuống cấp. Ví dụ, bức ảnh chụp vội biển đăng kí xe ô tô đang chạy thường bị nhoè, việc làm giảm độ nhoè là cần thiết trong việc nhận dạng ô tô. Một ứng dụng ít biết khác là nghiên cứu sự di cư của cá voi. Khi người ta nghiên cứu hành vi di cư của sư tử, hổ và các động vật khác, họ bắt các động vật và cột thẻ vào vị trí thuận lợi ở đuôi hoặc tai. Khi bắt được động vật ở nơi khác, thẻ cho biết thông tin về sự di cư của động vật. Tuy nhiên, cá voi rất khó bắt và cột thẻ. May thay, cá voi lại thích để lộ đuôi, mà đuôi của chúng có những đặc điểm có thể giúp để nhận biết chúng. Để nhận dạng một con cá voi, bức ảnh chụp vội đuôi của nó từ trên tàu được so sánh với hàng ngàn ảnh đuôi cá voi khác nhau trong một bộ sưu tập. Quan sát liên tiếp và nhận dạng một cá thể cá voi nào đó ta có thể theo dõi sự di cư của nó. Tuy nhiên, việc so sánh ảnh cực kỳ nhàn chán và phải dùng xử lý ảnh số để tự động hoá công việc.

Những ứng dụng xử lý ảnh số là vô hạn. Ngoài những ứng dụ ng đã thảo luận ở trên, còn bao gồm cả các lĩnh vực khác như điện tử gia đình, thiên văn học, sinh vật học, vật lý, nông nghiệp, địa lý, nhân chủng học, và nhiều lĩnh vực khác. Nhìn và nghe là hai phương tiện quan trọng nhất để con người nhận thức thế giới bên ngoài, do vậy không có gì đáng ngạc nhiên khi mà xử lý ảnh số có nhiều khả năng ứng dụng, không chỉ trong khoa học và kỹ thuật mà cả trong mọi hoạt động khác của con người.

Xử lý ảnh số có thể chia làm bốn lĩnh vực, tuỳ thuộc vào loại công việc. Đó là cải thiện ảnh, phục hồi ảnh, mã hoá ảnh, và lý giải nội dung (understanding) ảnh. Trong cải thiện ảnh, ảnh được xử lý để người xem, như trong truyền hình, hoặc là được xử lý trước để trợ giúp hoạt động của máy móc, như trong nhận dạng đối tượng bởi máy móc. Trong phục hồi ảnh, ảnh bị xuống cấp trong một số trường hợp, chẳng hạn như bị nhoè, và mục đích là để giảm bớt hoặc loại bỏ hẳn ảnh hưởng sự xuống cấp. Phục hồi ảnh có

liên quan mật thiết đến cải thiện ảnh. Khi ảnh bị xuống cấp, việc cải thiện ảnh thường đem lại kết quả làm giảm sự xuống cấp. Tuy nhiên có một số sự khác nhau quan trọng giữa phục hồi ảnh và cải thiện ảnh. Trong phục hồi ảnh, một ảnh lý tưởng bị xuống cấp và mục đích phục hồi là tạo ra ảnh sau xử lý giống như ảnh ban đầu. Trong việc cải thiện ảnh, mục đích cải thiện là làm cho ảnh sau xử lý trông đẹp hơn ảnh chưa được xử lý. Để minh họa sự khác nhau này, hãy lưu ý rằng một ảnh gốc chưa xuống cấp không thể phục chế hơn nữa, nhưng vẫn có thể được cải thiện bằng cách tăng độ nét. Trong mã hóa ảnh, mục đích là biểu diễn ảnh với một số ít bít nhất trong điều kiện chất lượng ảnh và độ rõ chấp nhận được cho từng ứng dụng cụ thể, chẳng hạn như hội nghị video. Mã hóa ảnh liên quan đến cải thiện ảnh và phục hồi ảnh. Nếu có thể cải tiến đáng kể bên ngoài (visual appearance) của ảnh được phục hồi, hoặc làm giảm sự xuống cấp do các nguồn nhiễu, - như nhiễu lượng tử mà thuật toán mã hóa ảnh gây ra, thì ta có thể làm giảm số lượng bít cần thiết để đại diện ảnh ở một mức chất lượng và độ rõ chấp nhận được trong lý giải ảnh(understanding), đầu vào là ảnh, mục đích là diễn đạt nội dung ảnh bằng một hệ ký hiệu nào đó. Những ứng dụng của lý giải ảnh bao gồm thị giác máy tính, kỹ thuật robot và nhận dạng mục tiêu. Lý giải ảnh khác với ba lĩnh vực khác của xử lý ảnh ở một khía cạnh chính. Trong cải tiến, phục hồi và mã hóa ảnh cả đầu vào và đầu ra đều là ảnh ,và khâu xử lý tín hiệu là phần then chốt trong các hệ thống đã thành công trên các lĩnh vực đó. Trong lý giải ảnh , đầu vào là ảnh, nhưng đầu ra thường là một biểu diễn bằng kí hiệu nội dung của ảnh đầu vào. Sự phát triển thành công của các hệ thống trong lĩnh vực này cần đến cả xử lý tín hiệu và những khái niệm trí tuệ nhân tạo.

Trong hệ lý giải ảnh điển hình, xử lý tín hiệu được dùng cho công việc xử lý mức thấp như làm giảm sự xuống cấp và trích ra các đường bờ (extraction of edges) hoặc các đặc tính ảnh khác, còn trí tuệ nhân tạo được dùng cho những công việc xử lý mức cao như thao tác kí hiệu và quản lý cơ sở tri thức. Chúng ta chỉ nghiên cứu một số kỹ thuật xử lý ở mức thấp dùng trong lý giải ảnh, coi như là một bộ phận của cải thiện, phục hồi, và mã hóa ảnh. Nghiên cứu kỹ hơn việc lý giải ảnh sẽ vượt quá phạm vi của cuốn sách này.

Trong chương này, chúng tôi trình bày cơ sở xử lý ảnh. Những cơ sở đó sẽ đặt nền móng cho phần thảo luận về cải thiện, phục hồi, mã hóa ảnh trong các chương sau. Trong phần 1, thảo luận về cơ sở xử lý ảnh. Trong phần 2 và 3, thảo luận những phân cơ bản của hệ thị giác ở con người. Trong phần 4, thảo luận những cơ sở của mô hình trường xử lý ảnh điển hình.

## 1. ÁNH SÁNG.

### 1.1. Ánh sáng là sóng điện từ.

Mọi vật mà chúng ta quan sát được nhờ ánh sáng. Có hai loại nguồn sáng. Loại thứ nhất gọi là nguồn sáng sơ cấp, tự nó phát ánh sáng. Ví dụ nguồn sáng sơ cấp gồm mặt trời, đèn điện, đèn cây (cây nến). Loại khác gọi là nguồn sáng thứ cấp, chỉ phản xạ hoặc khuếch tán ánh sáng được phát bởi nguồn khác, ví dụ nguồn sáng thứ cấp gồm mặt trăng, những đám mây và những quả táo.

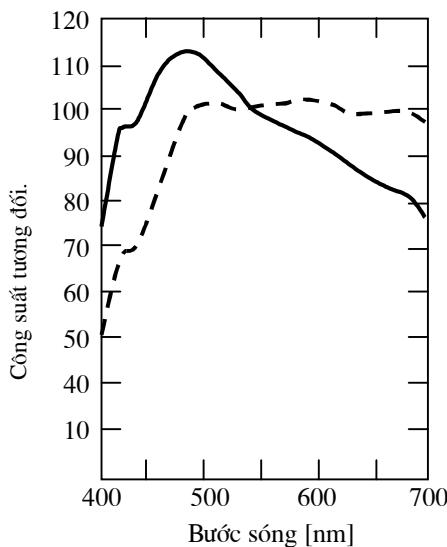
Ánh sáng là một phần của dải phổ liên tục bức xạ sóng điện từ. Sóng điện từ mang năng lượng và sự phân bố năng lượng của sóng điện từ đi qua một mặt phẳng không gian có thể mô tả bằng  $c(x,y,t, \lambda)$ , ở đó  $x$  và  $y$  là hai biến không gian,  $t$  là biến thời gian và  $\lambda$  là bước sóng. Hàm  $c(x,y,t, \lambda)$  được gọi là thông lượng bức xạ trên (diện tích  $x$  bước sóng) hoặc lượng bức xạ trên bước sóng. Bước sóng  $\lambda$  liên quan với tần số  $f$  bởi:

$$\lambda = c/f \quad (1.1)$$

$c$  là vận tốc của sóng điện từ, khoảng  $3 \cdot 10^8$  m/s trong chân không và không khí. Mặc dù  $c(x, y, t, \lambda)$  có thể biểu diễn theo hàm tần số, nhưng sử dụng bước sóng  $\lambda$  vẫn thuận tiện hơn. Đơn vị liên hệ với  $c(x,y,t, \lambda)$  là năng lượng trên (diện tích  $x$  thời gian  $x$  bước sóng) và là  $\text{J} \cdot (\text{m}^3 \cdot \text{s})$  trong hệ MKS (mét, kg, sec). Nếu chúng ta tích phân  $c(x,y,t, \lambda)$  theo biến  $\lambda$ , chúng ta nhận được lượng bức xạ có đơn vị là  $\text{J}/(\text{m}^2 \cdot \text{s})$  hoặc  $\text{W}/\text{m}^2$ . Bức xạ mặt trời xuyên qua mặt phẳng thẳng góc với tia bức xạ là  $1350 \text{ W}/\text{m}^2$  khi không có sự hấp thụ của không khí. Nếu chúng ta tích phân  $c(x,y,t, \lambda)$  với cả 4 biến  $x, y, t$  và  $\lambda$ , chúng ta có được tổng năng lượng (bằng  $J$ ) của sóng điện từ xuyên qua mặt phẳng không gian.

Ánh sáng khác với các sóng điện từ khác, - như sóng vô tuyến điện, là mắt người nhận biết được nó. Giả sử ta xét một điểm cố định  $(x',y')$  trong không gian và một thời điểm cố định  $(t')$ , thì hàm  $c(x,y,t, \lambda)$  có thể xem như là chỉ là hàm của biến  $\lambda$ . Chúng ta có thể diễn tả bởi  $c(x',y',t', \lambda)$  hoặc  $c(\lambda)$  cho thuận tiện. Ví dụ của  $c(\lambda)$  từ bức xạ mặt trời được biểu diễn trong hình 1.1. Mắt nhạy cảm với những sóng điện từ trong một dải cực kỳ hẹp của  $\lambda$ , đó là khoảng từ 350nm đến 750nm ( $1\text{nm} = 10^{-9}$  m). Hình 1.2 biểu diễn các loại sóng điện từ theo hàm của bước sóng  $\lambda$ . Bức xạ điện từ với  $\lambda$  lớn, từ vài cm đến hàng nghìn mét, có thể tạo ra bởi mạch điện. Bức xạ như vậy được

sử dụng cho truyền thông vô tuyến và radar. Bức xạ với  $\lambda$  ngay phía trên dải nhìn thấy được gọi là hồng ngoại, với  $\lambda$  ngay dưới vùng nhìn thấy được, gọi là tử ngoại. Cả bức xạ hồng ngoại và tử ngoại đều được phát bởi những nguồn sáng điển hình chẳng hạn như mặt trời. Bức xạ với  $\lambda$  ở phía xa dưới vùng nhìn thấy được gồm tia X, tia  $\gamma$ , và tia vũ trụ; với tia vũ trụ, bước sóng  $\lambda$  nhỏ hơn  $10^{-5}$  nm hoặc  $10^{-14}$  ms.



**Hình 1.1:** Thành phần phổ của bức xạ mặt trời, ở trên tầng khí quyển của trái đất (đường liên nét) và trên mặt đất tại Washington vào buổi trưa (đường nét đứt).

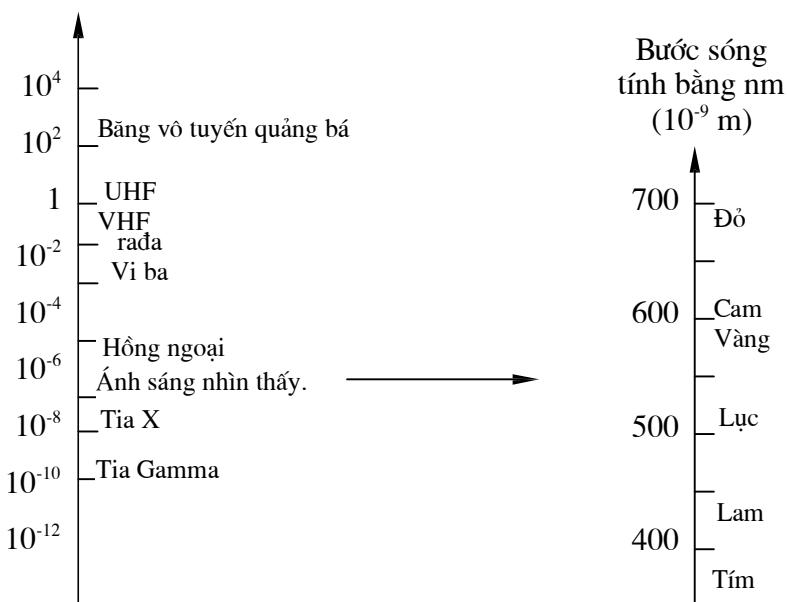
## 1.2. Độ sáng, màu sắc và độ bão hoà.

Sự nhận biết ánh sáng của loài người với  $c(\lambda)$  được mô tả chung bằng thuật ngữ độ sáng (brightness), màu sắc và độ bão hoà. Độ sáng (brightness) liên quan đến mức độ sáng của ánh sáng. Màu sắc liên quan đến màu, chẳng hạn như màu đỏ, màu cam hoặc màu mận chín(tía). Độ bão hoà đôi khi còn gọi là sắc độ, liên quan đến mức độ tươi hoặc xám của màu. Độ sáng, màu sắc và độ bão hoà là những thuật ngữ về nhận thức (perceptual terms), phụ thuộc vào một số nhân tố bao gồm dạng chi tiết của  $c(\lambda)$ , lịch sử quá khứ của những kích thích thị giác mà người quan sát đã trải qua, và môi trường cụ thể nơi ánh sáng được quan sát. Tuy nhiên có thể xét đến chúng một cách gần đúng như những đặc thù của  $c(\lambda)$ .

Để liên hệ cảm nhận độ sáng của loài người với  $c(\lambda)$  cần định nghĩa ra một số đại lượng trắc quang (photometric quantity). Những đại lượng liên hệ với  $c(\lambda)$ , chẳng

hạn như thông lượng bức xạ, lượng bức xạ và  $\text{W/m}^2$  được gọi là đơn vị đo bức xạ (radiometric unit). Các đại lượng vật lý đó có thể định nghĩa độc lập với người quan sát cụ thể. Sự đóng góp của  $c(\lambda_1)$  và  $c(\lambda_2)$  để tạo ra sự cảm nhận độ sáng của con người nói chung là hoàn toàn khác nhau khi  $\lambda_1 \neq \lambda_2$  mặc dù  $c(\lambda_1)$  có thể giống  $c(\lambda_2)$ . Chẳng hạn người quan sát không thể nhìn thấy một sóng điện từ với  $c(\lambda)$  bằng không trong vùng khả kiến của  $\lambda$ , mặc dù bên ngoài dải khả kiến  $c(\lambda)$  có thể rất lớn. Ngay cả trong vùng khả kiến, độ sáng cũng phụ thuộc  $\lambda$ . Vì lý do này, một tích phân đơn của  $c(\lambda)$  trên biến  $\lambda$  không phản ánh đúng sự cảm nhận độ sáng.

Bước sóng  
tính bằng m



**Hình 1.2:** Các loại sóng điện từ theo hàm của bước sóng  $\lambda$ .

Các đại lượng có xét đến đặc tính thị giác của con người, - do đó phản ảnh độ sáng tốt hơn tích phân của  $c(\lambda)$ , được gọi là những đại lượng trắc quang (photometric) .

Đại lượng trắc quang cơ bản là độ chói (luminance), được công nhận năm 1948 bởi CIE( Uỷ ban quốc tế về tiêu chuẩn ánh sáng và màu sắc) . Xét ánh sáng với  $c(\lambda) = 0$  tại mọi nơi ngoại trừ  $\lambda = \lambda_r$ , ở đây  $\lambda_r$  là một bước sóng tham chiếu cố định. Ánh sáng chỉ gồm một thành phần phổ (một bước sóng) gọi là ánh sáng đơn sắc. Giả sử chúng ta yêu cầu người quan sát so sánh độ chói của ánh sáng đơn sắc  $c(\lambda_r)$  với một ánh sáng

đơn sắc khác  $c'(\lambda_i)$ , ở đây  $\lambda_i$  là bước sóng thử. Giả sử người quan sát nói rằng  $c(\lambda_r)$  phù hợp với  $c'(\lambda_i)$  về độ sáng. Điểm mà độ sáng  $c(\lambda_r)$  và  $c'(\lambda_i)$  bằng nhau có thể nhận được với thí nghiệm cho chiếu hai vết sáng:  $c(\lambda_r)$  cố định và  $c'(\lambda_i)$  thay đổi, rồi yêu cầu người quan sát tăng hoặc giảm biên độ của  $c'(\lambda_i)$  cho đến khi chúng phù hợp về độ sáng. Tỷ số  $c(\lambda_r) / c'(\lambda_i)$ , ở đó  $c(\lambda_r)$  và  $c'(\lambda_i)$  phù hợp về độ sáng, được gọi là hiệu suất sáng tương đối của ánh sáng đơn sắc  $\lambda_i$  so với  $\lambda_r$ , và gần như độc lập với biên độ của  $c(\lambda_r)$  trong điều kiện quan sát bình thường. Bước sóng  $\lambda$  sử dụng là 550 nm (ánh sáng vàng-xanh lá cây), là bước sóng ở đó một người quan sát diễn hình có độ nhạy sáng cực đại. Với sự lựa chọn  $\lambda_r$  này, hiệu suất sáng tương đối  $c(\lambda_r) / c'(\lambda_i)$  luôn bé hơn hoặc bằng 1, vì  $c(\lambda_r)$  không lớn hơn  $c'(\lambda_i)$ ; nghĩa là ở  $\lambda_r$  cần ít năng lượng hơn để tạo ra độ sáng như nhau. Hiệu suất sáng tương đối là hàm của  $\lambda$  và được gọi là hàm hiệu suất sáng tương đối, kí hiệu bằng  $v(\lambda)$ . Cả hai ánh sáng đơn sắc  $c_1(\lambda_1)$  và  $c_2(\lambda_2)$  biểu hiện có độ sáng như nhau đối với người quan sát khi :

$$c_1(\lambda_1)v(\lambda_1) = c_2(\lambda_2)v(\lambda_2) \quad (1.2)$$

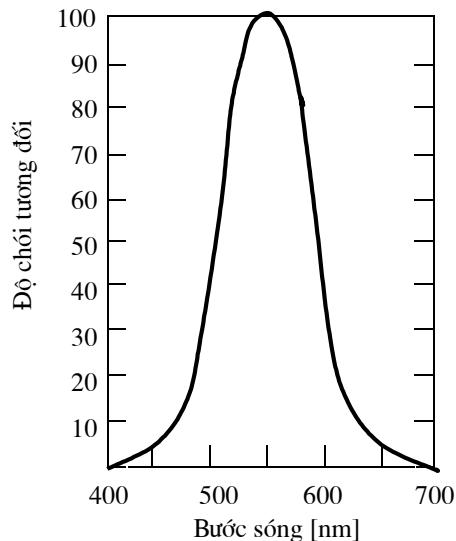
Hàm hiệu suất sáng tương đối  $v(\lambda)$  phụ thuộc vào người quan sát. Ngay cả với một người quan sát,  $v(\lambda)$  cũng hơi khác nhau khi đo ở những thời điểm khác nhau. Để loại bỏ sự biến thiên, năm 1929 CIE định nghĩa ra người quan sát chuẩn, dựa trên kết quả thực nghiệm đạt được từ một số người quan sát khác nhau. Kết quả hàm  $v(\lambda)$  được gọi là hàm hiệu suất sáng tương đối CIE và được minh họa trong hình 1.3. Hàm CIE đạt cực đại bằng 1 tại  $\lambda = 550$  nm.

Đơn vị cơ bản của độ chói là lumen (lm). Độ chói trên diện tích 1 của sáng với  $c(\lambda)$  được định nghĩa bởi:

$$l = k \int_{\lambda=0}^{\infty} c(\lambda)v(\lambda)d\lambda. \quad (1.3)$$

Trong công thức trên :      1 có đơn vị là lumen/m<sup>2</sup>  
 k=685 lumen/watt  
 $c(\lambda)$  có đơn vị là watt/m<sup>3</sup>  
 $\lambda$  có đơn vị là m  
 $v(\lambda)$  không có thứ nguyên.

Ánh sáng đơn sắc với độ chói  $1\text{W/m}^2$  tạo ra  $685 \text{lumen/m}^2$  khi  $v(\lambda) = 1$ . Điều này xảy ra khi  $\lambda = 555 \text{ nm}$ \*



**Hình 1.3** : Hàm hiệu suất sáng tương đối CIE.

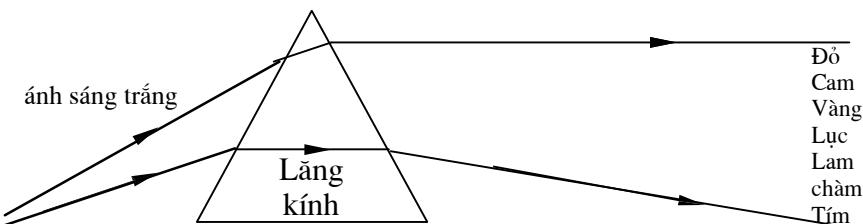
Với những bước sóng khác,  $v(\lambda) < 1$ , độ chói của ánh sáng đơn sắc phải lớn hơn  $1\text{W/m}^2$  để tạo ra độ chói trên diện tích  $685 \text{lumens/m}^2$ . Có nhiều đơn vị đo cường độ ánh sáng chẳng hạn như footcandle ( $\text{lumens/ft}^2$ ) và phot ( $\text{lumens/cm}^2$ ).

Ghi nhớ rằng độ chói và *độ chói trên diện tích* không đo được sự cảm nhận của người về độ sáng. Ví dụ ánh sáng với  $2 \text{lumen/m}^2$  không sáng gấp đôi ánh sáng với  $1 \text{lumen/m}^2$ . Có thể tạo ra môi trường để ánh sáng có giá trị *độ chói trên diện tích* nhỏ trông lại sáng hơn một ánh sáng khác mà *độ chói trên diện tích* lớn hơn. Tuy nhiên *độ chói trên diện tích* liên quan trực tiếp đến sự cảm nhận độ chói của con người nhiều hơn tích phân của  $c(\lambda)$ . Hơn nữa, trong những điều kiện quan sát điển hình (ánh sáng

\* Những thảo luận của chúng ta trong phần này có tính hómolog, với sự chấp nhận một số giả định hợp lý. Ví dụ (1.2) dựa trên luật bắc cầu, được phát biểu là nếu A và B sáng như nhau và B và C sáng như nhau thì A và C cũng sáng bằng nhau. Luật bắc cầu này được chứng minh gần đúng bằng thực nghiệm.

không quá yếu mà cũng không quá chói chang) thì ánh sáng với *độ chói trên diện tích* lớn hơn sẽ cho cảm giác sáng hơn là ánh sáng mà *độ chói trên diện tích* nhỏ hơn.

Màu sắc (hue) được định nghĩa là thuộc tính của màu cho phép chúng ta phân biệt màu đỏ với màu xanh lam. Trong một vài trường hợp, màu sắc có thể liên quan tới các đặc tính đơn giản của  $c(\lambda)$ . Ánh sáng với  $c(\lambda)$  là hằng số trong dải khả kiến (nhìn thấy được) có màu trắng hoặc không màu. Trong những điều kiện quan sát thường, ánh sáng đơn sắc xuất hiện màu và màu của nó phụ thuộc  $\lambda$ . Khi con người quan sát một chuỗi dải sáng đơn sắc đặt kề nhau, màu chuyển đổi êm ả từ màu sắc này sang màu sắc khác. Ánh sáng có thể bị lăng kính phân tích thành một chuỗi dải sáng đơn sắc như ta thấy trên hình 1.4. Thí nghiệm này được Newton thực hiện lần đầu vào năm 1666. Newton chia phổ màu trong dải khả kiến thành bảy loại: đỏ, cam, vàng, lục, lam, chàm, tím với bước sóng giảm dần, gọi là bảy màu cầu vồng. Thoạt tiên Newton chỉ bắt đầu với các màu đỏ, vàng, lục, lam, tím. Sau đó ông thêm màu cam và màu chàm để tạo thành số 7 (giống như chia 7 ngày một tuần, nhạc có 7 nốt và v.v...).



**Hình 1.4 :** Lăng kính phân tích ánh sáng trắng thành chuỗi ánh sáng đơn sắc.

Khi ánh sáng không đơn sắc nhưng  $c(\lambda)$  của nó có dải hẹp và hầu hết toàn bộ năng lượng của nó tập trung trong  $\lambda' - \Delta\lambda < \lambda < \lambda' + \Delta\lambda$  với  $\Delta\lambda$  nhỏ, thì màu sắc trông thấy tương tự như ánh sáng đơn sắc với  $\lambda = \lambda'$ . Tuy nhiên màu có biểu hiện kém tinh khiết hơn ánh sáng đơn sắc cùng màu sắc. Khi  $c(\lambda)$  là hàm bất kỳ, khó có thể coi màu sắc như một trong những đặc tính đơn giản của  $c(\lambda)$ . Bằng cách lựa chọn  $c(\lambda)$  thích hợp, có thể tạo ra màu sắc không tương ứng với bất kỳ ánh sáng đơn sắc nào. Trộn ánh sáng đỏ với ánh sáng xanh lam có thể tạo ra ánh sáng màu tía (purple).

Độ bão hòa liên quan đến tính tinh khiết hoặc sắc sỡ của màu. Ánh sáng đơn sắc có phổ tinh khiết và trông rất sắc sỡ, tinh khiết. Khi đó người ta nói là độ bão hòa cao. Còn khi thành phần phổ của  $c(\lambda)$  mở rộng, sẽ cảm nhận thấy màu kém chói lọi và tinh khiết, ta bảo là độ bão hòa kém. Độ bão hòa màu liên quan mật thiết với độ rộng hiệu dụng của  $c(\lambda)$ .

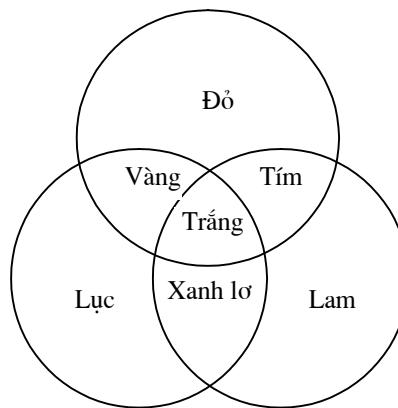
### 1.3. Hệ màu cộng và hệ màu trừ.

Khi tổ hợp hai ánh sáng  $c_1(\lambda)$  và  $c_2(\lambda)$ , ánh sáng nhận được là  $c(\lambda)$  được tính theo:

$$c(\lambda) = c_1(\lambda) + c_2(\lambda) \quad (1.4)$$

Khi ánh sáng cộng vào nhau như ở (1.4), ta được là hệ màu cộng (additive color system). Đem cộng nhiều nguồn sáng với những bước sóng khác nhau, sẽ tạo ra được nhiều màu khác nhau. Ví dụ màn đèn hình tivi màu được phủ với những chấm photpho nhỏ rực rỡ xếp thành từng cụm 3 màu. Mỗi nhóm gồm 1 điểm màu đỏ, một điểm màu lục và một điểm màu lam. Sử dụng 3 màu đó là vì khi tổ hợp một cách thích hợp chúng có thể tạo ra một dải màu rộng hơn mọi tổ hợp của những bộ ba màu khác, chúng là những màu cơ bản của hệ màu cộng. Màu của những ánh sáng đơn sắc thay đổi từ từ và khó xác định được những bước sóng riêng ứng với đỏ (R), lục (G) và lam (B). CIE chọn  $\lambda = 700$  nm cho màu đỏ,  $\lambda = 546,1$  nm cho màu xanh lục và  $\lambda = 435,8$  nm cho màu lam.

Ba màu cơ bản của hệ màu cộng được biểu diễn trên hình 1.5. Trong hệ màu cộng, sự trộn lẫn màu lam và màu lục với số lượng bằng nhau sẽ tạo ra màu lục lam (cyan). Sự trộn lẫn màu đỏ và màu lam với số lượng bằng nhau sẽ tạo ra màu đỏ thẫm (magenta) và sự trộn lẫn màu đỏ và màu lục với số lượng bằng nhau tạo ra màu vàng. Ba màu vàng (Y), lục lam (C) và đỏ thẫm (M) gọi là những màu thứ cấp của hệ màu cộng. Khi 3 màu R, G, B được kết hợp với số lượng bằng nhau, kết quả sẽ là màu trắng. Do vậy khi các thành phần R, G, B được sử dụng trong màn hình TV màu với số lượng như nhau, thì kết quả là sẽ ra hình ảnh đen trắng. Đem kết hợp các thành phần R, G và B với phân lượng khác nhau, có thể được tạo ra nhiều màu khác nhau. Ví dụ, sự trộn lẫn ánh sáng đỏ và ánh sáng màu lục yếu, không có ánh sáng màu lam, sẽ tạo ra ánh sáng nâu.



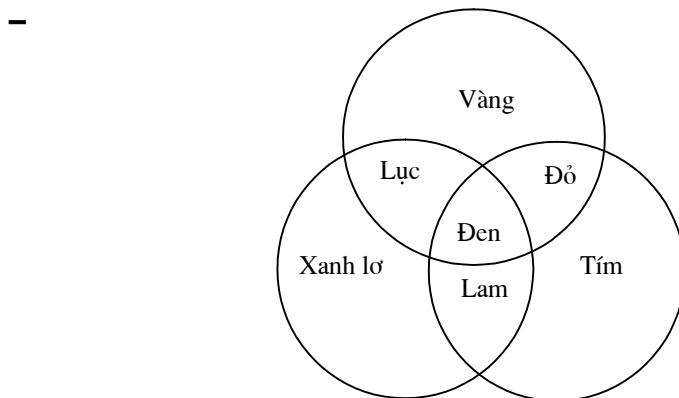
**Hình 1.5:** Các màu cơ bản của hệ thống màu cộng.

Thiên nhiên thường tạo ra màu sắc bằng cách lọc bỏ (trừ đi) một số bước sóng và phản xạ những bước sóng khác. Việc trừ bỏ bước sóng được thực hiện bởi những nguyên tử gọi là sắc tố (pigment), chúng hấp thụ những phần đặc biệt của phổ. Ví dụ, khi ánh sáng mặt trời gồm nhiều bước sóng khác nhau chiếu vào quả táo đỏ, hệ thống hàng tỷ phân tử sắc tố trên bề mặt của quả táo hấp thụ tất cả các bước sóng ngoại trừ bước sóng ứng với màu đỏ. Kết quả là ánh sáng phản xạ có hàm c(λ) gây cảm nhận màu đỏ. Các sắc tố lấy đi những bước sóng và hỗn hợp của hai loại sắc tố khác nhau sẽ tạo ra ánh sáng phản xạ mà bước sóng ngắn hơn. Đó là *hệ màu trừ* (subtractive color system). Khi hai thứ mục có màu khác nhau được trộn để tạo ra một màu khác trên giấy thì đây cũng là một *hệ màu trừ*.

Ba màu cơ bản của *hệ màu trừ* là vàng (Y), lục lam (cyan) và đỏ thẫm (M), chúng là những màu thứ cấp của *hệ màu cộng*. Ba màu này được biểu diễn trên hình 1.6. Bằng việc trộn các màu đó với những hàm lượng thích hợp, có thể tạo ra một dải màu rộng. Trộn màu vàng và màu lục lam tạo ra màu lục. Trộn màu vàng và màu đỏ thẫm tạo ra màu đỏ. Trộn màu lục lam và màu đỏ thẫm tạo ra màu lam. Do vậy ba màu đỏ, lục và lam, những màu cơ bản của *hệ màu cộng*, lại là những màu thứ cấp của *hệ màu trừ*. Khi tất cả ba màu cơ bản Y, C, M được kết hợp, kết quả là màu đen, các sắc tố hấp thụ tất cả bước sóng ánh sáng nhìn thấy.

Điều quan trọng cần lưu ý là: *hệ màu trừ* khác một cách cơ bản với *hệ màu cộng*. Trong *hệ màu cộng*, khi chúng ta thêm các màu với bước sóng khác nhau, ánh sáng nhận được gồm nhiều bước sóng hơn. Chúng ta bắt đầu với màu đen, tương ứng với

không có ánh sáng . Khi chúng ta đi từ màu cơ bản (RGB) đến các màu thứ cấp (YCM) và rồi đến màu trắng, chúng ta làm tăng các bước sóng trong ánh sáng nhận được.Trong hệ màu trù, chúng ta bắt đầu với màu trắng, tương ứng với không có sắc tố. Khi chúng ta đi từ các màu cơ bản (YCM) đến các màu thứ cấp (RGB) rồi đến màu đen, chúng ta làm giảm những bước sóng trong ánh sáng phản xạ nhận được.



**Hình 1.6:** Các màu cơ bản của hệ màu trù.

Trong một hệ màu cộng, chúng ta có thể coi ánh sáng đỏ, lục, lam, là kết quả của ánh sáng trắng đi qua ba bộ lọc thông dải khác nhau. Trộn hai màu có thể coi như ánh sáng trắng đi qua một bộ lọc tổ hợp song song của hai bộ lọc thông dải tương ứng. Trong hệ màu trù, chúng ta có thể coi các ánh sáng màu vàng, lục lam và đỏ thẫm như là kết quả của ánh sáng trắng đi qua ba bộ lọc chặn dải khác nhau. Trộn hai màu có thể coi là kết quả của ánh sáng trắng đi qua hai bộ lọc chặn dải tương ứng đặt nối tiếp.

#### 1.4. Biểu diễn ảnh đơn sắc và ảnh màu .

Với ảnh đen-trắng, ánh sáng  $c(\lambda)$  có thể được biểu diễn bởi một số I như sau:

$$I = k \int_{\lambda=0}^{\infty} c(\lambda) S_{BW}(\lambda) d\lambda \quad (1.5)$$

Trong đó  $S_{BW}(\lambda)$  là đặc tính phổ của cảm biến được sử dụng và  $k$  là hệ số tỷ lệ xích (scaling constant). Vì sự cảm nhận độ sáng có tầm quan trọng hàng đầu đối với ảnh đen trắng, nên  $S_{BW}(\lambda)$  được chọn giống như hàm hiệu suất sáng tương đối đã được thảo luận trong phần 1.2. Giá trị I thường được gọi là độ chói, cường độ, hay mức xám của ảnh

đen trắng. Vì I trong công thức (1.5) biểu diễn công suất trên đơn vị diện tích, nên nó bao giờ cũng không âm và hữu hạn, nghĩa là:

$$0 \leq I \leq I_{\max}$$

Trong đó  $I_{\max}$  là giá trị lớn nhất mà I đạt được. Trong xử lý ảnh, I được chia thang (scaled) sao cho nó nằm trong một phạm vi thuận lợi nào đó, ví dụ  $0 \leq I \leq 1$  hoặc  $0 \leq I \leq 255$ . Trong những trường hợp này 0 ứng với mức tối nhất và 1 hoặc 255 ứng với mức sáng nhất. Vì cách đặt mức thang này nên đơn vị trắc quang (photometric) hoặc bức xạ (radiometric) cụ thể gắn với I trở nên không quan trọng. Ảnh trắng đen, trong cảm nhận chỉ có một màu. Vì vậy có khi gọi nó là ảnh đơn sắc (monochrome).

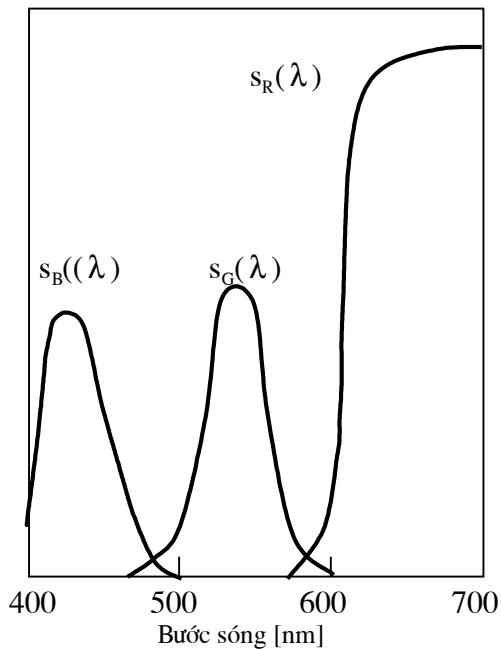
Ảnh mẫu có thể coi như 3 ảnh đơn sắc. Với ảnh màu, ánh sáng với hàm  $c(\lambda)$  được đại diện bởi 3 con số gọi là giá trị capse ba (tristimulus values). Một tập 3 con số thường dùng trong thực tế là R, G, và B, theo thứ tự đại biểu cho cường độ của các thành phần đỏ, lục và lam. Bộ ba giá trị R, G và B nhận được từ:

$$R = k \int_{\lambda=0}^{\infty} c(\lambda) S_R(\lambda) d\lambda \quad (1.7a)$$

$$G = k \int_{\lambda=0}^{\infty} c(\lambda) S_G(\lambda) d\lambda \quad (1.7b)$$

$$B = k \int_{\lambda=0}^{\infty} c(\lambda) S_B(\lambda) d\lambda \quad (1.7c)$$

Ở đó  $S_R(\lambda)$ ,  $S_G(\lambda)$  và  $S_B(\lambda)$  theo thứ tự là những đặc tính phổ của các cảm biến (bộ lọc) đỏ, lục và lam. Cũng như mức xám I trong ảnh đơn sắc, R, G, B là không âm và hữu hạn. Một bộ  $S_R(\lambda)$ ,  $S_G(\lambda)$  và  $S_B(\lambda)$  được biểu diễn trong hình 1.7. Ví dụ của  $f_R(x,y)$ ,  $f_G(x,y)$  và  $f_B(x,y)$  đại diện các thành phần đỏ, lục, lam của 1 ảnh màu, theo thứ tự được biểu diễn trong hình 1.8(a), (b) và (c). Ảnh màu được hình thành khi ba thành phần được kết hợp bởi màn hình TV màu.



**Hình 1.7:** Ví dụ đặc tính phổ của các cảm biến màu đỏ, lục và lam.

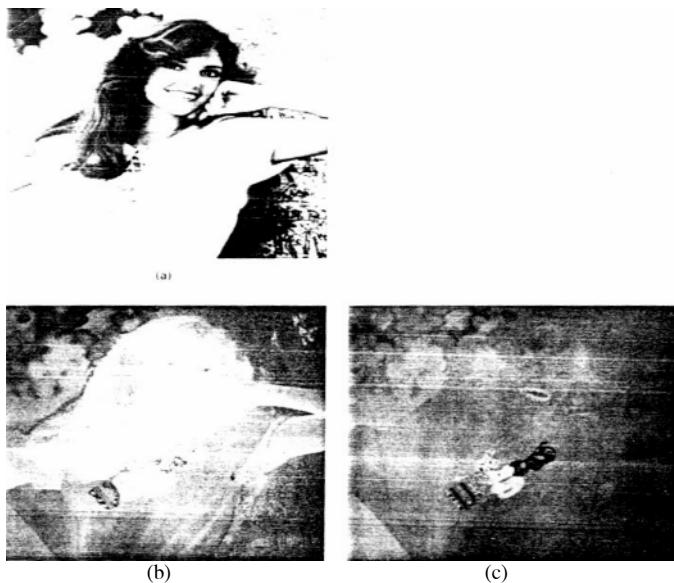
Một cách tiếp cận xử lý ảnh màu là xử lý 3 ảnh đơn sắc R, G và B riêng biệt và tổ hợp kết quả lại. Phương pháp tiếp cận này đơn giản và thường sử dụng trong thực tế. Vì độ sáng, màu sắc và độ bão hòa mỗi cái đều phụ thuộc cả 3 ảnh đơn sắc, nên việc xử lý riêng biệt R, G và B có thể tác động đến màu sắc và độ bão hoà, mặc dù có khi mục đích xử lý chỉ là thay đổi độ sáng.

Bộ ba giá trị R, G và B có thể được chuyển thành một số bộ ba giá trị khác. Một bộ cụ thể, được biết đến như *độ chói - sắc độ* (chrominance-luminance), khá hữu dụng trong thực tế. Khi R, G và B là các giá trị được sử dụng trong máy thu hình TV (theo hệ màu NTSC), thì giá trị *độ chói - sắc độ* tương ứng Y, I và Q liên hệ với R, G và B bởi:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.8a)$$

Và

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.273 & -0.647 \\ 1.000 & -1.104 & 1.701 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad (1.8b)$$



**Hình 1.9:** Các thành phần Y, I và Q của ảnh màu trong hình 1.8(d),  
 (a) thành phần Y; (b) thành phần I; (c) thành phần Q.

Thành phần Y được gọi là thành phần chói, vì nó phản ánh độ chói  $l$  trong công thức (1.3). Nó có vai trò chính trong sự nhận biết độ sáng của ảnh màu, và cũng có thể sử dụng được với ảnh đen trắng. Các thành phần I và Q gọi là các thành phần sắc độ, và chúng có vai trò chính trong sự nhận biết màu sắc và độ bão hòa của ảnh màu. Các thành phần  $f_Y(x,y)$ ,  $f_I(x,y)$  và  $f_Q(x,y)$  ứng với ảnh màu trong hình 1.8, theo thứ tự được biểu diễn như ba ảnh đơn sắc trong hình 1.9(a),(b) và (c). Vì  $f_I(x,y)$  và  $f_Q(x,y)$  có thể âm nên ta cộng thêm thiên áp cho chúng để hiển thị. Cường độ xám trung bình trong hình 1.9(b) và (c) đại biểu cho biên độ không của  $f_I(x,y)$  và  $f_Q(x,y)$ . So với bộ RGB, bộ ba giá trị YIQ có thuận lợi là ta có thể chỉ xử lý riêng thành phần Y. Ảnh đã xử lý sẽ khác với ảnh chưa xử lý trong biểu hiện độ sáng của nó. Một thuận lợi khác là hầu hết thành phần tần số cao của ảnh màu đều ở trong thành phần Y. Do vậy, lọc thông thấp các thành phần I và Q sẽ không ảnh hưởng đáng kể đến ảnh màu. Đặc tính này có thể được khai thác trong mã hoá ảnh màu số hoặc trong phát tín hiệu TV màu analog.

Khi mục đích của xử lý ảnh vượt quá yêu cầu tái tạo chính xác cảnh “gốc” theo cảm nhận của con người, chúng ta sẽ không giới hạn trong phạm vi dải sóng con người nhìn thấy được. Chẳng hạn khi muốn phát hiện một đối tượng phát nhiệt, thì việc có được một ảnh bằng cảm biến hồng ngoại dễ hơn nhiều so với ảnh màu thông thường .

Ảnh hồng ngoại có thể đạt được theo cách tương tự theo công thức (1.7), chỉ cần thay đổi một cách đơn giản các đặc tính phổ của cảm biến được sử dụng.

## 2. HỆ THỐNG THỊ GIÁC NGƯỜI

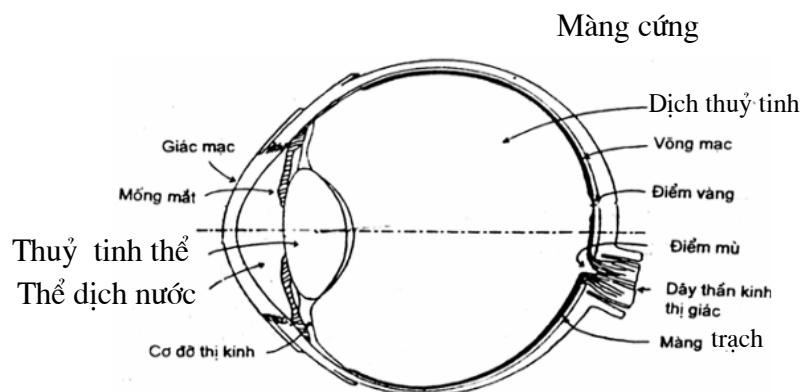
### 2.1. Mắt.

Hệ thống thị giác người là bộ phận phức tạp nhất hiện hữu. Hệ thống thị giác cho phép chúng ta tổ chức và hiểu biết nhiều phần tử phức tạp trong môi trường quanh ta. Hầu như với tất cả động vật, thị giác là phương tiện để duy trì sự sống còn. Với loài người thị giác không chỉ là trợ giúp sự sống còn mà còn là một công cụ của tư duy và phương tiện để làm cho cuộc sống phong phú hơn.

Hệ thống thị giác bao gồm mắt biến đổi ánh sáng thành tín hiệu thần kinh, và các bộ phận hữu quan của não xử lý các tín hiệu thần kinh để lấy ra thông tin cần thiết. Mắt,

khởi đầu hệ thống thị giác, là một hình cầu với đường kính khoảng 2 cm. Về mặt chức năng mà nói, thì mắt là thiết bị thu gom và hội tụ ánh sáng lên mặt sau của nó.

Hình cắt ngang của mắt được biểu diễn trong hình 1.10. Tại phía trước của mắt trông ra thế giới bên ngoài, là giác mạc cứng (cornea), một màng mỏng dai và trong suốt. Chức năng chính của giác mạc là để khúc xạ ánh sáng. Vì có hình tròn, nó hoạt động như thấu kính hội tụ của camera. Nó chịu trách nhiệm về gần 2/3 tổng ánh sáng khúc xạ cần thiết cho việc hội tụ chính xác.



**Hình 1.10** . Hình cắt ngang của mắt người.

Phía sau giác mạc có một thể dịch nước (aqueous humour) là một dung dịch trong veo, dễ lưu động. Qua giác mạc và thể dịch nước có thể trông thấy tròng đen

(iris), -- còn gọi là mống mắt. Bằng việc thay đổi kích cỡ đồng tử (con ngươi), -- một lỗ tròn nhỏ ở giữa tròng đen, tròng đen điều khiển lượng ánh sáng vào mắt . Đường kính đồng tử khoảng từ 1,5 mm đến 8 mm, khi tiếp xúc với ánh sáng càng chói thì đường kính đồng tử càng thu nhỏ . Màu của mống mắt qui định màu của mắt. Khi chúng ta nói rằng một người có mắt xanh, thì nghĩa là mống mắt màu xanh. Màu mống mắt tạo nên sự hấp dẫn của mắt, không có ý nghĩa gì về chức năng thị giác.

Phía sau mống mắt là thuỷ tinh thể, gồm nhiều sợi trong suốt được bao bọc trong màng mỏng đàn hồi trong suốt, có kích thước và hình dạng như một hạt đậu nhỏ. Thuỷ tinh thể phát triển trong suốt thời gian sống của con người. Do vậy thủy tinh thể của một người 80 tuổi rộng hơn 50% của người 20 tuổi. Như một cù hành, các tế bào thuộc lớp già nhất nằm ở trung tâm, và các tế bào thuộc lớp trẻ hơn nằm xa trung tâm. Thuỷ tinh thể có hình dạng hai mặt lồi và chiết suất 1,4 cao hơn tất cả các phần khác của mắt mà ánh sáng đi qua. Tuy nhiên thuỷ tinh thể được bao bọc bởi môi trường có chiết suất gần kề chiết suất của nó. Vì lý do này sự khúc xạ ánh sáng tại thuỷ tinh thể có góc khúc xạ nhỏ hơn nhiều so với tại giác mạc. Giác mạc có chiết suất khúc xạ 1,38 nhưng nó tiếp xúc với không khí có chiết suất bằng 1. Chức năng chính của thuỷ tinh thể là hội tụ chính xác ánh sáng vào màn ảnh phía sau mắt gọi là võng mạc. Một hệ thống với thấu kính cố định và khoảng cách cố định giữa thấu kính và màn ảnh, có thể hội tụ những vật ở một khoảng cách cụ thể. Ví dụ, nếu vật ở xa hội tụ rõ nét thì vật ở gần sẽ hội tụ phía sau màn ảnh. Để có thể hội tụ vật ở gần tại một thời điểm và vật ở xa tại vài thời điểm khác, camera thay đổi khoảng cách giữa thấu kính (cố định) và màn ảnh. Đó là trường hợp mắt của nhiều loại cá.

Trong trường hợp mắt người, hình dạng thuỷ tinh thể, chứ không phải là khoảng cách giữa thuỷ tinh thể và màn ảnh, được thay đổi. Quá trình thay đổi hình dạng để nhìn được cả gần và xa gọi là sự điều tiết của mắt. Thay đổi hình dạng là đặc tính quan trọng nhất của thuỷ tinh thể. Sự điều tiết của mắt xảy ra gần như ngay lập tức và được điều khiển bởi mi mắt, một nhóm cơ bao quanh thuỷ tinh thể.

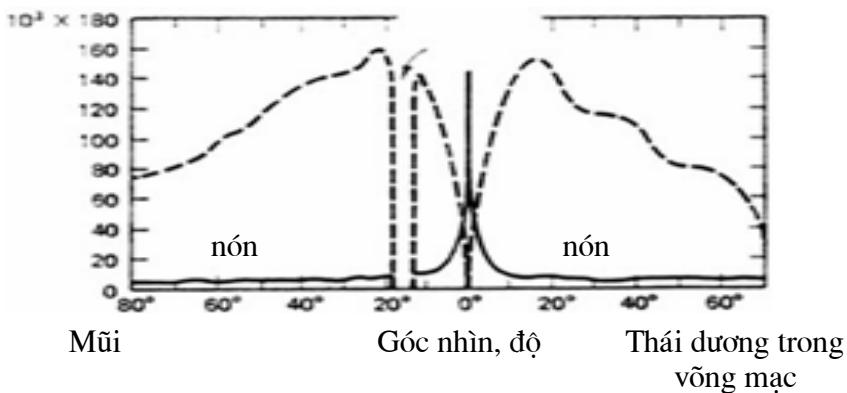
Phía sau thuỷ tinh thể là thuỷ tinh dịch, là một chất trong suốt như thạch. Nó được phối hợp về mặt quang học sao cho ánh sáng đã được thuỷ tinh thể hội tụ rõ nét rồi thì ánh sáng cứ giữ nguyên lộ trình. Thuỷ tinh dịch chứa trong toàn bộ không gian giữa thuỷ tinh thể và võng mạc, chiếm khoảng 2/3 dung tích mắt. Một trong những chức năng của nó là để giữ nguyên hình dạng mắt.

Phía sau dịch thuỷ tinh là võng mạc, nó phủ khoảng 65% phía trong nhãn cầu. Đây là màn hình, nơi ánh sáng vào được hội tụ và các tế bào tiếp nhận quang chuyển ánh sáng thành tín hiệu thần kinh. Tất cả các bộ phận của mắt mà chúng ta nói đến đều phục vụ cho chức năng đặt một hình ảnh rõ nét lên bề mặt cơ quan cảm nhận. Việc ảnh được tạo ra trên võng mạc, và mắt chỉ đơn giản là một thiết bị nhận ảnh, mãi đến tận đầu thế kỷ 17 người ta mới biết. Ngay cả thời Hy Lạp cổ đại đã biết cấu trúc của mắt một cách chính xác và đã tiến hành phẫu thuật mắt khá tinh vi cũng chỉ lập luận rằng có những tia tương tự ánh sáng (light-like) phát ra từ mắt đập vào vật và làm nó có thể thấy được. Cuối cùng sự thật xuất hiện, năm 1625 Scheiner chứng minh được rằng ánh sáng thâm nhập vào mắt và sự nhìn bắt nguồn từ ánh sáng thâm nhập vào mắt. Tách và và đem trung võng mạc của động vật và nhìn nó từ phía sau, ông đã thấy được ảnh lặp lại rất nhỏ của những vật trước nhãn cầu.

Có hai loại tế bào cảm nhận ánh sáng trong võng mạc. Chúng được gọi là tế bào hình nón và hình que. Hình nón, với số lượng khoảng 7 triệu, kém nhạy sáng hơn hình que và chủ yếu là để nhìn ban ngày. Chúng cũng có trách nhiệm cảm nhận màu sắc. Có ba loại hình nón theo thứ tự nhạy nhất với ánh sáng đỏ, lục và lam. Đây là cơ sở sinh lý học định tính của việc biểu diễn ảnh màu với ba ảnh đơn sắc đỏ, lục và lam. Hình que, số lượng khoảng 120 triệu, nhạy sáng hơn hình nón và về cơ bản để nhìn ban đêm. Vì hình nón chịu trách nhiệm cho ảnh màu không phản ứng khi ánh sáng tù mù, nên chúng ta không thể thấy màu trong bóng tối.

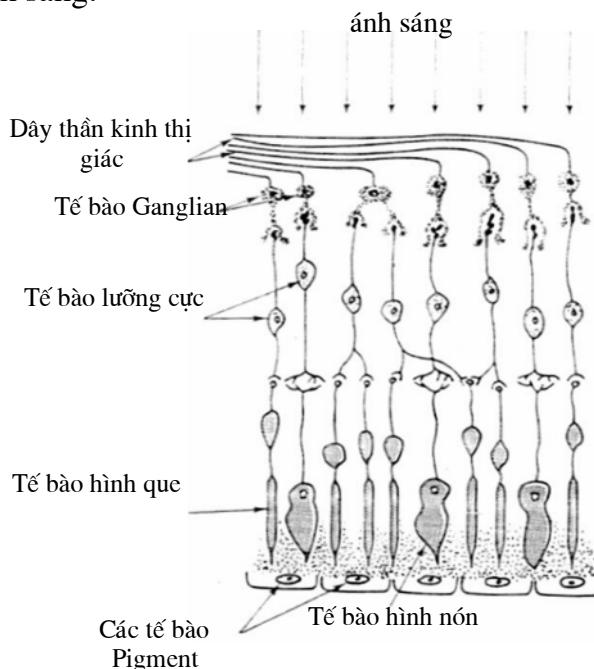
Tế bào hình que và hình nón phân bố khắp võng mạc. Tuy nhiên sự phân bố của chúng không đều. Sự phân bố của tế bào hình que và hình nón trong võng mạc được biểu diễn trong hình 1.11. Ngay phía sau điểm chính giữa con ngươi có một chỗ trũng trên võng mạc, gọi là điểm vàng (fovea). Ở đó tập trung đa số tế bào hình nón và hoàn toàn không có tế bào hình que.

Do đó, đây là vùng nhìn rõ nhất trong ánh sáng trắng. Khi ta nhìn thẳng vào một vật phía trước, vật được hội tụ trong điểm vàng (fovea). Vì điểm vàng (fovea) rất nhỏ, ta thường xuyên di chuyển sự chú ý từ vùng này sang vùng khác, khi xem xét một vùng rộng hơn. Tế bào hình que, hoạt động tốt nhất khi trời tối, được tập trung ở vùng xa điểm vàng (fovea). Vì không có tế bào hình que trong điểm vàng (fovea), nên một vật hội tụ trong điểm vàng (fovea) không thể thấy được trong bóng tối. Do đó ban đêm để thấy một vật, vào ta phải nhìn hơi nghiêng.



**Hình 1.11:** Sự phân bố tế bào hình que (đường chấm chấm) và hình non (đường liền nét) trên võng mạc.

Có nhiều lớp mỏng trong võng mạc. Tuy tế bào hình que và hình nón là các tế bào cảm nhận ánh sáng, đáng lý ra chúng phải nằm kề thuỷ tinh dịch, nhưng chúng lại ở xa hơn thuỷ tinh dịch. Do vậy ánh sáng phải đi qua các lớp khác của võng mạc, chẳng hạn đi qua các sợi thần kinh để tới tế bào hình nón, hình que. Điều này được mô tả trong hình 1.12. Thật không hiểu tại sao thiên nhiên lại chọn cách làm như vậy, nhưng trong thực tế cách sắp đặt này vẫn hoạt động tốt. Nhưng ít ra thì ta cũng thấy là ở điểm vàng (fovea) các dây thần kinh được đẩy sang một bên để các tế bào hình nón được phơi ra trước ánh sáng.



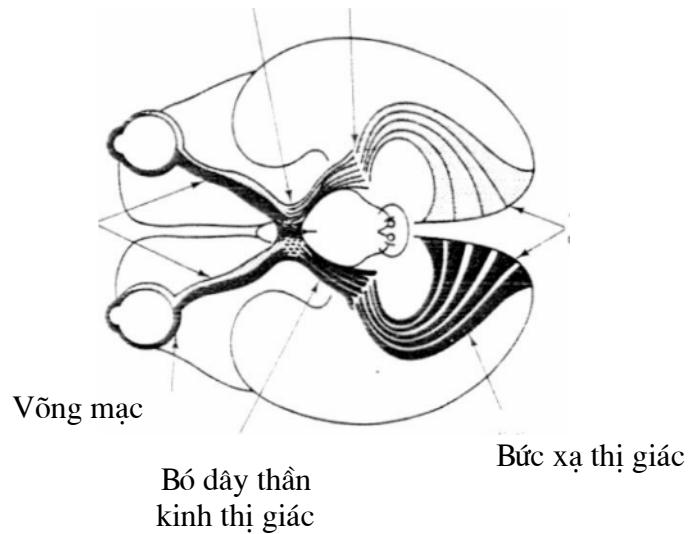
**Hình 1.12 :** Các lớp trong võng mạc. Lưu ý rằng ánh sáng phải đi qua nhiều lớp trước khi tới được các tế bào cảm nhận ánh sáng .

Vì cách sắp xếp đặc biệt này, các dây thần kinh ánh sáng phải xuyên qua các lớp tế bào cảm nhận ánh sáng trên đường tới não. Thay vì vượt qua các lớp tế bào cảm nhận ánh sáng ở khắp võng mạc, chúng được bó lại tại một vùng nhỏ bằng cõi đầu ghim trong võng mạc, gọi là điểm mù. Vì không có các tế bào cảm nhận ánh sáng trong vùng này, chúng ta không thể nhìn thấy ánh sáng hội tụ trên điểm mù.

Khi ánh sáng đậm tới tế bào hình nón và hình que, một phản ứng điện hoá phức tạp xảy ra, và ánh sáng được chuyển thành các xung thần kinh, truyền đến não thông qua dây thần kinh thị giác. Có khoảng 130 triệu tế bào cảm nhận ánh sáng (hình nón và hình que), nhưng chỉ có khoảng 1 triệu giây thần kinh. Điều đó có nghĩa là trung bình cứ một dây thần kinh phục vụ hơn 100 tế bào cảm nhận ánh sáng. Trong thực tế không phải là chia đều như vậy. Với một số tế bào hình nón trong điểm vàng (fovea) mỗi dây thần kinh phục vụ cho một tế bào, làm tăng tính nhạy sáng trong vùng này. Tuy nhiên, các tế bào hình que lại được chia đều cho các dây thần kinh. Đây là lý do tại sao tính nhạy sáng (visual acuity) vào ban đêm không tốt bằng ban ngày, tuy có nhiều tế bào hình que hơn hình nón.

## Chỗ các dây thần kinh thị giác giao nhau

## Vật cong gập như đầu gối

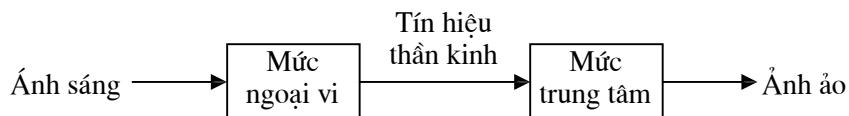


**Hình 1.13** : Đường các tín hiệu thần kinh đi từ võng mạc đến vỏ não thị giác.

Sau khi các bó dây thần kinh rời khỏi hai mắt, hai bó gặp nhau ở một điểm giao. Điều đó biểu diễn trên hình 1.13. Từ đó mỗi bó lại chia thành hai nhánh. Mỗi nhánh từ bó này lại kết với một nhánh từ bó kia, tạo thành hai bó mới. Sự giao kết của dây thần kinh thị giác từ hai mắt như vậy là một phần trong cơ chế tạo ra ảnh lập thể của ta, việc trộn lẫn các ảnh từ hai mắt cho phép trường thị giác cảm nhận không gian ba chiều. Hai bó mới đó đi qua hai cạnh trái và phải của một vật cong gấp như đầu gối. Các dây ban đầu kết thúc tại đây, các dây mới tiếp tục đến vỏ não thị giác, nơi các tín hiệu thần kinh được xử lý và tạo ra khả năng nhìn. Vỏ não thị giác là một phần nhỏ của vỏ não. Chưa ai biết gì nhiều về cách xử lý tín hiệu thần kinh thị giác trong vỏ não thị giác.

## 2.2. Mô hình về mức ngoại vi của hệ thị giác.

Hệ thị giác loài người thảo luận trong phần 2.1 có thể xem như hai hệ thống nối tiếp (xem hình 1.14). Hệ thứ nhất đại biểu cho mức ngoại vi của hệ thị giác, chuyển ánh sáng thành tín hiệu thần kinh. Hệ thứ hai đại biểu cho mức trung tâm của hệ thị giác, xử lý tín hiệu thần kinh để lấy ra thông tin.

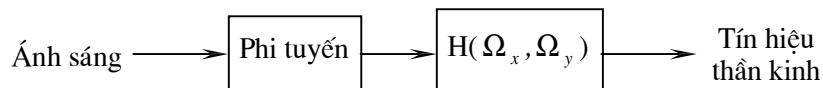


**Hình 1.14 :** Hệ thị giác người là một sự nối tiếp của hai hệ. Hệ thứ nhất đại biểu cho mức ngoại vi của hệ thị giác, chuyển ánh sáng thành tín hiệu thần kinh. Hệ thứ hai đại biểu cho mức trung tâm của hệ thị giác, xử lý tín hiệu thần kinh để lấy ra thông tin cần thiết.

Về quá trình xử lý ở mức trung tâm người ta hiểu biết còn quá ít, nhưng quá trình xử lý ở mức ngoại vi thì đã được hiểu biết cẩn kẽ, và đã có nhiều nỗ lực để tạo ra mô hình của nó. Hình 1.15 (mô hình Stockham) biểu diễn một mô hình rất đơn giản cho ảnh đơn sắc phù hợp với một số hiện tượng thị giác đã biết. Trong mô hình này, cường độ ảnh đơn sắc  $I(x,y)$  được biến đổi phi tuyến, chẳng hạn bằng thuật toán lôgarit, nén những cường độ mức cao nhưng dãn những cường độ mức thấp. Kết quả

được lọc bởi một hệ LSI (linear shift-invariant, hệ dịch bất biến tuyến tính) có đáp ứng tần số - không gian là  $H(\Omega_x, \Omega_y)$ . Phép biến đổi phi tuyến được đề xuất trên căn cứ một số kết quả thí nghiệm tâm-vật lý sẽ được thảo luận trong phần tiếp theo. Hệ LSI với  $H(\Omega_x, \Omega_y)$ , có đặc tính thông dải, được đề xuất trên căn cứ kích cỡ có hạn của con người, cũng như trên độ phân giải có hạn vì số tế bào nhạy sáng là hữu hạn, và trên quá trình cấm vùng bên (lateral inhibition process). Kích cỡ có hạn của con người và độ phân giải có hạn do số lượng tế bào nhạy sáng hữu hạn được phản ảnh bởi phân thông thấp trong đặc tính thông dải  $H(\Omega_x, \Omega_y)$ . Quá trình cấm vùng bên xuất xứ từ việc một dây thần kinh phải phục vụ nhiều tế bào hình nón và hình que. Đáp ứng của dây thần kinh là một tổ hợp tín hiệu từ các tế bào hình que và hình nón. Trong khi một số tế bào hình que và hình nón góp phần tích cực, một số khác góp phần tiêu cực (ức chế). Quá trình cấm vùng bên được phản ảnh bởi phân thông cao trong đặc tính thông dải  $H(\Omega_x, \Omega_y)$ . Mặc dù mô hình trong hình 1.15 rất đơn giản và chỉ áp dụng đối với quá trình xử lý ở mức ngoại vi, nó cũng tỏ ra rất hiệu quả khi phân tích một số hiện tượng thị giác sẽ được bàn đến ở tiết sau.

Một cách để khai thác mô hình ở hình 1.15 là xử lý một ảnh ở một miền ở gần kề chỗ sự nhìn xảy ra. Cái đó có thể có ích trong một số ứng dụng. Ví dụ, trong mã hoá ảnh, những thông tin có trong ảnh nhưng bị thị giác loại bỏ thì không cần thiết phải mã hoá. Xử lý ảnh trong một miền ở gần kề chỗ sự nhìn xảy ra thì phải nhấn mạnh hơn vào cái gì quan trọng với hệ thị giác. Đó là một lý do tại sao một số thuật toán xử lý ảnh thực hiện trong miền log cường độ chứ không phải trong miền cường độ.



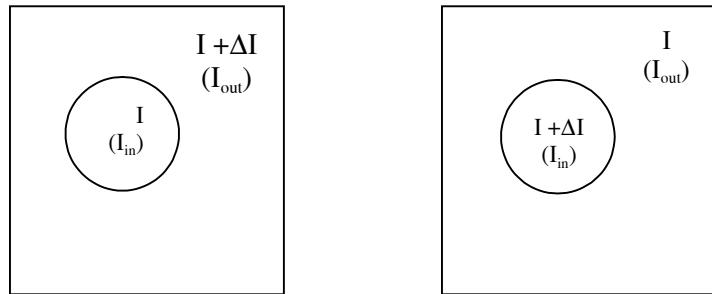
**Hình 1.15:** Mô hình đơn giản của hệ thị giác con người ở mức ngoại vi

### 3. CÁC HIỆN TƯỢNG THỊ GIÁC.

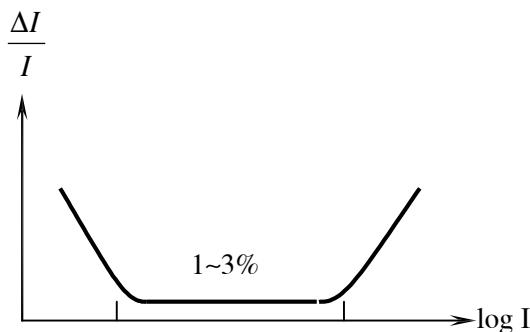
#### 3.1. Độ nhạy cảm cường độ.

Một cách lượng hoá khả năng của con người phân biệt 2 tác nhân kích thích thị giác giống nhau, chỉ khác về cường độ hoặc độ chói là *đo mức chênh lệch vừa đủ để nhận thấy* (just-noticeable difference, - j.n.d).

J.n.d có thể được định nghĩa và đo bằng nhiều cách. Một cách là thông qua thí nghiệm tâm lý gọi là *sự phân biệt cường độ*. Giả sử ta đưa tác nhân kích thích thị giác trong hình 1.16 cho một người quan sát xem. Vùng phía trong là một ảnh đơn sắc có cường độ đều  $I_{in}$ , được chọn ngẫu nhiên là  $I$  hoặc  $I + \Delta I$ . Vùng bên ngoài là một ảnh đơn sắc cường độ  $I_{out}$ , được chọn là  $I + \Delta I$  khi  $I_{in} = I$ , và là  $I$  khi  $I_{in} = I + \Delta I$ . Ta yêu cầu người quan sát lựa chọn xem trong hai cường độ  $I_{in}$  và  $I_{out}$  cái nào sáng hơn. Khi  $\Delta I$  rất lớn, người quan sát sẽ trả lời đúng hầu hết mọi lần hỏi, - đúng theo nghĩa là chọn vùng  $I + \Delta I$ . Khi  $\Delta I$  rất nhỏ, người quan sát chỉ trả lời đúng khoảng một 50% lần hỏi. Khi chúng ta di chuyển từ  $\Delta I$  rất rộng đến  $\Delta I$  rất hẹp, số phần trăm lần trả lời đúng của người quan sát giảm liên tục, và chúng ta có thể định nghĩa  $\Delta I$  là điểm mà người quan sát đưa ra câu trả lời đúng trong 75% số lần hỏi.



**Hình 1.16 :** Hai tác nhân kích thích dùng trong thí nghiệm phân biệt cường độ. Mỗi lần thử đưa một trong hai tác nhân kích thích cho người quan sát xem và yêu cầu chọn xem giữa  $I_{in}$  và  $I_{out}$  cái nào sáng hơn. Tác nhân kích thích dùng trong thí nghiệm được chọn ngẫu nhiên từ hai tác nhân kích thích. Kết quả của thí nghiệm này có thể dùng để đo  $\Delta I$  theo hàm của  $I$ .



**Hình 1.17:** Biểu đồ của  $\Delta I/I$  theo hàm của  $I$ . Giá trị  $\Delta I$  là *mức chênh lệch vừa đủ để nhận thấy* (j.n.d). Trên một phạm vi rộng của  $I$ , tỷ số  $\Delta I/I$  gần như là hằng số. Quan hệ này được gọi là định luật Weber.

Đường biểu diễn  $\Delta I/I$  theo hàm của  $I$ , trong đó  $\Delta I$  là j.n.d được biểu diễn trên hình 1.17.

$$\frac{\Delta I}{I} = \text{const} \quad (1.9)$$

Mối quan hệ này được gọi là định luật Weber. Định luật này phát biểu rằng *mức chênh lệch vừa đủ để nhận thấy*  $\Delta I$  tỷ lệ với  $I$ . Khi ta tăng  $I$  cần có  $\Delta I$  lớn hơn để nhận thấy  $I + \Delta I$  khác  $I$ . Đây là một cách hệ thị giác duy trì sự nhạy cảm với một dải rộng của cường độ tác nhân kích thích. Luật Weber không những đúng cho thị giác mà hầu như còn đúng cho tất cả các giác quan khác của con người: thính giác, khứu giác, vị giác và xúc giác.

Khi cho  $\Delta I$  tiến đến 0, (1.9) viết lại như sau:

$$\frac{dI}{I} = d(\log I) = \text{constant}. \quad (1.10)$$

Từ (1.10) thấy rằng j.n.d là hằng số trong miền  $\log I$  với một dải rộng giá trị của  $I$ . Điều này thích hợp với khái niệm áp dụng thuật toán phi tuyến, - chẳng hạn log, cho cường độ ảnh trong mô hình đơn giản ở hình 1.15. Thí nghiệm phân biệt cường độ là một công việc rất đơn giản đối với người quan sát, và có lẽ không cần đến xử lý phức

tập ở mức trung tâm. Cho nên kết quả thí nghiệm phân biệt cường độ chỉ liên quan tới xử lý ở mức ngoại vi trong hệ thị giác.

Kết quả thí nghiệm phân biệt cường độ phát biểu rằng  $\Delta I$  tăng khi  $I$  tăng, điều đó giải thích một phần nào tại sao trong vùng tối dễ nhận thấy nhiều ngẫu nhiên với hàm mật độ xác suất đều hơn là ở trong vùng sáng. Điều này được minh họa trong hình 1.18. Ảnh trong hình 1.18 là kết quả của sự tăng nhiễu trắng với mật độ xác xuất đề u ở một ảnh ban đầu không bị xuống cấp. Sự xuất hiện nhiều hạt nhỏ do nhiễu, trong vùng nền tối đều thấy rõ hơn trong vùng nền sáng đều. Vì ứng với  $I$  lớn hơn có  $\Delta I$  lớn hơn, nên trong vùng sáng phải có mức nhiễu cao hơn nhiều mới nhận thấy sự xuống cấp mà một mức nhiễu thấp gây ra trong vùng tối. Sự quan sát này cho thấy, khi xử lý ảnh giảm nhiễu trong vùng tối quan trọng hơn giảm nhiễu trong vùng sáng.



**Hình 1.18.** Ảnh 512\*512 pixels xuống cấp bởi nhiễu trắng với hàm mật độ xác suất đều. Cùng mức nhiễu thì ở vùng tối dễ thấy rõ hơn trong vùng sáng, ở vùng mức nền đều dễ thấy rõ là vùng có các cạnh viền.

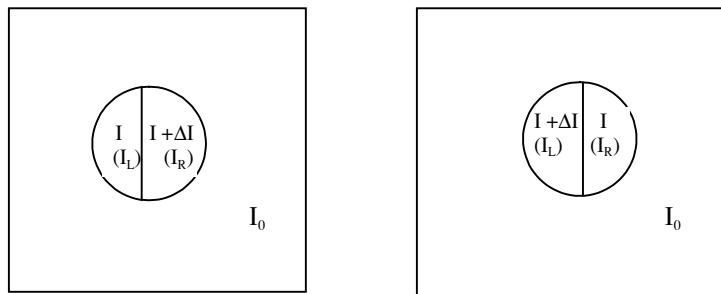
### 3.2. Sự thích nghi (Adaptation).

Trong thí nghiệm phân biệt cường độ thảo luận ở trên, cường độ biểu diễn tại thời gian bất kỳ đã cho là  $I$  và  $I + \Delta I$ . Nếu giả sử người quan sát sử dụng một số thời gian trước khi ra quyết định, thì kết quả nhận được là khi người quan sát đã thích nghi với mức cường độ  $I$ . Khi mức cường độ mà người quan sát đã thích nghi được khác với  $I$ , thì khả năng phân giải cường độ của người quan sát đã giảm. Giả sử chúng ta tiến hành thí nghiệm phân biệt cường độ đã thảo luận trong phần 3.1 nhưng bao quanh  $I$  và  $I + \Delta I$  là một vùng rộng lớn hơn với cường độ  $I_o$ , như ta thấy trên hình 1.19. Kết quả biểu đồ  $\Delta I/I$  theo hàm của  $I$  và  $I_o$  được biểu diễn trong hình 1.20. Khi  $I_o$  bằng  $I$  kết quả

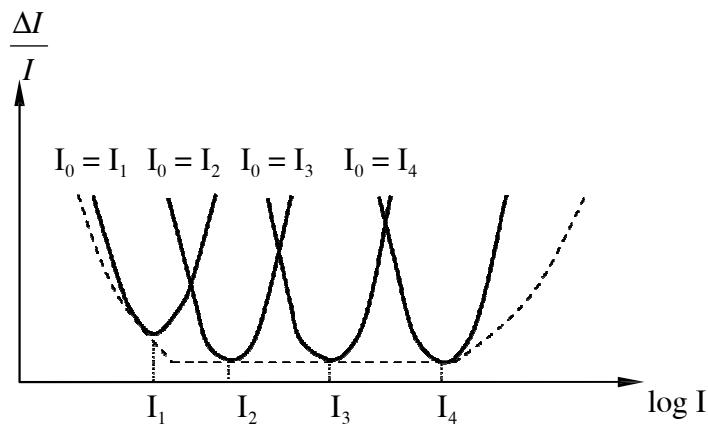
giống như hình 1.17. Khi  $I_0$  khác  $I$ ,  $\Delta I$  tăng lên so với trường hợp  $I_0 = I$ , nghĩa là mức độ nhạy cảm cường độ của người quan sát giảm. Kết quả chỉ ra rằng độ nhạy cảm cường độ lớn nhất khi ở gần mức thích nghi của người quan sát. Đây là cách khác để hệ thị giác đáp ứng một phạm vi rộng của cường độ tại những thời điểm khác nhau.

### 3.3 Hiệu ứng dải Mach và đáp ứng tần số không gian.

Xét một ảnh mà cường độ là hằng số theo chiều dọc nhưng tăng theo hàm bậc thang trên chiều ngang như trong hình 1.21(a). Cường độ theo chiều ngang được bi



**Hình 1.19**. Hai tác nhân kích thích dùng trong nghiên cứu hiệu ứng thích nghi về độ nhạy cảm cường độ. Mỗi lần thử ta cho người quan sát nhìn một trong hai tác nhân kích thích và yêu cầu chọn xem  $I_R$  hay  $I_L$  sáng hơn. Tác nhân kích thích trong mỗi lần thử được chọn ngẫu nhiên từ hai tác nhân. Các kết quả của thí nghiệm này có thể dùng để đo  $\Delta I$  theo hàm  $I$  và  $I_0$ .



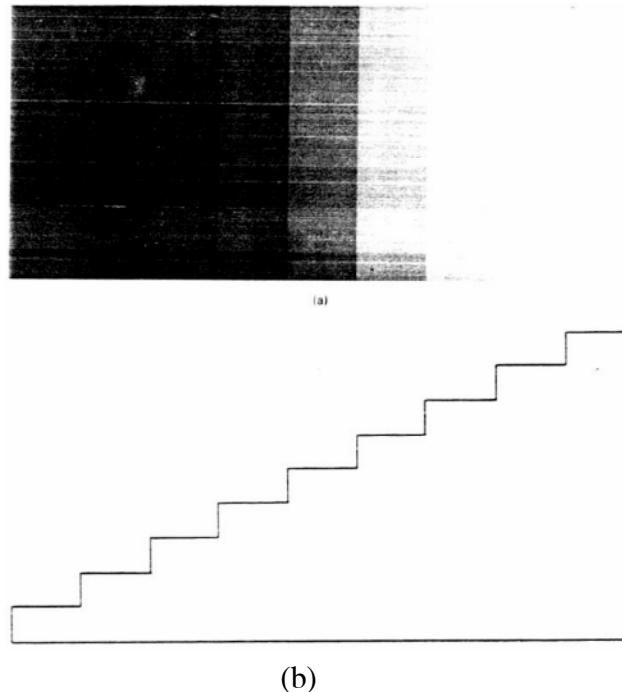
**Hình 1.20.** Biểu đồ của  $\Delta I/I$  theo hàm của  $I$  và  $I_0$ . Khi  $I_0$  bằng  $I$ ,  $\Delta I/I$  giống như hình 1.17 (đường vế chấm trong hình). Khi  $I_0$  khác  $I$ ,  $\Delta I$  tăng so với trường hợp  $I_0 = I$ . Điều này chứng tỏ rằng độ nhạy cảm cường độ của người quan sát đã giảm.

điễn trên hình 1.21(b). Tuy cường độ trong mỗi vùng hình chữ nhật là không đổi, nhưng ta cảm nhận thấy ở mỗi vùng về phía trái đều sáng hơn và về p hía phải tối hơn, điều đó gọi là hiệu ứng dài *Mach*. Hiện tượng này phù hợp với bộ lọc không gian trong mô hình ở mức ngoại vi của hệ thị giác trong hình 1.15. Khi áp dụng bộ lọc cho tín hiệu có những mất liên tục sắc cạnh, sẽ xảy ra sự quá ngưỡng (overshoot) và dưới ngưỡng (undershoot). Điều đó một phần là do hiện tượng cảm nhận độ sáng không đều trong vùng cường độ đều. Như vậy rút ra kết luận là trong xử lý ảnh không cần duy trì chính xác hình dạng viền cạnh.

Sự có mặt của bộ lọc thông giải không gian trong hệ thị giác có thể mô tả trong hình 1.22 . Ảnh I(x,y) trong hình 1.22 được tính theo công thức:

$$I(x, y) = I_o(y)\cos(\omega(x)x) + \text{constant} \quad (1.11)$$

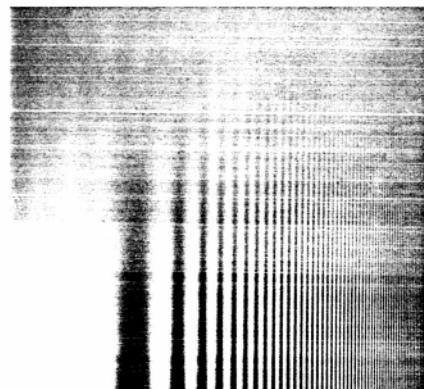
Hằng số được chọn sao cho  $I(x, y)$  là dương với tất cả  $(x, y)$ . Khi chúng ta dịch chuyển theo chiều ngang từ trái qua phải, tần số không gian  $\omega(x)$  tăng.



**Hình 1.21** . Minh họa hiệu ứng dài *Mach*.

Khi chúng ta dịch chuyển theo chiều dọc từ trên xuống dưới cường độ  $I_o(y)$  tăng. Nếu đáp ứng tần số không gian là hằng số trong suốt dải tần, thì mức độ nhạy cảm

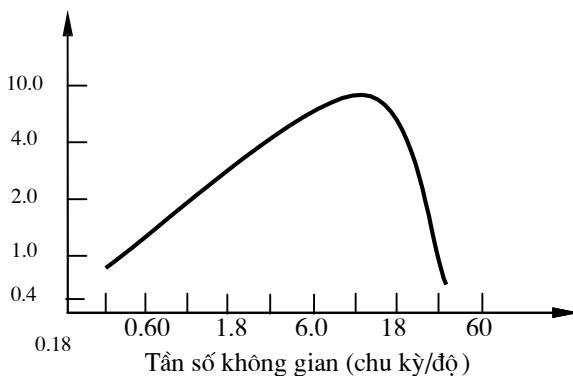
cường độ có thể là hằng số theo chiều ngang. Trong hình 1.22, chúng ta nhạy cảm với độ tương phản trong vùng trung tâm hơn trong vùng tâm thấp và cao, biểu thị tính chất thông dải của hệ thị giác. Đáp ứng tần số bộ lọc không gian  $H(\Omega_x, \Omega_y)$ , được đo chính xác hơn với giả định mô hình trong hình 1.15 là đúng, được biểu diễn trong hình 1.23. Trục ngang biểu diễn tần số không gian/góc nhìn. Tần số không gian về cảm nhận của một ảnh thay đổi theo hàm khoảng cách giữa mắt và ảnh. Khi khoảng cách tăng, tần số không gian cảm nhận cũng tăng. Để xét đến hiện tượng này thường dùng tần số không gian / góc nhìn (tần số không gian liên quan đến miền không gian trên võng mạc) để xác định  $H(\Omega_x, \Omega_y)$ . Đáp ứng tần số  $H(\Omega_x, \Omega_y)$  lớn nhất với tần số không gian trong dải 5 – 10 chu kỳ/độ và giảm khi tần số không gian tăng, hoặc bắt đầu giảm từ giá trị 5 – 10 chu kỳ/độ.



**Hình 1.22** . Vạch sóng hình sin đã điều chế minh họa tính chất thông dải của mức ngoại vi hệ thị giác con người.

### 3.4 Mặt nạ không gian.

Khi đem nhiều ngẫu nhiên có mức đồng đều thêm vào ảnh thì trong vùng nền đều dễ nhận thấy hơn là trong vùng có độ tương phản cao. Hiệu ứng này rõ rệt hơn hiệu ứng ảnh hưởng của độ sáng đến biểu hiện nhiều đã thảo luận trong phần 3.1. Ảnh trong hình 1.18, minh họa hiệu ứng độ sáng toàn bộ lên độ rõ nhiều.



**Hình 1.23.** Đáp ứng tần số  $H(\Omega_x, \Omega_y)$  trong mô hình trong hình 1.15.

Trong hình, nhiễu trong vùng biên không rõ bằng nhiễu trong vùng nền đều. Ngoài ra nhiễu trong vùng biên tối không rõ bằng nhiễu trong vùng nền đồng đều sáng. Một cách để giải thích điều đó là xét tỷ số tín hiệu trên nhiễu (SNR) tại chỗ. Nếu định nghĩa tỷ số SNR tại chỗ là tỷ số lượng biến thiên tín hiệu trên lượng biến thiên nhiễu trong một khu vực nhỏ, thì với mức nhiễu bằng nhau SNR trong vùng độ tương phản cao lớn hơn trong vùng nền đều. Một cách nhìn khác là mặt nạ không gian (hiệu ứng che khuất).

Trong vùng độ tương phản cao, mức tín hiệu cao và có khuynh hướng che khuất nhiễu nhiều hơn.

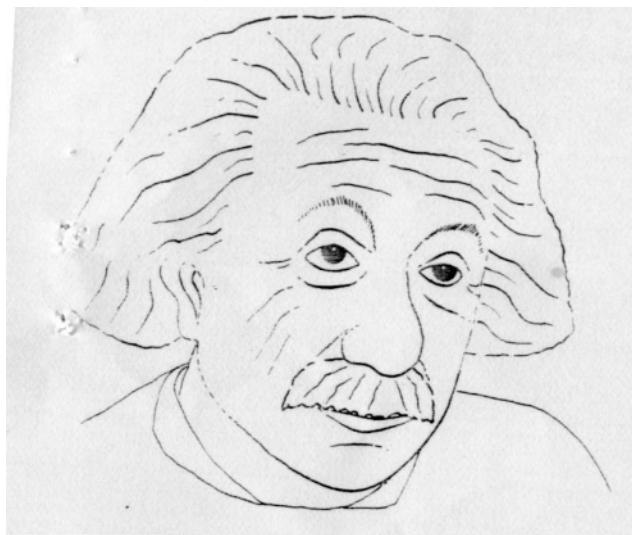
Hiệu ứng che khuất (mặt nạ không gian) có thể được khai thác trong xử lý ảnh. Ví dụ nỗ lực giảm nhiễu nền bằng bộ lọc không gian sẽ dẫn đến làm mờ ảnh ở mức độ nhất định. Trong vùng có độ tương phản cao, nơi hiệu ứng làm mờ ảnh do lọc không gian tạo ra rõ nét hơn, nhiễu lại không rõ, do đó chỉ cần lọc không gian ít.

### 3.5. Các hiện tượng thị giác khác.

Đối với người quan sát ảnh rõ nét trông đẹp hơn ảnh gốc. Điều này được sử dụng trong việc cải thiện biểu hiện của ảnh đối với người quan sát. Theo kinh nghiệm chung, cảnh quan nhân tạo hấp dẫn người xem cũng là điều thường tình. Khía cạnh tích cực của hiện tượng này là có thể đem khai thác trong những ứng dụng như sản xuất chương trình truyền hình thương mại. Khía cạnh tiêu cực là đôi khi nó làm cho việc

phát triển một thuật toán hoàn hảo dùng các kỹ thuật xử lý bằng máy tính khó khăn hơn. Ví dụ, một số thuật toán xử lý ảnh có khả năng làm giảm một lượng lớn nhiễu nén. Tuy nhiên, trong xử lý ảnh cũng có khi gây ra nhiễu nhân tạo. Nhưng ngay cả khi lượng nhiễu nhân tạo đưa vào ít hơn nhiều so với lượng nhiễu nền đã khử đi, thì nhiễu nhân tạo vẫn có thể gây sự chú ý của người xem, và người xem sẽ thí ch ảnh chưa xử lý hơn ảnh đã xử lý.

Các hiện tượng thị giác thảo luận trong phần trước, có thể giải thích đơn giản, ít ra thì cũng giải thích một cách định tính; nhưng nhiều hiện tượng thị giác khác không thể giải thích đơn giản, một phần là do chúng ta thiếu kiến thức. Ví dụ, hiện tượng thị giác có liên quan một phần đến xử lý ở mức trung tâm thì chưa thể giải thích cho vừa ý được. Hình 1.24 biểu diễn một phác họa chỉ gồm một số ít nét. Chưa hiểu được tại sao chúng ta liên hệ được ảnh này với Einstein. Nhưng ví dụ này cũng chứng minh được là, sự phác thảo những nét chính của đối tượng là quan trọng để nhận dạng được nó. Điều này có thể khai thác trong các ứng dụng như nhận dạng đối tượng bằng ảnh vi tính và phát triển hệ thống điện thoại video tốc độ bít thấp cho người điếc.



**Hình 1.24** Phác họa mặt Einstein bằng một số ít nét vẽ.

Các hiện tượng đã thảo luận ở trên liên quan đến sự cảm nhận ánh sáng chiếu sáng liên tục. Khi ánh sáng chiếu gián đoạn, sự cảm nhận của chúng ta phụ thuộc nhiều vào tần số ánh sáng. Xét ánh sáng phát ra trong những xung rất ngắn N lần trên giây.

Khi N nhỏ, có thể nhận ra là những ánh chớp rời nhau. Khi N tăng lên, ánh sáng nhấp nháy làm người quan sát khó chịu. Khi chúng ta tăng N hơn nữa, khó nhận ra là ánh sáng đang nhấp nháy và người quan sát có thể không phát hiện được cường độ sáng biến thiên theo thời gian. Tân số tại điểm mà người quan sát cảm nhận ánh sáng phát liên tục gọi là tân số ánh sáng nhấp nháy tối hạn (critical flicker frequency) hoặc tân số hoà đồng (fusion). Tân số hoà đồng tăng khi kích cỡ và tổng cường độ phát sáng của nguồn nhấp nháy tăng. Tân số hoà đồng có thể nhỏ bằng vài chu kỳ/giây với nguồn sáng nhỏ lờ mờ, và có thể vượt quá 100 chu kỳ/giây với nguồn sáng kích cỡ lớn và cường độ sáng lớn hơn.

Ánh sáng gián đoạn thông dụng trong cuộc sống hàng ngày. Ánh sáng huỳnh quang không phát liên tục, nhưng nhấp nháy với tốc độ đủ cao (hơn 100 lần/giây) nên tạo ra cảm giác liên tục trong những điều kiện quan sát điển hình. Tránh sự cảm nhận ánh sáng nhấp nháy là yếu tố quan trọng quyết định tốc độ phục hồi màn đèn hình (đèn tia âm cực, cathode ray tube, - CRT ). Như sẽ thảo luận sâu hơn trong tiết 4, màn hiển thị đèn hình chỉ được chiếu sáng trong một khoảng thời gian ngắn. Để một ảnh được hiển thị liên tục người xem không cảm thấy nháy, màn hình phải được phục hồi ở tốc độ đủ cao. Chẳng hạn, màn hiển thị đèn hình được làm sáng 60 lần/giây. Với tỉ lệ ken dòng 2:1, như trong hệ truyền hình NTSC, điều này ứng với 30 màn/giây. Với ảnh động hiện 24 màn/giây, một màn hiện 2 lần. Như vậy tốc độ nhấp nháy hiệu dụng là 48 màn/giây. Với điện ảnh, khi chiếu phim trong rạp rất tối nên làm giảm tân số hoà đồng xuống dưới 40 chu kỳ/giây. Vì lý do này, xem ảnh động không thấy nháy, ngay cả khi màn hình tối đến gần một nửa thời gian.

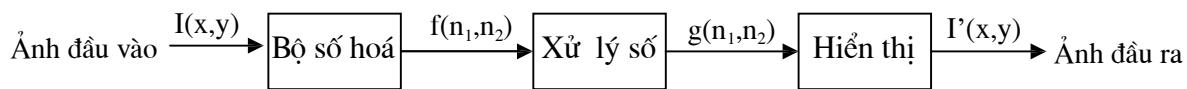
Tuy mỗi màn của điện ảnh hoặc truyền hình không động đậy thật sự, và chỉ một số có hạn màn được chiếu trong một giây, các vật trong cảnh xuất hiện với chuyển động liên tục. Hiệu ứng này gọi là sự thể hiện dịch chuyển (*motion rendition*), liên quan chặt chẽ đến hiện tượng *phi* ( $\phi$ ). Xét hai nguồn sáng xung động (pulsating) ở cách nhau gần  $1^0$  của góc nhìn từ người quan sát . Khi mỗi nguồn loé lên 1 ms và thời gian cách nhau 10 ms, ta thấy như ánh sáng đổi ngôi liên tục từ nguồn nọ sang nguồn kia. Khi thời gian cách nhau chỉ là 1 ms, ta thấy như hai nguồn loé sáng đồng thời. Khi thời gian cách nhau lớn hơn 1 giây, ta thấy được là hai nguồn sáng nhấp nháy riêng biệt. Đó là *hiện tượng phi*.

Nói chung, tốc độ mành đủ cao để tránh ánh sáng nhấp nháy cũng lại thích hợp với sự thể hiện dịch chuyển. Việc một vật thể hiện chuyển động liên tục trên màn ảnh hoặc màn truyền hình không nhất thiết đòi hỏi tốc độ lấy mẫu trong miền thời gian phải lớn hơn tốc độ Nyquist. Với các vật sự chuyển động đủ nhanh thì lấy mẫu theo miền thời gian 24 lần/giây hoặc 30 lần/giây là thấp hơn tốc độ Nyquist nhiều, và xảy ra hiệu ứng ánh giả tạm thời (temporal aliasing). Hiệu ứng này không phải bao giờ cũng gây ra gián đoạn chuyển động. Trong phim, đôi khi chúng ta thấy bánh xe chuyển động liên tục, nhưng ngược chiều. Trong trường hợp này, thể hiện được sự dịch chuyển, nhưng có hiệu ứng ánh giả tạm thời (temporal aliasing) nghiêm trọng xảy ra. Hiểu biết hiện tại của chúng ta về hiệu ứng ánh sáng lặp loè, sự dịch chuyển, hiệu ứng ánh giả tạm thời (temporal aliasing), và mối tương quan giữa chúng còn rất hạn chế. Hiểu toàn diện chủ đề này sẽ có ích trong một số ứng dụng chẳng hạn như giảm tốc độ bít bằng bỏ một số khung trong một chuỗi khung ảnh.

## 4. HỆ THỐNG XỬ LÝ ẢNH

### 4.1 Tổng quan hệ thống xử lý ảnh

Một hệ xử lý ảnh điển hình có xử lý số tín hiệu được biểu diễn trong hình 1.25. Nguồn ảnh đầu vào  $I(x,y)$  là một vật hoặc một cảnh tự nhiên, nhưng nó cũng có thể được tạo ra bởi một hệ thống khác, như bộ lọc, đèn hình CRT hoặc máy viđeô catxet (VCR). Bộ số hoá chuyển tín hiệu nguồn đầu vào thành tín hiệu điện mà biên độ biểu diễn cường độ ảnh và sau đó đem số hoá nó bằng bộ chuyển đổi tương tự -số (A/D).



**Hình 1.25.** Hệ thống xử lý ảnh điển hình.

Sau đó chuỗi  $f(n_1, n_2)$  ở đầu ra bộ số hoá được xử lý bởi một giải thuật (algôrith) xử lý ảnh số. Algôrith này có thể cài trên máy tính, bộ vi xử lý, hoặc một phần cứng chuyên dụng. Algôrith cụ thể được sử dụng phụ thuộc vào mục đích xử lý, có thể là cải thiện ảnh, phục hồi ảnh, mã hoá ảnh, lý giải ảnh, hoặc một tổ hợp bất kỳ của chúng.

Kết quả xử lý được hiển thị cho người dùng xem, nhưng đôi khi cũng đưa vào đầu vào một hệ thống khác. Bộ hiển thị được dùng là đèn hình CRT, nhưng cũng có thể là bức ảnh hoặc băng video catxet. Nếu kết quả là một biểu diễn bằng ký hiệu như trong lý giải ảnh, thì thiết bị hiển thị có thể là máy in.

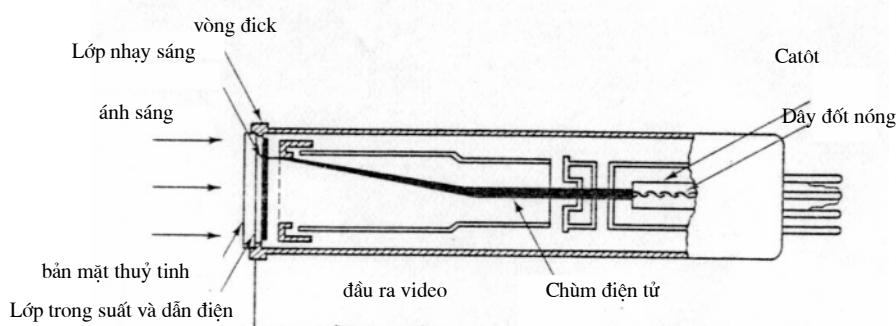
## 4.2 Bộ số hoá

Bộ số hoá chuyển nguồn ảnh đầu vào thành tín hiệu điện và lấy mẫu tín hiệu điện, sử dụng bộ chuyển đổi A/D. Các chức năng cụ thể của bộ số hoá phụ thuộc nguồn ảnh đầu vào. Khi đầu vào là tín hiệu điện, như trường hợp băng video catxet VCR, bộ số hoá giao diện với nguồn đầu vào và được sử dụng để lấy mẫu tín hiệu điện tùy theo khuôn dạng được dùng trong khâu chuyển đổi nguồn đầu vào thành tín hiệu điện.

Khi nguồn đầu vào là ảnh, một camera điện tử chuyển ảnh thành tín hiệu điện, và kết quả được số hoá bằng bộ chuyển đổi A/D. Trong một số hệ thống camera, các đường dẫn song song cho phép đo đồng thời cường độ ánh sáng tại nhiều điểm trong không gian. Tuy nhiên, trong hệ điển hình chỉ có một đường, và tại một thời điểm đã cho chỉ có thể đo cường độ ánh sáng tại một điểm. Trong trường hợp này, dùng một bộ quét để tuần tự quét hết bề mặt ảnh đầu vào. Trong hầu hết bộ quét, một khe nhỏ quét ảnh theo một lộ trình nhất định gọi là trườn g quét (raster). Cường độ sáng nhận được trên bề mặt khe nhỏ được chuyển thành tín hiệu điện và đo, giá trị đo được coi là với cường độ ảnh tại điểm không gian đó. Quá trình này có thể xem như phép nhân chập cường độ ảnh đầu vào  $I(x,y)$  với khe rồi lấy mẫu tích chập. Như vậy, hiệu quả của khe là lọc thông thấp  $I(x,y)$ . Điều đó giới hạn độ phân giải không gian  $I(x,y)$  và có thể sử dụng để làm giảm méo dạng bậc thang (antialiasing), cần thiết trong bộ biến đổi A/D. Với ảnh tĩnh, ảnh được quét một lần, nhưng cũng có thể quét nhiều lần và lấy trung bình các khung để giảm nhiễu. Với ảnh động, ảnh được quét trong những khoảng thời gian có tính tuần hoàn.

Khi đầu vào là phim hoặc bức ảnh, thiết bị được sử dụng để chuyển cường độ ảnh thành tín hiệu điện là một bộ quét bằng điểm sáng lướt. Một điểm sáng nhỏ quét nguồn đầu vào, và ánh sáng mà bức ảnh phản chiếu (hoặc truyền qua phim) được một bề mặt có gắn các bộ tách sóng quang thu gom. Nguồn chấm sáng nhỏ là màn hình CRT. Trong tiết 4.3 thảo luận chi tiết hơn về CRT.

Khi nguồn ảnh đầu vào là một vật hoặc một cảnh tự nhiên, thiết bị thông dụng nhất để chuyển cường độ ánh sáng thành tín hiệu điện là vidicon và họ hàng của nó như Saticon và Newvicon. Vidicon và các thiết bị cùng họ được dùng đến tận đầu những năm 1980 trong tất cả mọi ứng dụng truyền hình, bao gồm camera quảng bá, camera vidêô xách tay, camera theo dõi. Cấu trúc của một camera vidicon biểu diễn trong hình 1.26. Tại cuối ống vidicon (phía bên trái trong hình) ở bên trong vỏ bọc thuỷ tinh là một bản ảnh. Bản này có hai lớp. Lớp đối mặt với ánh sáng từ nguồn đầu vào là một lớp mỏng ôxít thiếc, trong suốt với ánh sáng nhưng dẫn điện.



**Hình 1.26.** Cấu trúc camera vidicon.

Lớp thứ hai đối mặt với súng điện tử, có phủ một lớp vật liệu nhạy sáng. Ở vidicon cơ sở vật liệu này là antimony trisulfide. Ánh sáng từ nguồn đầu vào đi qua các thấu kính quang học là cơ cấu hội tụ, thông qua bản thuỷ tinh phẳng và qua lớp thứ nhất của bản ảnh. Ánh sáng được hội tụ trên lớp thứ hai. Bản ảnh nhạy sáng (lớp thứ hai) được quét bởi súng điện tử và dòng điện nhận được là tín hiệu ra của camera, được một bộ chuyển đổi A/D số hoá. Lộ trình quét là từ phải sang trái, từ dưới lên trên. Vì ảnh nguồn đầu vào bị các thấu kính đảo ngược, nên trường quét này tương đương quét từ trái sang phải từ trên xuống dưới trong mặt phẳng ảnh đầu vào.

Lớp nhạy sáng là bán dẫn, có tính cách điện khi không có ánh sáng chiếu vào. Khi ánh sáng đập vào lớp này, điện tử di chuyển sang trong lớp ôxit thiếc dẫn điện tạo ra điện tích dương trên mặt bản ảnh quay về phía súng điện tử. Số điện tử chuyển động, hoặc nói cách khác là số điện tích dương đổi điện với súng điện tử biểu diễn cường độ ảnh tại điểm không gian đó. Khi chùm điện tử năng lượng thấp từ súng điện tử quét lên bản ảnh, nó phóng đủ số điện tử để trung hoà với các điện tích dương. Dòng điện phóng

này được một vòng bia bằng kim loại dẫn điện nối liền với lớp ôxit thiếc gom lại. Dòng điện tại vòng bia kim loại là tín hiệu camera. Điện tử xuất phát từ cực catốt, - ở đầu cuối bên kia của đèn vidicon. Các thấu kính tĩnh điện và từ hối tụ các điện tử thành một chùm hẹp.

Đáp ứng phổ của vidicon cơ bản với ảnh đen trắng tương tự như hàm hiệu suất chói tương đối C.I.E đã được thảo luận trong phần 4.2. Với ảnh màu, một camera màu dùng phương pháp quang học tách ánh sáng đầu vào thành các thành phần đỏ, lục và lam. Mỗi thành phần được đưa đến đầu vào của một camera vidicon. Như vậy, một camera màu chứa 3 ống vidicon riêng biệt.

Tín hiệu camera đại biểu cho cường độ nguồn ảnh đầu vào, được bộ biến đổi A/D lấy mẫu để tạo thành ảnh số. Cỡ ảnh số thông dụng là 128x128, 256x256, 512x512 và 1024x1024 điểm ảnh (pixels). Khi chúng ta giảm số pixels thì độ phân giải không gian, - cũng gọi là độ nét, bị giảm và các chi tiết trong ảnh bắt đầu biến mất. Biên độ mỗi pixel được lượng tử hoá 256 mức (biểu diễn bằng 8 bit). Thông thường, mỗi mức biểu thị bằng một số nguyên, 0 tương ứng với mức tối nhất và 256 tương ứng với mức sáng nhất. Khi chúng ta giảm số mức lượng tử hoá biên độ, nhiều lượng tử phụ thuộc tín hiệu bắt đầu xuất hiện, thoát đầu như là nhiễu ngẫu nhiên rồi sau đó như là những đường viền giả tạo. Với ảnh màu, mỗi thành phần đỏ, lục và lam được lượng tử hoá 8 bit/pixel, tổng số là 24bit/pixel.

Vidicon và các thiết bị cùng họ được gọi là các cảm biến quang dẫn hoặc đèn cảm biến (tube sensors), và được dùng cho đến đầu những năm 1980 trong tất cả các ứng dụng truyền hình. Từ giữa những năm 1980, có sự phát triển vượt bậc trong lĩnh vực sensor thể rắn. Trong một sensor thể rắn điển hình, các phân tử cảm biến xếp thành một mảng hai chiều (2-D) tích hợp trên một chíp. Mỗi phân tử sensor được đặt tại một vị trí pixel và cảm nhận cường độ ánh sáng tại pixel đó, một cơ cấu quét đọc các giá trị cường độ ở các pixel.

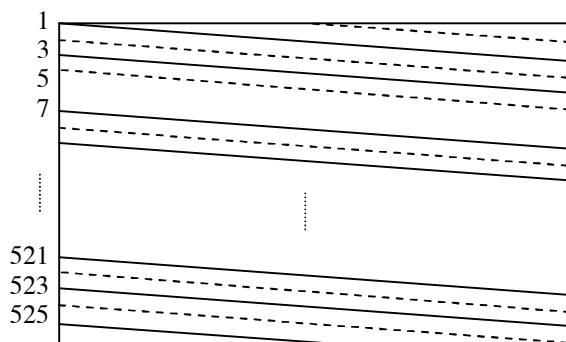
CCD (charge coupled device) là một ví dụ của phân tử sensor thể rắn. Khi một mảng CCD hướng về ánh sáng, các gói điện tích tỷ lệ với cường độ sáng phát triển. Các gói điện tích được dịch đến một mảng CCD lưu trữ không hướng về phía ánh sáng. Giá trị cường độ sáng được đọc từ mảng lưu trữ. Các phương pháp khác nhau đã được phát triển để đọc giá trị cường độ sáng ở mảng lưu trữ, tùy theo cấu hình cụ thể của mảng nhận ảnh và mảng lưu trữ trong CCD.

Các sensor thể rắn có nhiều thuận lợi hơn các sensor quang dẫn. Chúng nhỏ gọn và bền vững hơn. Sensor thể rắn có cấu trúc rõ rệt và vị trí mỗi pixel được biết chính xác trong cả không gian và thời gian. Kết quả là, phép xử lý để trích ra màu sắc đơn giản hơn và có thể đạt được sự đồng đều màu sắc. Sensor thể rắn cũng có độ nhạy về cảm nhận quang cao hơn nhiều. Trong một sensor quang dẫn điển hình, từng pixel một được một sensor ánh sáng đơn kiểm tra. Trong ứng dụng truyền hình điển hình thời gian phát hiện ánh sáng đo bằng micro giây và độ nhạy về cảm nhận ánh sáng thấp. Trong cảm biến (sensor) thể rắn điển hình dùng một mảng các phân tử cảm biến (sensor), ứng với một pixel là một phân tử. Do vậy, với mỗi bức ảnh chỉ cần kiểm tra mỗi phân tử sensor một lần. Năng lượng sáng có thể được tích phân suốt thời gian của một khung chứ không phải một pixel, làm tăng tiềm năng cảm nhận ánh sáng. Sensor thể rắn cũng có hệ số trễ nhỏ hơn sensor quang dẫn. Sự trễ (lag) là bức xạ còn dư của sensor sau khi cường độ sáng đã thay đổi hoặc nguồn sáng đã dời chỗ.

Sensor thể rắn có một số nhược điểm so với sensor quang dẫn. Một là về độ phân giải không gian: muốn có độ phân giải không gian cao hơn thì số pixel phải lớn hơn, nghĩa là số phân tử sensor cần tích hợp trên một chip phải nhiều hơn. Hai là tỷ số tín hiệu trên nhiễu thấp. Mặc dù có một số nhược điểm nhưng công nghệ sensor thể rắn vẫn đang phát triển nhanh chóng, và các sensor thể rắn chắc chắn sẽ thay thế các sensor quang dẫn trong hầu hết các ứng dụng truyền hình trong tương lai gần.

Trong hệ truyền hình quảng bá NTSC, mỗi giây truyền đi 30 khung hình. Mỗi khung gồm 525 dòng quét ngang, mỗi khung chia thành 2 mành, mành chẵn và mành lẻ. Mỗi mành đều được quét từ trái sang phải và từ trên xuống dưới, và có 262,5 dòng ngang. Mành chẵn gồm các dòng đánh số chẵn, và mành lẻ gồm các dòng đánh số lẻ. Các đường ngang trong mành chẵn và lẻ được xen vào nhau tạo thành khung. Điều này được biểu diễn trong hình 1.27 và được gọi là tỷ lệ ken dòng 2:1. Sử dụng tỷ lệ ken dòng 2:1 là để cho độ phân giải đọc bằng 525 dòng trên mỗi khung, với tốc độ 30 khung/giây, nhưng tần số nhấp nháy là 60 chu kỳ/giây, làm cho ta không thấy nháy trên màn hình máy thu. Không có sự ken dòng thì một khung phải hiển thị hai lần để đạt được tần số nhấp nháy là 60 chu kỳ/giây. Điều này yêu cầu phải lưu trữ khung hoặc phải có dải thông lớn hơn. Độ phân giải không gian của một khung truyền hình hiển thị một ảnh tĩnh tương tự như một ảnh số 512x512 pixel. Trong truyền hình quảng bá, tín hiệu là analog ở cả máy phát và máy thu. Xử lý số các tín hiệu đó bao gồm việc lấy mẫu khi dùng bộ biến đổi A/D.

Đầu ra của bộ số hoá là một dãy số. Mặc dù trong hình 1.25 đầu ra được biểu diễn bằng một dãy 2-D là  $f(n_1, n_2)$ , trong thực tế đầu ra cũng có thể là 3 dãy  $f_R(n_1, n_2)$ ,  $f_G(n_1, n_2)$  và  $f_B(n_1, n_2)$ , tương ứng các thành phần đỏ, lục và lam của ảnh màu. Đầu ra cũng có thể là dãy 3 chiều  $f(n_1, n_2, n_3)$ , là hàm của hai biến không gian và một biến thời gian đại biểu cho một dãy khung. Các tín hiệu đó được xử lý bằng algôrithm xử lý ảnh số, có thể thực hiện trên máy tính, vi xử lý hoặc một phần cứng chuyên dụng.



**Hình 1.27:** Mành chǎn (đường nét liền) và mành lẻ (đường nét chấm) trong tỷ lệ kẽn dòng 2:1.

### 4.3. Hiển thị.

Thiết bị hiển thị thông dụng nhất trong môi trường xử lý ảnh là CRT. Một CRT bao gồm một súng điện tử và một màn hình phốt pho, như biểu diễn trên hình 1.28. Súng điện tử tạo ra chùm điện tử được hội tụ trên một vùng hẹp của màn hình phốt pho nhờ các thấu kính tĩnh điện, sử dụng cả trường tĩnh điện và trường từ. Chùm điện tích kích thích phốt pho phát sáng. Với ống hình đơn sắc, sử dụng một chùm. Với ống hình màu, sử dụng sử dụng ba chùm riêng biệt, mỗi cái kích thích phốt pho ứng với một trong ba màu phốt pho. Màn hình được quét từ trái sang phải và từ trên xuống dưới. Các chất phốt pho là kim loại nhẹ, như kẽm, trong dạng sulfide và sulfate. Vật liệu phốt pho được xử lý để có những phân tử rất mịn, trát ở phía trong bản thuỷ tinh. Với ống hình đơn sắc, phốt pho phủ một lớp đều. Với ống hình màu, phốt pho được lắng đọng thành những điểm hoặc những lớp dọc cho từng màu.

Khi các chùm điện tử tốc độ cao kích thích phốt pho, các điện tử trong nguyên tử phốt pho chuyển lên mức năng lượng cao. Khi các chùm điện tử chuyển đến điểm khác

trên màn hình, thì các điện tử nhảy về mức năng lượng thấp và phát ra ánh sáng. Bức xạ ánh sáng phát ra từ màn hình gọi là sự phát qua ng. Khi ánh sáng bị dập tắt, thì màn hình phát huỳnh quang. Thời gian để ánh sáng màn hình suy giảm xuống còn 1% giá trị cực đại gọi là thời gian dư huy của màn hình (screen persistence). Với thời gian dư huy trung bình hoặc ngắn, thường dùng cho màn hình tivi, thời gian suy giảm (decay) khoảng 5 ms. Có những loại phôt pho có thời gian dư huy dài, nhưng như vậy có thể gây ra sự chập hai khung, kết quả là sẽ làm nhòe ở mức độ đáng kể chuỗi ảnh có chuyển động.

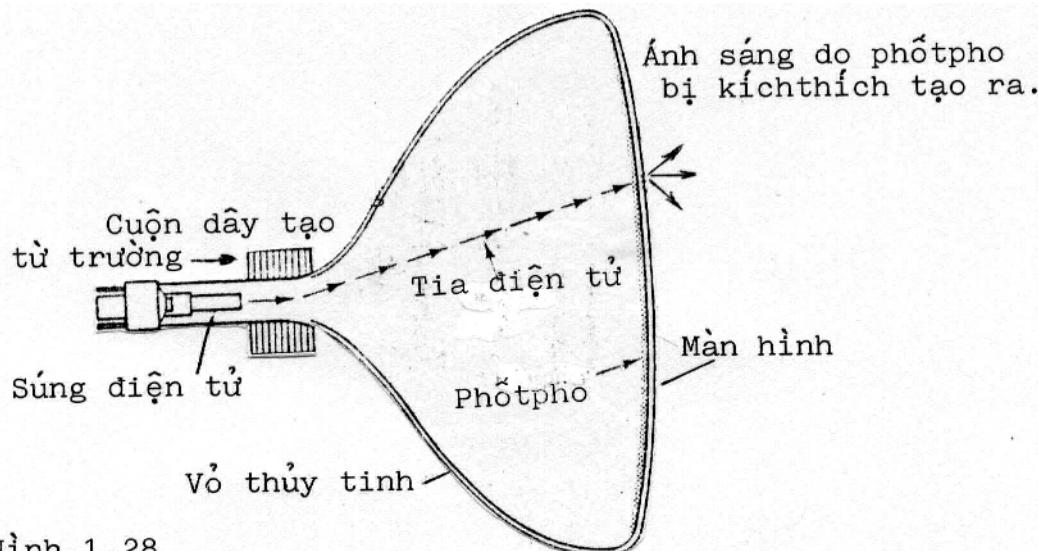
Vì thời gian dư huy của màn hình ngắn, nên hầu hết CRT dùng làm hiển thị đều chỉ sáng lên trong một khoảng thời gian rất ngắn khi chùm điện tử quét qua. Để hiển thị ảnh tĩnh hoặc một chuỗi ảnh, màn hình phải được làm tươi. Màn hình hiển thị thường có RAM bán dẫn để phục hồi. Bộ nhớ phục hồi (refresh memory) có khả năng giữ trên một khung hình với cỡ thích hợp, được dùng để phục hồi màn hình.

Màn hiển thị sử dụng trong môi trường xử lý ảnh là loại chất lượng cao. Một màn hình điển hình có tỉ số cạnh chiều (rộng trên chiều cao) là 1 và được định cỡ để hiển thị khung vuông. Cỡ hiển thị chuẩn là 512x512 và 1024x1024, nhưng các cỡ khác, chẳng hạn như 640x480 và 1024x1280 cũng được chấp nhận. Tốc độ phục hồi là 30 khung/giây với tỉ lệ ken dòng 2:1, giống truyền hình quảng bá. Các dòng quét đơn trên màn hình chất lượng cao phải sắc nét. Do đó, dòng ngang chỉ có thể xuất hiện trong màn lẻ hoặc trong màn chẵn, và tốc độ nhấp nháy với dòng quét sẽ là 30 chu kỳ/giây. Với tốc độ này, có thể thấy rõ sự nhấp nháy. Để tránh vấn đề này, có những màn hiển thị được phục hồi với tốc độ 50 khung/giây hoặc 60 khung/giây không ken dòng. Điều đó làm cho màn hiển thị không nháy và người xem dễ chịu hơn.

Trong nhiều màn hiển thị, bộ nhớ phục hồi không chỉ dùng phục hồi màn hình mà còn dùng cho xử lý. Hình ảnh được hiển thị có thể được phóng to, thu nhỏ, cuộn, hoặc xử lý bằng một bộ lọc đơn trong bộ nhớ phục hồi theo thời gian thực. Vì phần cứng cải tiến liên tục, nên khối hiển thị sẽ có được nhiều khả năng tinh tế hơn.

Một hình ảnh hiển thị trên màn hình CRT được coi là “mềm”, theo nghĩa là nó chỉ có tính chất nhất thời (temporary). Có một cách tiếp cận để có một bản sao “cứng” là đem chụp ảnh trên màn hình lên phim nhựa. Điều này có thể thực hiện bằng những camera có gắn thêm bộ phận ngắm hiện hình lên màn CRT nhỏ. Ngoài ra, cũng còn

nhiều loại thiết bị phụ khác tạo ra bản sao “cứng”, kể cả việc tạo ra cùng một lúc nhiều bản.



Hình 1.28

### Giải thích thêm về hiện tượng aliasing

- Theo “The IEEE Standard Dictionary of Electrical and Electronics Terms” (6<sup>th</sup> edition, 1996):

**Aliasing:** Sự diễn đạt sai đối với thị giác (visual misrepresentation) xảy ra khi ảnh chứa đựng nhiều chi tiết hơn khả năng diễn đạt mà độ phân giải của thiết bị hiển thị cho phép.

*Ghi chú:* Kết quả của aliasing là đường nằm nghiêng hiện ra thành đường lởm chởm bậc thang.

**Anti-aliasing:** (1) Theo định lý Nyquist, tần số cao nhất có thể phục hồi được bằng nửa tần số lấy mẫu. Aliasing xảy ra khi tồn tại những tần số cao hơn nửa giá trị tần số lấy mẫu. Kết quả là những tần số cao này xuất hiện “giả mạo” như những thành phần tần số thấp. Để khắc phục phải cho qua lọc thông thấp để khử hết những tần số cao hơn các thành phần mà tần số lấy mẫu cho phép phục hồi.

(2) Kỹ thuật để khử hiệu ứng thị giác do aliasing gây ra.

- Theo “Prentice Hall’s Illustrated Dictionary of Computing” (2<sup>nd</sup> edition, 1995) thì:

Aliasing là “Sự xuất hiện (ngoài ý muốn) trên màn hình những đường mép lởm chởm răng cưa viền quanh ảnh khi dùng những tệp đồ họa do máy tính tạo ra. Aliasing thể hiện đặc biệt rõ khi ta vạch một đường chéo mà nó lại hiện ra dưới dạng một đường bậc thang (đường răng cưa)”

- Theo “English-Chinese Dictionary of Information Science” (1994)

**Aliasing** 1. Hiện tượng tín hiệu giả. 2. Hiệu ứng bậc thang

**Aliasing effect** Hiệu ứng chồng chéo (Giao địệp hiệu ứng)

38 bis

## CHƯƠNG 2

### CẢI THIỆN ẢNH

#### v MỞ ĐẦU

Cải thiện ảnh là quá trình xử lý để cải thiện thể hiện của ảnh đối với cho người xem, hoặc để cải thiện một hệ xử lý ảnh khác các phương pháp và mục tiêu thay đổi tùy theo ứng dụng. Khi ảnh được cải thiện cho người xem như ở truyền hình, mục đích là cải thiện sự cảm thụ: chất lượng ảnh, độ dễ hiểu hoặc thể hiện đối với thị giác. Trong ứng dụng khác như dùng máy nhận dạng đối tượng, ảnh được tiền xử lý để hỗ trợ cho máy. Vì mục tiêu cải thiện ảnh phụ thuộc vào bối cảnh ứng dụng, và tiêu chí cải thiện thường là chủ quan hoặc quá phức tạp cho nên khó đổi ra thành những phép đo khách quan hữu dụng. Algorit cải thiện ảnh vì vậy có xu hướng đơn giản, định lượng và không theo thể thức (ad hoc). Ngoài ra, trong một ứng dụng đã có sẵn algorit xử lý tốt cho loại ảnh này không nhất thiết cũng tốt cho loại ảnh khác.

Cải thiện ảnh liên quan mật thiết tới phục hồi ảnh, điều đó sẽ được thảo luận trong chương 3. Khi ảnh bị xuống cấp, cải thiện ảnh thường đem lại kết quả là phục hồi ảnh gốc. Tuy nhiên vẫn có một vài sự khác nhau quan trọng giữa phục hồi ảnh và cải thiện ảnh. Trong phục hồi ảnh, khi một ảnh lý tưởng bị xuống cấp thì mục tiêu là làm cho ảnh qua xử lý càng giống ảnh gốc càng tốt. Trong cải thiện ảnh, mục tiêu là làm cho ảnh được xử lý tốt hơn ảnh chưa xử lý theo một nghĩa nào đó. Trong trường hợp này, ảnh lý tưởng phụ thuộc vào bối cảnh của vấn đề và thường không được định nghĩa rõ ràng. Để minh họa sự khác nhau này, lưu ý rằng ảnh gốc không xuống cấp không còn gì để phục hồi nữa, nhưng vẫn có thể đem cải thiện để tăng độ nét bằng cách cho qua bộ lọc thông cao.

Trong một số bối cảnh cải thiện ảnh là điều mong muốn. Trong một lớp ván đề quan trọng, ảnh được cải thiện bằng cách thay đổi độ tương phản hoặc dải động. Chẳng hạn, một ảnh điển hình dấu không xuống cấp cũng sẽ có thể hiện tốt hơn khi các đường

biên ảnh được làm sắc nét hơn. Tương tự, khi một ảnh có dải động lớn được ghi vào trong một môi trường với dải động hẹp như phim hoặc giấy thì độ tương phản và do đó cả các chi tiết của ảnh sẽ bị giảm, đặc biệt trong những vùng rất tối và rất sáng. Ảnh chụp từ máy bay bị giảm độ tương phản khi cảnh bị mây hoặc sương mù bao phủ. Khi đó, làm tăng mức tương phản cục bộ và làm giảm dải động toàn bộ sẽ có ý nghĩa đáng kể về cải thiện ảnh.

Một vấn đề khác trong cải thiện ảnh, là ảnh bị xuống cấp có thể được cải thiện bằng cách làm giảm sự xuống cấp. Các ví dụ về xuống cấp của ảnh là mờ, nhiễu nền ngẫu nhiên lớn, nhiễu lốm đốm và nhiễu lượng tử. Trong lĩnh vực này cải thiện trùng với phục hồi ảnh. Một algorit đơn giản và phi thê thức (ad hoc), không khai thác các đặc tính của tín hiệu và sự xuống cấp, thường được coi là một algorit cải thiện ảnh. Một có algorit tính toán học cao hơn và phức tạp hơn, có khai thác các đặc tính của tín hiệu và sự xuống cấp, có tiêu chí sai số rõ ràng để so sánh ảnh được xử lý với ảnh gốc chưa xuống cấp, thường được coi là một algorit phục hồi. Sự phân biệt này khá mơ hồ và tùy ý. Nhưng cần phải đưa ra một số quyết định tùy ý để phân chia một vài đề mục giữa chương này với chương sau (chương Phục hồi ảnh).

Ta biết rằng đường biên là một đối tượng chứa rất nhiều thông tin quan trọng, có thể dùng trong những ứng dụng lý giải ảnh. Bước đầu tiên trong ứng dụng đó là tiên xử lý một ảnh thành một bản đồ đường biên. Vì sự phát hiện đường biên của ảnh chính xác hơn sẽ cải thiện chất lượng của hệ lý giải ảnh khai thác thông tin đó, cho nên việc đổi ảnh thành bản đồ đường biên của nó có thể xem như một quá trình cải thiện ảnh.

Một lớp quan trọng khác trong cải thiện ảnh là hiển thị dữ liệu hai chiều (2-D), dữ liệu này có thể đại biểu cho cường độ của ảnh, cũng có thể không. Một ảnh có độ phân giải thấp  $128 \times 128$  pixel có thể làm vừa ý thị giác của người xem hơn bằng cách đếm nội suy để tạo ra ảnh lớn hơn, ví dụ  $256 \times 256$  pixel. Trong phép ước lượng phổ 2-D, các giá trị ước lượng của phổ thường được hiển thị thành bản đồ đường biên. Mặc dù dữ liệu (2-D) như vậy không phải là ảnh theo đúng nghĩa thường hiểu, nhưng vẫn có thể

biểu diễn chúng như ảnh. Có thể hiển thị chúng như ảnh trắng - đen, có khi cải thiện thêm bằng màu, cố để cho thể hiện tốt hơn và thông tin nó mang theo được diễn đạt rõ ràng hơn. Trong những ứng dụng khác, như ảnh radar hồng ngoại, có cả thông tin về cự ly cũng như cường độ ảnh. Đem thể hiện thông tin về cự ly bằng màu có thể nêu bật cự ly tương đối của các đối tượng trong ảnh. Thậm chí chất lượng ảnh tốt cũng có thể được cải thiện bằng cách cố tình gây một số méo dạng. Chẳng hạn, khi một

đối tượng trong ảnh được tô màu giả thì có thể làm nổi bật đối tượng đối với người xem.

Trong chương này, ta nghiên cứu các phương pháp cải thiện ảnh đã thảo luận ở trên: tiết 1 bàn về thay đổi độ tương phản và dải động, tiết 2 bàn về làm trơn nhiễu, tiết 3 bàn về phát hiện đường biên ảnh. Trong tiết 4 thảo luận về các phương pháp nội suy ảnh và sự ước lượng chuyển động, có thể sử dụng cho nội suy ảnh. Tiết 5 bàn về cải thiện ảnh bằng phương pháp giả mâu.

## 1. THAY ĐỔI ĐỘ TƯƠNG PHẢN VÀ DẢI ĐỘNG

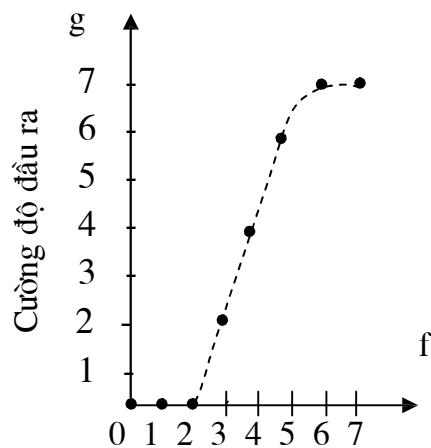
### 1.1 THAY ĐỔI MỨC XÁM

Thay đổi mức xám là phương pháp đơn giản và có hiệu quả để thay đổi độ tương phản hoặc dải động của ảnh. Trong phương pháp này, mức xám hoặc mức cường độ của ảnh đầu vào  $f(n_1, n_2)$  được thay đổi theo một phép biến đổi xác định. Phép biến đổi  $g = T[f]$ , là quan hệ giữa cường độ ảnh đầu vào  $f$  với cường độ ảnh đầu ra  $g$  được biểu diễn bởi một hình vẽ hoặc một bảng. Ta hãy xem một minh họa đơn giản của phương pháp này. Hình 2.1(a) là ảnh  $4 \times 4$  pixel với mỗi pixel được biểu diễn bằng 3 bit, vậy là có 8 mức, gồm  $f = 0$ (mức tối nhất), 1, 2, 3,..., 7(mức sáng nhất). Phép biến đổi liên hệ giữa cường độ đầu vào với cường độ đầu ra được biểu diễn bằng đồ thị hoặc bảng số như trong Hình 2.1(b). Với mỗi pixel trong ảnh đầu vào có pixel tương ứng trong ảnh đầu ra, nhận được từ đồ thị hoặc bảng số trong Hình 2.1(b). Kết quả được biểu diễn trên Hình 2.1(c). Bằng cách chọn phép biến đổi phù hợp có thể thay đổi được độ tương phản hoặc dải động.

Phép biến đổi cụ thể phụ thuộc vào ứng dụng. Trong một số ứng dụng, việc lựa chọn phép biến đổi căn cứ vào tính chất vật lý. Chẳng hạn khi bộ hiển thị có đặc tính phi tuyến thì mục đích của biến đổi là bù phi tuyến. Trong trường hợp đó, phép biến đổi phù hợp được xác định từ đặc tính phi tuyến của bộ hiển thị.

3	3	4	4
2	3	4	5
2	3	4	5
2	3	4	5

(a)



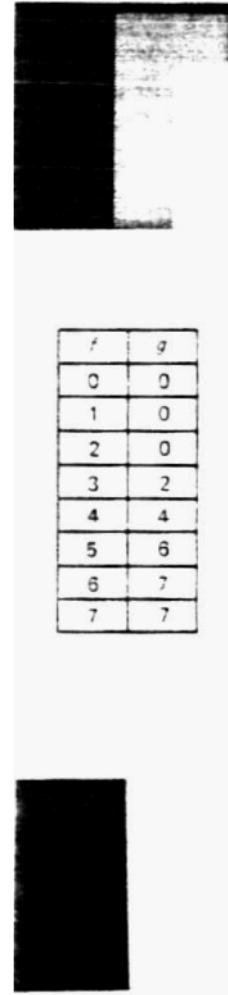
(b)

f	g
0	0
1	0
2	0
3	2
4	4
5	6
6	7
7	7



2	2	4	4
0	2	4	6
0	2	4	6
0	2	4	6

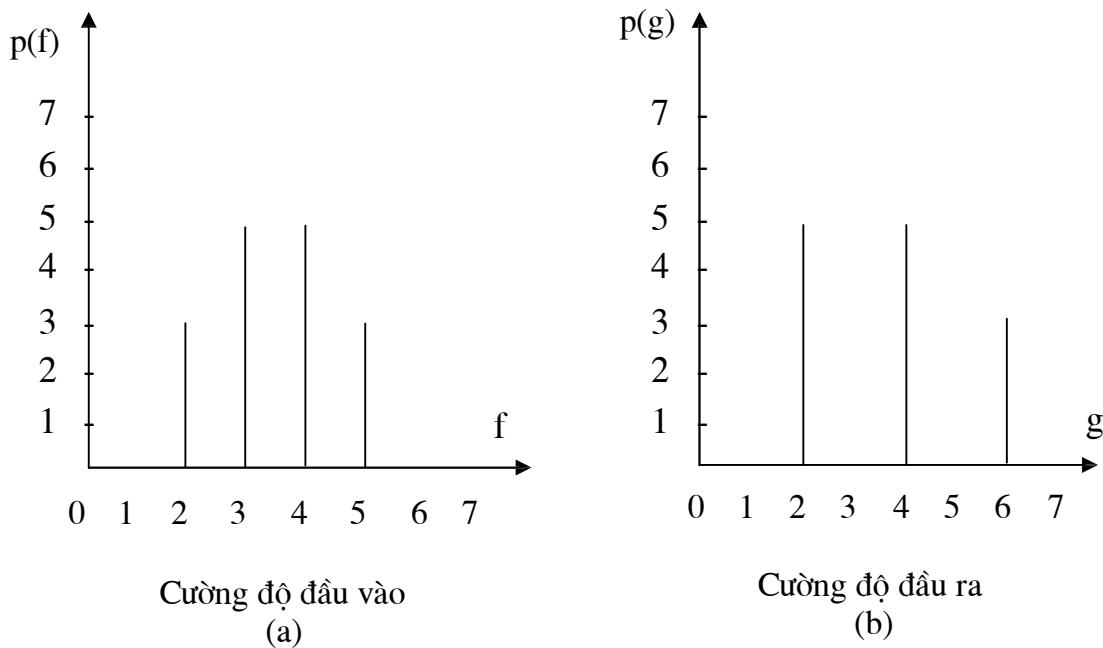
(c)



**Hình 2.1:** Ví dụ về thay đổi mức xám.

- (a) Ảnh  $4 \times 4$  pixel, mỗi pixel được biểu diễn bằng 3 bit;
- (b) Hàm biến đổi mức xám;
- (c) Kết quả thay đổi ảnh trong hình (a) khi sử dụng hàm biến đổi mức xám trong hình (b).

Trong những ứng dụng thường gặp, có thể nhận được phép biến đổi tốt bằng cách tính tổ chức đồ (histogram) của ảnh đầu vào và nghiên cứu đặc tính của nó. Tổ chức đồ của ảnh, ký hiệu là  $p(f)$ , đại biểu cho số pixel có một cường độ nhất định  $f$ , là một hàm của  $f$ . Chẳng hạn, ảnh  $4 \times 4$  pixel trên Hình 2.1(a) có tổ chức đồ là Hình 2.2(a). Tổ chức đồ hiển thị một vài đặc tính quan trọng của ảnh giúp ta xác định được phép biến đổi mức xám mong muốn. Trên Hình 2.2(a) cường độ ảnh đư ợc tụt lại trong một vùng nhỏ thì dải động không được sử dụng tốt. Trong trường hợp đó, dùng phép biến đổi trong Hình 2.1(b) sẽ làm tăng dải động toàn bộ và ảnh sau khi biến đổi có độ tương phản cao hơn. Hình 2.2(b), là tổ chức đồ của ảnh đã xử lý ở Hình 2.1(c), đã chứng tỏ điều đó.

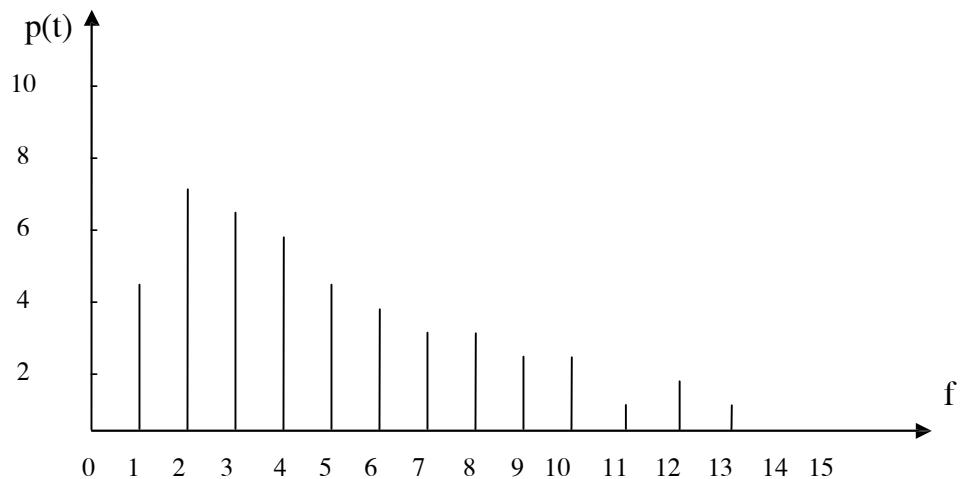


**Hình 2.2:** Tổ chức đồ của ảnh  $4 \times 4$  pixel:

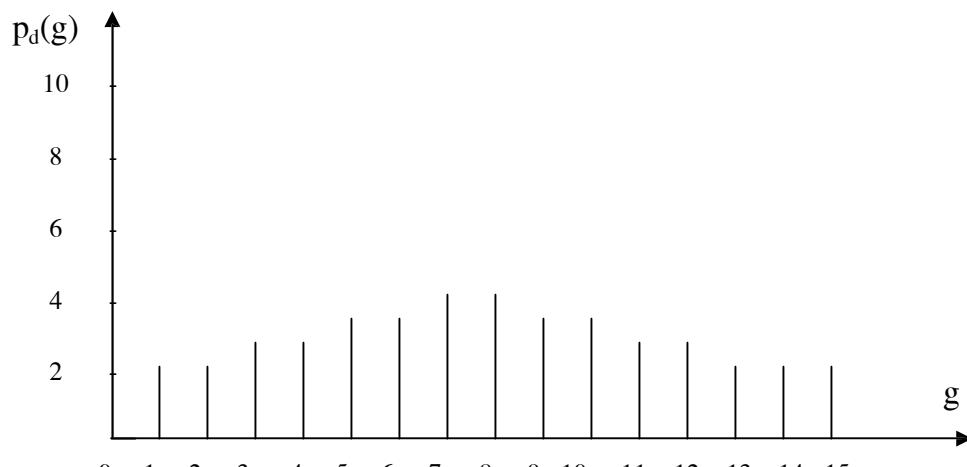
- (a) Ảnh trong Hình 2.1(a);
- (b) Ảnh trong Hình 2.1(c).

Vì việc tính tổ chức đồ của một ảnh và thay đổi mức xám bằng một phép biến đổi mức xám đã cho không cần phải tính toán nhiều, cho nên trong thực tế phép biến đổi mức xám mong muốn có thể do một kỹ thuật viên có kinh nghiệm xác định trên thời gian thực. Trên cơ sở việc tính toán tổ chức đồ ban đầu, kỹ thuật viên chọn phép biến đổi mức xám để tạo ra ảnh được xử lý. Bằng cách nhìn vào ảnh được xử lý và tổ chức đồ của ảnh, kỹ thuật viên có thể chọn một phép biến đổi mức xám khác và nhận được một ảnh đã xử lý mới, cứ thế tiếp tục cho đến khi nhận được ảnh đầu ra vừa ý.

Khi xét thấy kỹ thuật viên phải xử lý quá nhiều ảnh, thì cần tự động hóa việc chọn phép biến đổi mức xám. Trong trường hợp này phương pháp gọi là thay đổi tổ chức đồ rất có lợi. Với phương pháp này, người ta chọn phép biến đổi mức xám có tổ chức đồ mong muốn cho từng ảnh một. Tổ chức đồ mong muốn của ảnh đầu ra, ký hiệu là  $p_d(g)$ , có ích cho những ảnh thường gặp loại ảnh có giá trị cực đại ở vùng giữa dải động và giảm chậm khi cường độ tăng hoặc giảm. Với một ảnh đã cho, ta muốn xác định hàm biến đổi sao cho ảnh đầu ra có tổ chức đồ giống như  $p_d(g)$ . Vấn đề này có thể xem như một bài toán sơ đẳng về lý thuyết xác suất. Thông thường tổ chức đồ  $p(f)$  và  $p_d(g)$  theo thứ tự có thể coi như hàm mật độ xác suất theo một thang tỷ lệ nào đó của các biến ngẫu nhiên  $f$  và  $g$ . Chẳng hạn  $p(3)/16$  trong Hình 2.2(a) là xác suất để một pixel được chọn ngẫu nhiên trong ảnh  $4 \times 4$  pixel ở Hình 2.1(a) có mức cường độ là 3. Ta muốn tìm một biến đổi  $g = T[f]$  với điều kiện ràng buộc là  $T[f]$  phải là một hàm đơn điệu không giảm của  $f$ , sao cho  $p(g) \approx p_d(g)$ . Một cách tiếp cận để giải quyết bài toán xác suất này là nhận được các hàm phân bố xác suất  $P(f)$  và  $P_d(g)$  bằng cách lấy tích phân các hàm mật độ xác suất  $p(f)$  và  $p_d(g)$  và sau đó chọn hàm biến đổi sao cho  $P(f) \approx P_d(g)$  ở  $g = T[f]$ . Đặt điều kiện ràng buộc  $T[f]$  phải là một hàm đơn điệu không giảm là để đảm bảo rằng, một pixel với cường độ cao hơn pixel khác thì trong ảnh đầu ra nó sẽ không trở thành một pixel có cường độ thấp hơn.



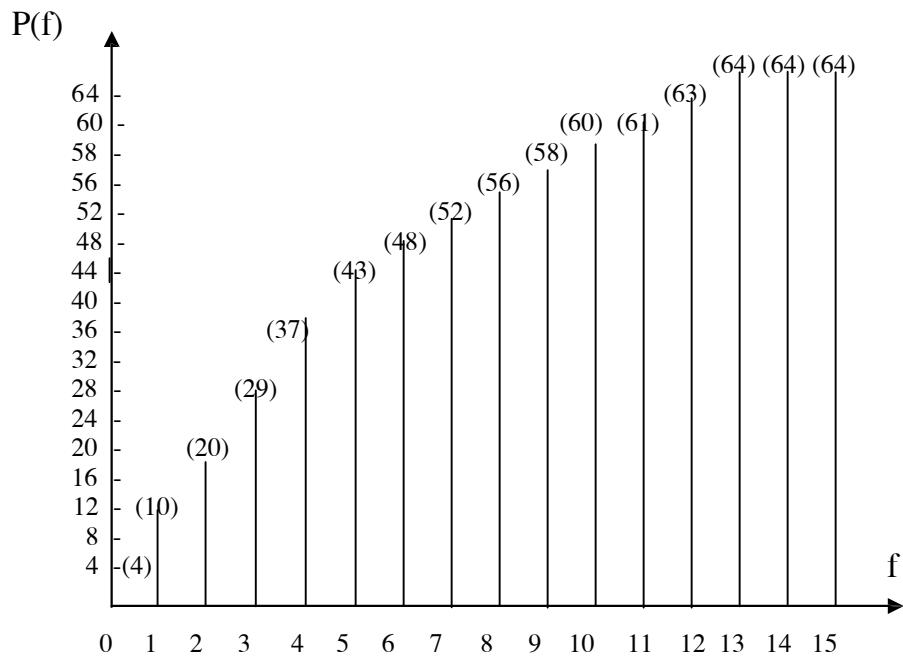
Cường độ đầu vào  
(a)



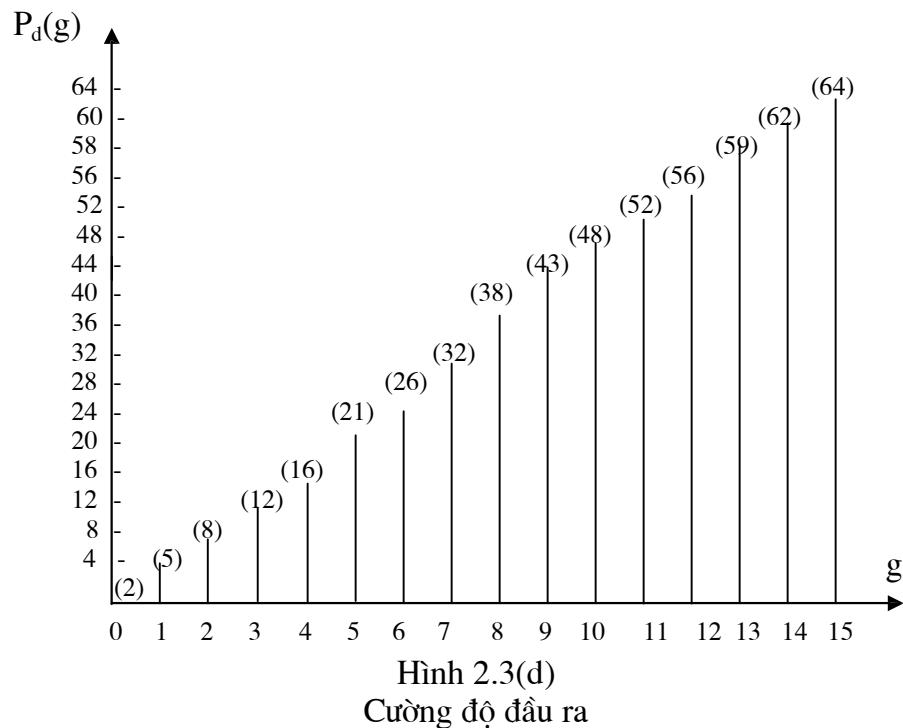
Cường độ đầu ra  
(b)

**Hình 2.3:** Tổ chức đồ và tổ chức đồ tích luỹ.

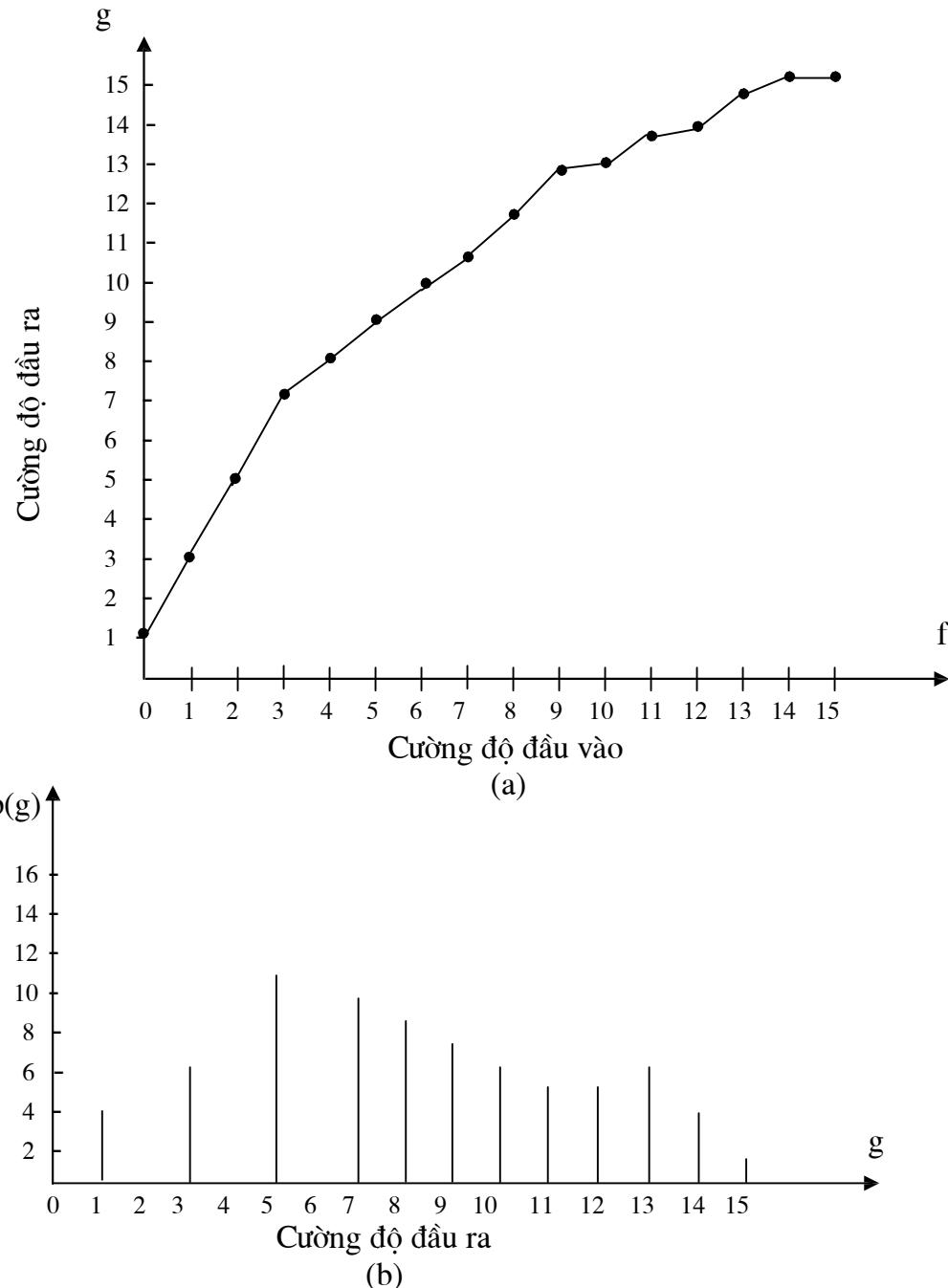
- (a) Tổ chức đồ ảnh 8 x 8 pixel;
- (b) Tổ chức đồ mong muốn;
- (c) Tổ chức đồ tích luỹ suy diễn từ hình (a);
- (d) Tổ chức đồ tích luỹ suy diễn từ hình (b).



Hình 2.3(c)  
Cường độ đầu vào



Hình 2.3(d)  
Cường độ đầu ra



Hình 2.4:

- (a) Hàm biến đổi mức xám biến đổi gần đúng tổ chức đồ trong Hình 2.3(a) thành tổ chức đồ mong muốn trong Hình 2.3(b);
- (b) Tổ chức đồ của ảnh biến đổi mức xám nhận được bằng cách áp dụng hàm biến đổi trong hình (a) cho một ảnh có tổ chức đồ như trên Hình 2.3(a).

Áp dụng cách tiếp cận này cho bài toán thay đổi tổ chức đồ bao gồm các biến f và g rời rạc, thuật toán ta tính các tổ chức đồ luỹ tích  $P(f)$  và  $P_d(g)$  từ  $p(f)$  và  $p_d(g)$  bằng:

$$P(f) = \sum_{k=0}^f p(k) = P(f-1) + p(f) \quad (2.1a)$$

$$P_d(g) = \sum_{k=0}^g p_d(k) = P_d(g-1) + p_d(g) \quad (2.1b)$$

Hình 2.3 biểu diễn một ví dụ về tổ chức đồ luỹ tích. Hình 2.3(a) và (b) cho ví dụ của  $p(f)$  và  $p_d(g)$ , Hình 2.3(c) và (d) cho  $P(f)$  và  $P_d(g)$  nhận được bằng cách sử dụng (2.1). Từ  $P(f)$  và  $P_d(g)$ , có thể nhận được hàm biến đổi mức xám  $g = T[f]$  bằng cách chọn  $g$  cho từng giá trị  $f$  sao cho  $P_d(g) \approx P(f)$ . Hàm biến đổi mức xám nhận được từ Hình 2.3 được biểu diễn trên Hình 2.4(a), tổ chức đồ của ảnh nhận được từ phép biến đổi đó được biểu diễn trên Hình 2.4(b). Nếu giữ nguyên tổ chức đồ mong muốn  $p_d(g)$  phù hợp cho nhiều ảnh đầu vào khác nhau thì chỉ cần từ  $p_d(g)$  tính ra  $P_d(g)$  một lần mà thôi.

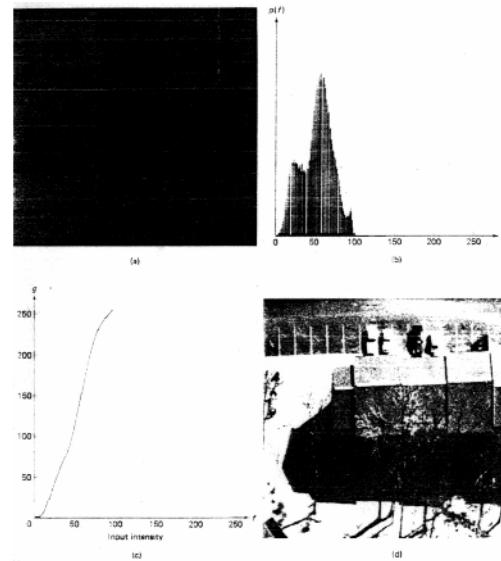
Trong ví dụ ta xét ở trên, lưu ý rằng tổ chức đồ của ảnh đã xử lý không giống tổ chức đồ mong muốn. Đó là trường hợp chung khi  $f$  và  $g$  là hai biến rời rạc và ta yêu cầu tất cả các pixel có cường độ đầu vào như nhau được ánh xạ vào một cường độ đầu ra như nhau. Cũng lưu ý rằng tổ chức đồ luỹ tích mong muốn  $P_d(g)$  gần như một đường thẳng. Một phép thay đổi tổ chức đồ đặc biệt được gọi là san bằng (equalisation) tổ chức đồ, tổ chức đồ nhận được là một hằng số. Khi đó tổ chức đồ luỹ tích sẽ là một đường thẳng. Ảnh xử lý bằng quân bằng tổ chức đồ có độ tương phản cao hơn ảnh chưa xử lý, nhưng trông có vẻ không tự nhiên.

Tuy phép thay đổi mức xám về khái niệm cũng như về tính toán là đơn giản nhất, nó vẫn đem lại cho người xem kết quả khả quan trong cải thiện chất lượng ảnh hoặc độ dễ hiểu, nhờ đó thường được sử dụng nhiều trong các ứng dụng xử lý ảnh. Điều này được minh họa bằng hai ví dụ. Hình 2.5(a) biểu diễn một ảnh gốc  $512 \times 512$  pixel, với mỗi pixel được biểu diễn bằng 8 bít. Hình 2.5(b) biểu thị tổ chức đồ của ảnh trong Hình 2.5(a). Tổ chức đồ cho thấy rõ là một số lượng lớn các pixel ảnh được tập trung ở những mức cường độ thấp trong dải động, nghĩa là trong những vùng tối ảnh sẽ thể hiện rất tối và suy giảm độ tương phản. Bằng cách tăng độ tương phản trong vùng tối thì có thể làm cho các chi tiết của ảnh rõ hơn. Điều này được thực hiện bằng cách sử dụng hàm biến đổi biểu diễn trên Hình 2.5(c). Ảnh được xử lý bằng hàm trong Hình 2.5(c)

được biểu diễn trên Hình 2.5(d), tổ chức đồ của nó như trên Hình 2.5(e). Một ví dụ khác được biểu diễn trên Hình 2.6. Trên Hình 2.6(a) là ảnh gốc còn trên Hình 2.6(b) là ảnh đã được xử lý độ tương phản.

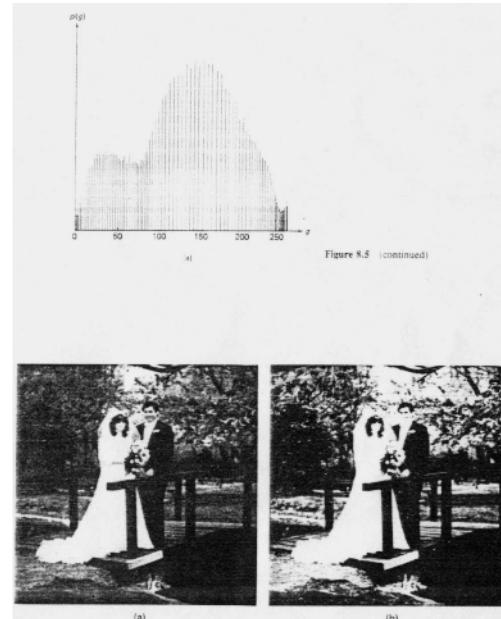
**Hình 2.5:** Ví dụ về thay đổi mức xám.

- (a) Ảnh gốc 256 x 256 pixels;
- (b) Tổ chức đồ của ảnh trong hình (a);
- (c) Hàm biến đổi được sử dụng trong sự biến đổi mức xám;
- (d) Ảnh đã xử lý;
- (e) Tổ chức đồ của ảnh đã xử lý trong hình (d).



**Hình 2.6:** Ví dụ về thay đổi mức xám.

- (a) Ảnh gốc 512 x 512 pixels;
- (b) Ảnh đã xử lý.



Phương pháp thay đổi tổ chức đồ được thảo luận ở trên cũng có thể được áp dụng vào ảnh màu. Để cải thiện ảnh độ tương phản mà chỉ ảnh hưởng nhỏ tới màu sắc hoặc độ bão hòa, ta có thể biến đổi ảnh RGB  $f_R(n_1, n_2)$ ,  $f_G(n_1, n_2)$  và  $f_B(n_1, n_2)$  thành ảnh YIQ  $f_Y(n_1, n_2)$ ,  $f_I(n_1, n_2)$  và  $f_Q(n_1, n_2)$  bằng cách sử dụng biến đổi trong công thức (2.8). Sự thay đổi mức xám chỉ áp dụng với ảnh Y  $f_Y(n_1, n_2)$ , sau đó đem kết quả tổ hợp lại với  $f_I(n_1, n_2)$  và  $f_Q(n_1, n_2)$  không xử lý. Lại dùng biến đổi (2.8), nhận được ảnh đã xử lý RGB  $g_R(n_1, n_2)$ ,  $g_G(n_1, n_2)$  và  $g_B(n_1, n_2)$ . Trên Hình 2.7(a) là ảnh gốc 512x512 pixel và trên Hình 2.7(b) là ảnh đã được xử lý bằng biến đổi mức xám.

## 1.2. BỘ LỌC THÔNG CAO VÀ MẶT NẠ MỜ

Bộ lọc thông cao làm nổi bật các thành phần tần số cao của tín hiệu đồng thời làm giảm thành phần tần số thấp. Vì các đường biên hoặc chi tiết tinh vi trên ảnh góp phần chủ yếu trong việc tạo ra các thành phần số cao của ảnh, nên bộ lọc thông cao thường làm tăng độ tương phản cục bộ và làm cho ảnh sắc nét.

Mặt nạ mờ được các nghệ sĩ nhiếp ảnh biết đến từ lâu, có liên quan chặt chẽ với bộ lọc thông cao. Khi áp dụng mặt nạ mờ, ảnh gốc bị làm mờ sau đó lấy một phần của ảnh mờ che lấp ảnh nguồn. Điều đó được thực hiện bằng cách đem bản âm của ảnh mờ cộng với ảnh gốc. Ảnh đã xử lý bởi mặt nạ mờ có thể được biểu diễn bằng:

$$g(n_1, n_2) = af(n_1, n_2) - bf_L(n_1, n_2) \quad (2.2)$$

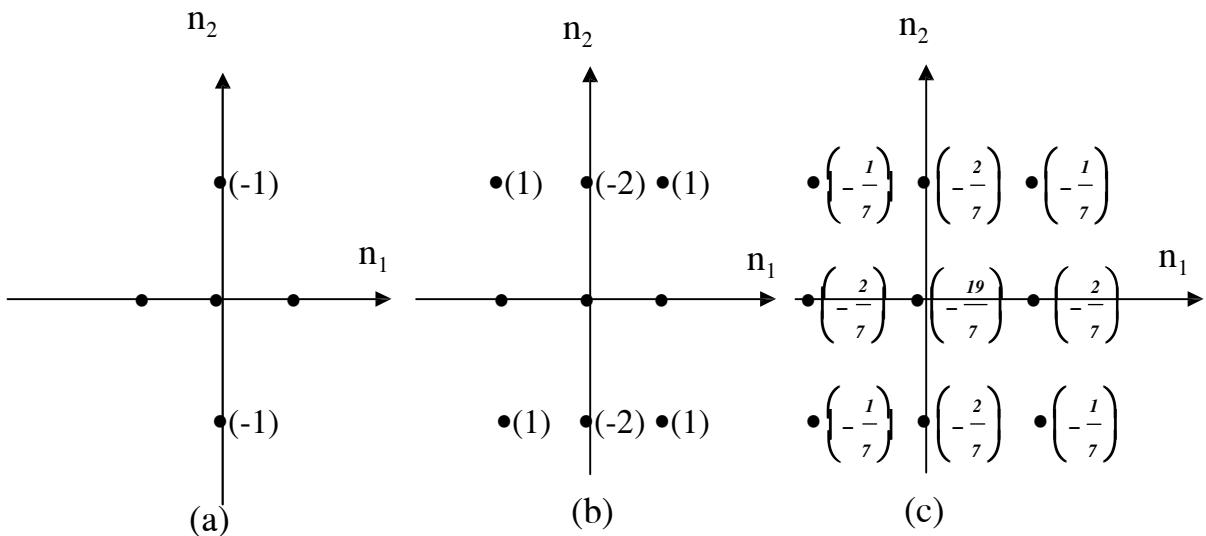
trong đó  $f(n_1, n_2)$  là ảnh gốc,  $f_L(n_1, n_2)$  là ảnh đã qua bộ lọc thông thấp hoặc ảnh mờ, a và b là các đại lượng vô hướng với  $a > b > 0$ ,  $g(n_1, n_2)$  là ảnh đã xử lý. Đem viết lại  $f(n_1, n_2)$  như là tổng của ảnh  $f_L(n_1, n_2)$  đã qua bộ lọc thông thấp và ảnh đã qua bộ lọc thông cao  $f_H(n_1, n_2)$ , ta có thể viết (2.2) là.

$$g(n_1, n_2) = (a-b)f_L(n_1, n_2) + a f_H(n_1, n_2) \quad (2.3)$$

từ (2.3) thấy rõ là các thành phần tần số cao được làm nổi bật so với thành phần tần số thấp và mặt nạ mờ là một dạng của bộ lọc thông cao.

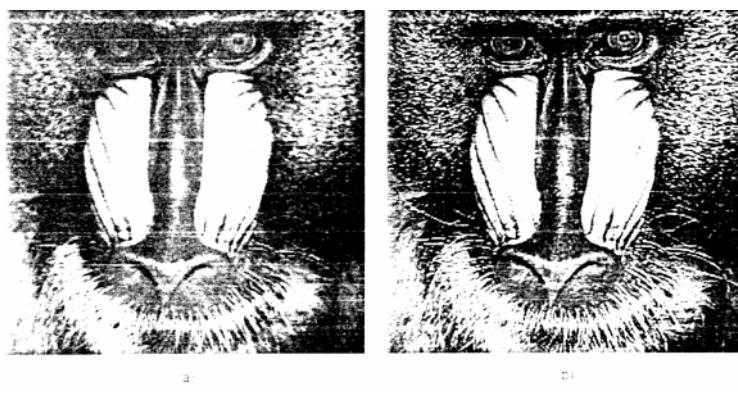
Một vài ví dụ điển hình về đáp ứng tần số của bộ lọc thông cao sử dụng để cải thiện độ tương phản được biểu diễn trên Hình 2.8. Một đặc tính chung của tất cả bộ lọc ở Hình 2.8 là tổng biên độ của mỗi đáp ứng xung là bằng 1 vì vậy đáp ứng tần số của bộ lọc  $H(\omega_1, \omega_2) = 1$  khi  $\omega_1 = \omega_2 = 0$  và cho thành phần một chiều đi qua trọn vẹn. Đặc tính này có hiệu quả là bảo tồn cường độ trung bình của ảnh gốc trong ảnh đã xử lý. Chú ý rằng đặc tính này bản thân không thể đảm bảo cường độ ảnh xử lý nằm trong khoảng [0, 255]. Nếu các giá trị cường độ của một vài pixel trong ảnh đã xử lý nằm ra ngoài phạm vi này chúng có thể bị ghim giá trị từ 0 tới 255 hoặc đặt lại thang độ ảnh để cường độ của tất cả các pixel thuộc ảnh đã xử lý đều nằm trong phạm vi từ 0 tới 255.

$$\begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline -2 & 5 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array} \quad \frac{1}{7} \cdot \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline -2 & 19 & -2 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$



**Hình 2.8:** Đáp ứng xung của các bộ lọc thông cao dùng cho cải thiện ảnh.

Hình 2.9 minh họa tính năng bộ lọc thông cao, Hình 2.9(a) là ảnh gốc  $256 \times 256$  pixel và Hình 2.9(b) là kết quả sử dụng bộ lọc thông cao trong Hình 2.9(a). Mặc dù ảnh gốc không bị xuống cấp, bộ lọc thông cao làm tăng độ tương phản cục bộ nhờ đó ảnh thể hiện sắc nét hơn. Tuy vậy, vì bộ lọc thông cao làm nổi bật các thành phần tần số cao, mà tạp âm nền thường có thành phần tần cao đáng kể cho nên lọc thông cao làm tăng công suất nhiễu nền. So sánh vùng nền Hình 2.9(a) và Hình 2.9(b) thấy rằng ảnh qua bộ lọc thông cao nhiều nhiễu hơn ảnh chưa qua xử lý. Sự nổi bật nhiễu nền là một hạn chế đối với bất kỳ algorit nào có tác dụng làm tăng độ tương phản tại chỗ và làm cho ảnh sắc nét.



**Hình 2.9:** Ví dụ về lọc thông cao.

(a)

(b)

(a) Ảnh gốc  $256 \times 256$  pixel;

(b) Ảnh đã qua bộ lọc thông cao.

### 1.3. XỬ LÝ ĐỒNG CẤU

Khi đem ảnh với một dải động lớn, chẳng hạn phong cảnh tự nhiên vào một ngày trời nắng, ghi trên một môi trường với dải động nhỏ như phim hoặc giấy, độ tương phản thường bị giảm, đặc biệt trong những vùng rất tối hoặc rất sáng. Một cách tiếp cận để cải thiện ảnh là làm giảm dải động và tăng độ tương phản cục bộ trước khi đem ghi trên một môi trường với dải động nhỏ.

Có một phương pháp đã được triển khai để làm giảm dải động và tăng độ tương phản cục bộ dựa trên việc áp dụng một hệ đồng cấu bằng phép nhân với một mô hình tạo ảnh. Ảnh thường được hình thành bởi sự ghi ánh sáng phản xạ từ một đối tượng được một nguồn quang chiếu sáng. Dựa trên sự quan sát này, mô hình toán của ảnh là

$$f(n_1, n_2) = i(n_1, n_2)r(n_1, n_2) \quad (2.4)$$

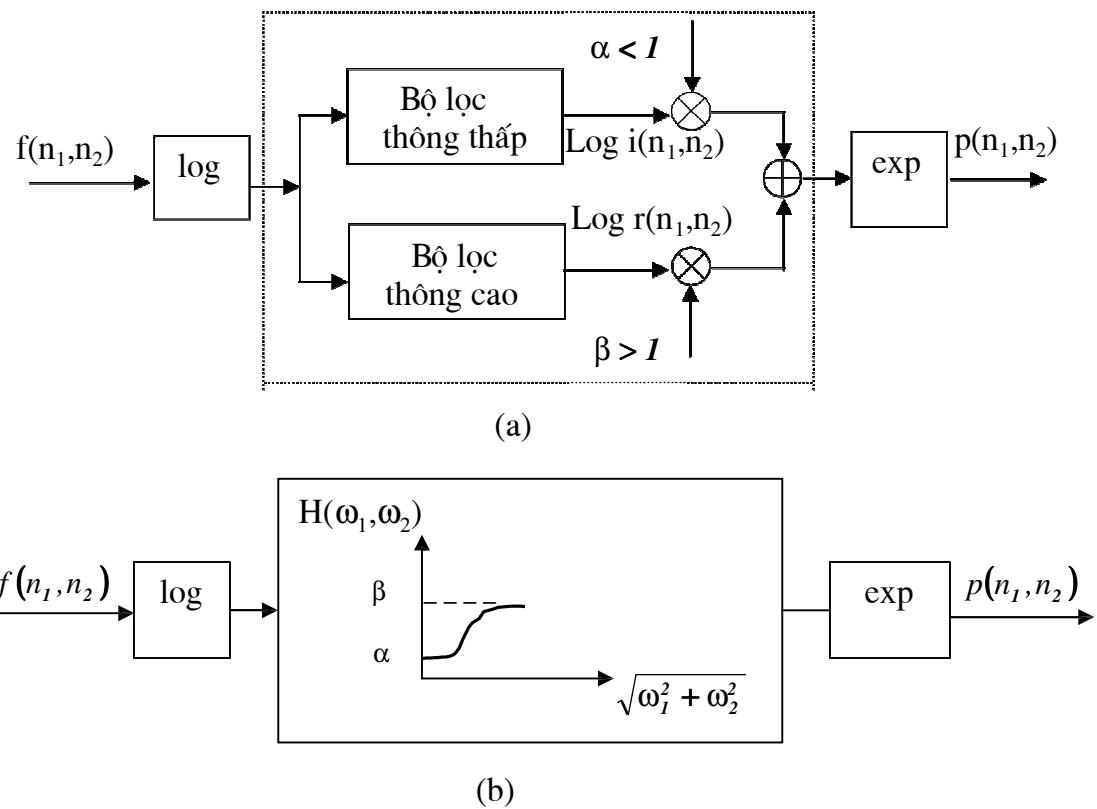
trong đó  $i(n_1, n_2)$  là đại biểu cho sự chiếu sáng và  $r(n_1, n_2)$  đại biểu cho sự phản xạ. Để ứng dụng hệ đồng cấu cho cải thiện ảnh, giả thiết rằng thành phần chiếu sáng  $i(n_1, n_2)$  là nhân tố chủ yếu ảnh hưởng tới dải động của ảnh, biến thiên chậm, còn thành phần phản xạ  $r(n_1, n_2)$  là nhân tố chủ yếu ảnh hưởng tới độ tương phản cục bộ của đối tượng lại biến thiên nhanh. Để giảm dải động và tăng độ tương phản cục bộ thì phải giảm  $i(n_1, n_2)$  và tăng  $r(n_1, n_2)$ . Để tách  $i(n_1, n_2)$  ra khỏi  $r(n_1, n_2)$  trong (2.4), ta lấy logarit cả hai vế của (2.4):

$$\log f(n_1, n_2) = \log i(n_1, n_2) + \log r(n_1, n_2) \quad (2.5)$$

Nếu giả thiết rằng  $\log i(n_1, n_2)$  vẫn thay đổi chậm và  $\log r(n_1, n_2)$  vẫn thay đổi nhanh, thì lọc thông thấp  $\log f(n_1, n_2)$  sẽ nhận được  $\log i(n_1, n_2)$ , còn lọc thông cao  $\log f(n_1, n_2)$  sẽ nhận được  $\log r(n_1, n_2)$ . Khi đã tách riêng được  $\log i(n_1, n_2)$  và  $\log r(n_1, n_2)$  thì cho suy giảm  $\log i(n_1, n_2)$  sẽ giảm được dải động, còn tăng  $\log r(n_1, n_2)$  sẽ làm tăng độ tương phản cục bộ. Sau đó đem  $\log i(n_1, n_2)$  và  $\log r(n_1, n_2)$  đã qua xử lý tổ hợp lại và đem mũ hoá (exponentiate) kết quả thì sẽ trở lại miền cường độ ảnh. Điều này được biểu diễn trên Hình 2.10(a). Hệ ở trong Hình 2.10(a) có thể đơn giản hoá bằng cách thay hệ bên trong đường vẽ chấm bằng bộ lọc thông cao tương ứng. Sơ đồ hệ đã đơn giản hoá như trong Hình 2.10(b). Một ví dụ minh họa tính năng của hệ này được biểu diễn trên Hình 2.11. Hình 2.11(a) là ảnh gốc  $256 \times 256$  pixel, Hình 2.11(b) là ảnh đã xử lý bằng hệ thống trên Hình 2.10(b).

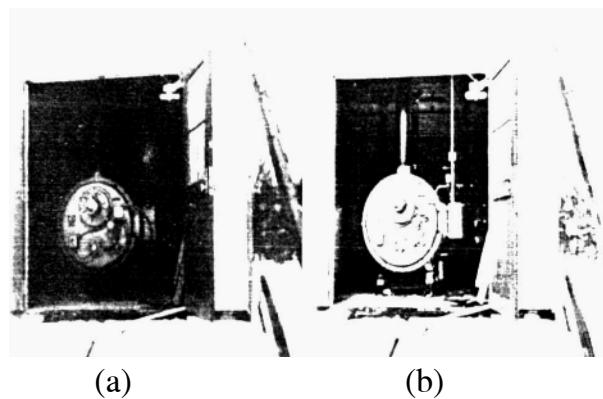
Một hệ như trên Hình 2.10, thực hiện lấy logarit rồi đến một thuật toán tuyến tính, cuối cùng mũ hoá được gọi là một hệ đồng cấu với phép nhân. Đó là nguồn gốc của các thuật ngữ xử lý đồng cấu (homomorphic processing) và lọc đồng cấu (homomorphic filtering).

Tóm lại algorit của bộ lọc đồng cấu là trước hết lấy logarithmic hai vế (2.4) sau đó lọc tuyến tính lại chuyển về ảnh cũ bằng phép mũ hoá.



**Hình 2.10:** Hệ thống đồng cấu dùng cho cải thiện ảnh.

- (a) Hệ thống đồng cấu dùng cho cải thiện độ tương phản và thay đổi dải động; (b) Hệ thống trong hình (a) sau khi đã đơn giản hóa.



**Hình 2.11:** Ví dụ về xử lý đồng cấu cho cải thiện ảnh.

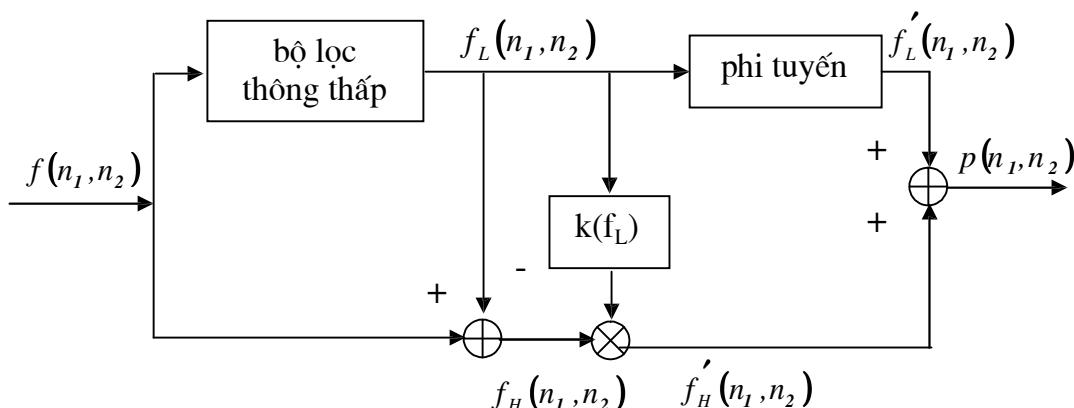
- (a) Ảnh gốc  $256 \times 256$  pixel;  
 (b) Ảnh đã xử lý bằng hệ thống đồng cấu cho phép nhân.

Mặc dù hệ trên Hình 2.10 đã được phát triển từ một mô hình (model) hình thành ảnh và một hệ đồng cấu, vẫn có thể coi nó đơn giản là một bộ lọc thông cao trong miền log cường độ. Tính năng lọc thông cao trong miền log cường độ phù hợp với tính năng hệ thị giác con người, đã được thảo luận trong tiết 1.2.2, ở mức ngoại vi của hệ thị giác con người cường độ ảnh bị biến đổi bởi một dạng phi tuyến giống như một toán tử lôgarit.

Như vậy, với hệ thị giác con người miền log cường độ quan trọng hơn miền cường độ.

#### 1.4. PHÉP THAY ĐỔI THÍCH NGHI ĐỘ TƯƠNG PHẢN CỤC BỘ VÀ GIÁ TRỊ TRUNG BÌNH ĐỘ CHÓI CỤC BỘ

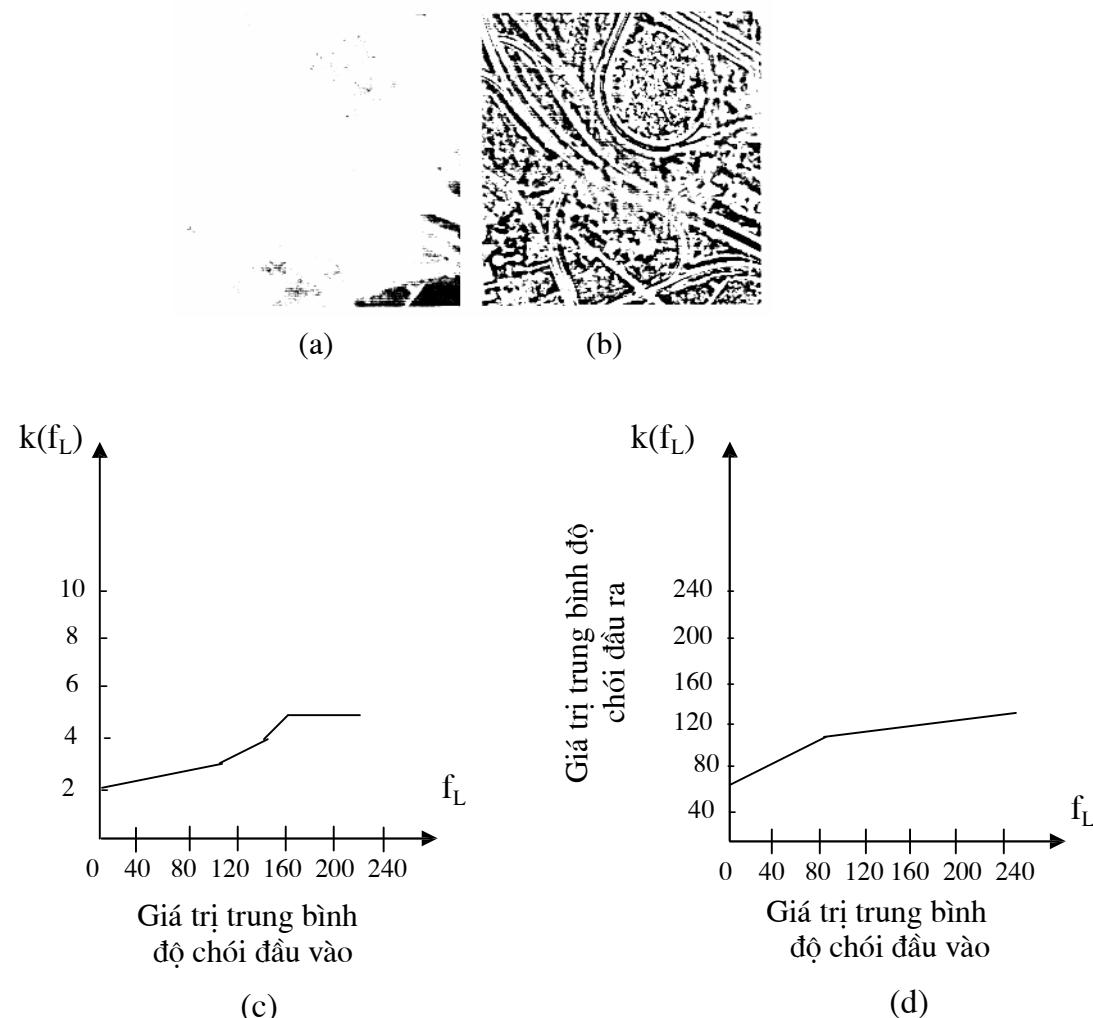
Trong một vài ứng dụng, muốn thay đổi độ tương phản cục bộ và giá trị trung bình độ chói cục bộ khi đặc tính cục bộ của ảnh thay đổi. Trong những ứng dụng như vậy dùng phép xử lý ảnh thích nghi là hợp lý.



**Hình 2.12:** Hệ thống để thay đổi độ tương phản cục bộ và giá trị trung vị độ chói cục bộ như là một hàm của giá trị trung vị độ chói.

Một ứng dụng cải thiện ảnh là thay đổi thích nghi độ tương phản cục bộ và giá trị trung bình độ chói cục bộ của ảnh chụp từ máy bay qua những độ dày thay đổi của lớp mây bao phủ. Theo một mô hình đơn giản của ảnh bị xuống cấp do lớp mây bao phủ, vùng ảnh ở khu vực bị mây che tăng giá trị trung bình độ chói cục bộ vì chịu ảnh hưởng của ánh sáng mặt trời phản xạ từ lớp mây và giảm độ tương phản cục bộ do tín

hiệu từ mặt đất bị suy hao khi đi qua lớp mây. Một cách tiếp cận để cải thiện ảnh là làm tăng độ tương phản cục bộ và giảm giá trị trung bình độ chói cục bộ bất cứ lúc nào phát hiện thấy mây bao phủ. Một cách để phát hiện lớp mây bao phủ là đo giá trị trung bình độ chói cục bộ. Khi giá trị trung bình độ chói cục bộ ở mức cao thì có thể là có mây bao phủ.



**Hình 2.13:** Ví dụ về cải thiện ảnh bằng phép lọc thích nghi.

- (a) Ảnh gốc 256 x 256 pixel chụp từ máy bay qua một lớp mây có độ dày thay đổi; (b) Kết quả xử lý ảnh trong hình (a) bằng hệ xử lý trên Hình 2.12;
- (c) Hàm  $k(f_L)$  sử dụng trong việc xử lý;
- (d) Độ phi tuyến sử dụng trong việc xử lý.

Một hệ để làm giảm ảnh hưởng của lớp mây bao phủ được biểu diễn trên Hình 2.12. Hệ này làm thay đổi độ tương phản cục bộ và giá trị trung bình độ chói cục bộ. Trong hình,  $f(n_1, n_2)$  là ảnh gốc, dãy  $f_L(n_1, n_2)$  là giá trị trung bình độ chói cục bộ của  $f(n_1, n_2)$  đạt được bằng cách cho đi qua bộ lọc thông thấp, dãy  $f_H(n_1, n_2)$  là độ tương phản cục bộ đạt được bằng cách lấy  $f_H(n_1, n_2) = f(n_1, n_2) - f_L(n_1, n_2)$ . Độ tương phản cục bộ được thay đổi bằng cách nhân  $f_H(n_1, n_2)$  với  $k(f_L)$ , - một đại lượng vô hướng hàm của  $f_L(n_1, n_2)$ . Độ tương phản đã thay đổi được ký hiệu là  $f'_H(n_1, n_2)$ . Nếu  $k(f_L) > 1$  thì độ tương phản cục bộ tăng, ngược lại độ tương phản cục bộ giảm. Giá trị trung bình độ chói tại chỗ được biến đổi bởi một điểm phi tuyến và độ chói trung bình tại chỗ đã thay đổi được ký hiệu là  $f'_L(n_1, n_2)$ . Độ tương phản cục bộ và giá trị trung bình độ chói cục bộ sau khi thay đổi được tổ hợp lại thành ảnh được xử lý là  $p(n_1, n_2)$ . Để tăng độ tương phản cục bộ và giảm giá trị trung bình độ chói cục bộ khi giá trị trung bình độ chói cục bộ cao, ta chọn  $k(f_L)$  lớn cho giá trị  $f_L$  lớn và chọn thuật toán phi tuyến, có xét đến sự thay đổi giá trị trung bình độ chói cục bộ và sự tăng độ tương phản. Hình 2.13 cho thấy kết quả ứng dụng hệ trong Hình 2.12 để cải thiện ảnh chụp từ máy bay qua sự thay đổi của lượng mây bao phủ. Hình 2.13(a) là ảnh gốc  $256 \times 256$  pixel, Hình 2.13(b) là ảnh đã xử lý. Hàm  $k(f_L)$  và thuật toán phi tuyến đã sử dụng được biểu diễn trên Hình 2.13(c) và 2.13(d).

Hệ trong Hình 2.12 có thể được xem như một trường hợp đặc biệt của xử lý hai kênh. Ảnh xử lý được chia làm hai thành phần, độ tương phản cục bộ và giá trị trung bình độ chói cục bộ, hai thành phần này được thay đổi riêng rẽ, sau đó tổ hợp kết quả lại. Trong hệ ở Hình 2.12 giá trị trung bình độ chói cục bộ được thay đổi bởi thuật toán phi tuyến và độ tương phản cục bộ được thay đổi bởi hệ số nhân  $k(f_L)$ . Trong chương 4 và 5 sau này, ta sẽ thấy phép xử lý hai kênh cũng tỏ ra rất hiệu quả trong phục hồi và mã hoá ảnh.

Khái niệm thích nghi một hệ cải thiện ảnh để làm thay đổi các đặc tính cục bộ, nói chung là một ý tưởng rất hay, có thể đem áp dụng cho những bối cảnh khác nhau. Chẳng hạn phép biến đổi mức xám hay lọc thông cao đã thảo luận ở tiết trên, có thể thay đổi cho thích nghi với sự biến thiên các đặc tính cục bộ. Mặc dù hệ thích nghi thường yêu cầu tính toán nhiều hơn hệ không thích nghi, nói chung tính năng hệ thích nghi được đánh giá là tốt hơn. Khi phải giải quyết bài toán xử lý ảnh với yêu cầu chất lượng cao, nên nghĩ đến các hệ thích nghi. Hệ thích nghi cũng rất hiệu quả trong phục hồi cũng như mã hoá ảnh.

## 2. LÀM TRƠN NHIỀU

Ngoài các biện pháp cải thiện ảnh bằng thay đổi độ tương phản và dải động còn có thể cải thiện ảnh bằng các biện pháp làm giảm những sự xuống cấp có thể xảy ra. Cải thiện ảnh trong lĩnh vực này trùng với phục hồi ảnh. Trong tiết này, ta thảo luận algorit đơn giản làm giảm nhiễu ngẫu nhiên hay nhiễu muối-tiêu. Algorit này yêu cầu tính toán nhiều và phức tạp hơn.

### 2.1. BỘ LỌC THÔNG THẤP

Năng lượng của một ảnh điển hình tập trung chủ yếu ở các thành phần tần số thấp. Đó là do độ tương quan lớn về không gian giữa các pixel lân cận.

Năng lượng của những nguồn làm cho ảnh xuống cấp như nhiễu ngẫu nhiên dải rộng thường trải rộng ra trong miền tần số. Bằng cách làm giảm các thành phần tần số cao trong khi giữ nguyên các thành phần tần số thấp, bộ lọc thông thấp giảm nhiễu rất nhiều mà chỉ làm giảm tín hiệu chút ít.

Bộ lọc thông thấp cũng có thể sử dụng cùng với bộ lọc thông cao trong xử lý ảnh trước khi bị nhiễu làm xuống cấp. Trong mã hoá ảnh, ta có ảnh gốc không bị xuống cấp để xử lý trước khi nó bị nhiễu làm xuống cấp, - ví dụ như nhiễu lượng tử. Trong những ứng dụng như vậy, ảnh chưa bị xuống cấp có thể đi qua bộ lọc thông cao trước khi nó xuống cấp, rồi sau khi xuống cấp lại cho qua bộ lọc thông thấp. Kết quả là ảnh cải thiện được độ dẽ hiểu. Chẳng hạn, khi xuống cấp do nhiễu ngẫu nhiên băng rộng, trong ảnh bị xuống cấp SNR (tỷ số tín trên tạp) hiệu dụng ở các thành phần tần số cao thấp hơn ở các thành phần tần số thấp, nhờ đặc tính thông thấp của ảnh. Cho ảnh qua bộ lọc thông cao trước khi xuống cấp thường cải thiện được SNR ở các thành phần tần số cao, mặc dầu phải chịu hy sinh chút ít ở các thành phần tần số thấp.

Hình 2.14 cho những ví dụ minh họa đáp ứng xung của bộ lọc thông thấp thường dùng cho cải thiện ảnh. Để minh họa cho tính năng bộ lọc thông thấp dùng cho cải thiện ảnh, đưa ra hai ví dụ. Hình 2.15(a) biểu diễn ảnh gốc  $256 \times 256$  pixel không có nhiễu (noise-free) và Hình 2.15(b) biểu diễn ảnh đã bị xuống cấp bởi nhiễu ngẫu nhiên Gauss băng rộng với SNR bằng 15 dB. SNR được định nghĩa bằng  $10\log_{10}$ (phương sai ảnh/phương sai nhiễu). Hình 2.15(c) biểu diễn kết quả lọc thông thấp ảnh bị xuống cấp. Bộ lọc thông thấp được sử dụng biểu diễn trên Hình 2.14(c). Hình 2.15 cho thấy rõ ràng là sự lọc thông thấp làm giảm nhiễu cộng, nhưng đồng thời cũng làm mờ ảnh.

Mờ là tác nhân chủ yếu hạn chế sự lọc thông thấp. Hình 2.16(a) biểu diễn ảnh gốc  $512 \times 512$  pixel với 8 bit/pixel. Hình 2.16(b) biểu diễn ảnh được mã hoá bởi hệ PCM với kỹ thuật nhiễu giả Robert 2 bít/pixel. Kỹ thuật nhiễu giả Robert được thảo luận trong chương 4. Hình 2.16(c) cho kết quả sự lọc thông cao trước khi mã hoá và sự lọc thông thấp sau khi mã hoá. Bộ lọc thông cao và bộ lọc thông thấp sử dụng trong những ví dụ này theo thứ tự là các bộ lọc ở Hình 2.8(c) và 1.14(c)

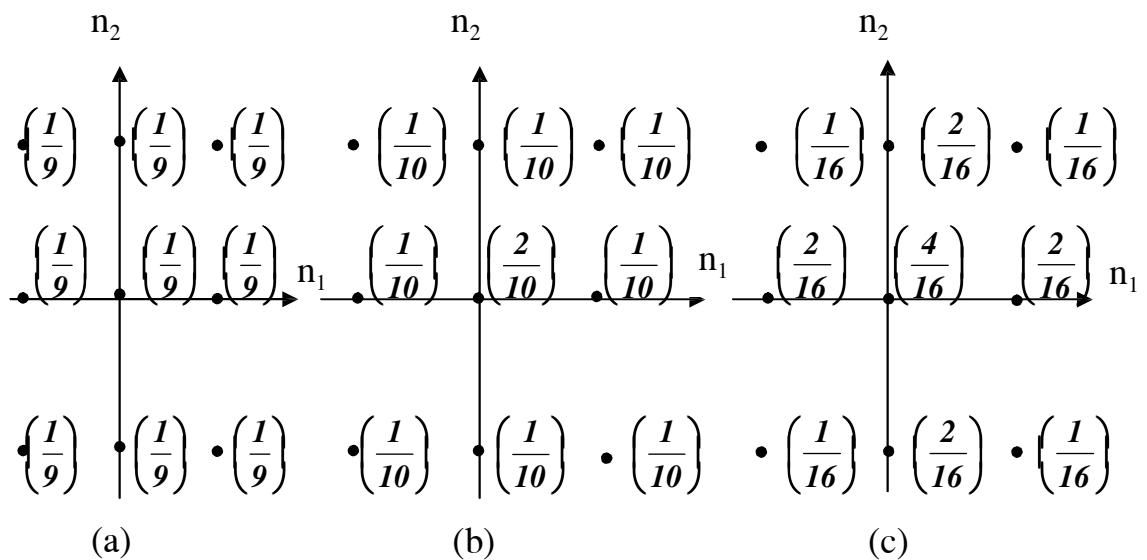
## 2.2. PHÉP LỌC TRUNG VỊ (LỌC MEDIAN)

Lọc trung vị là một quá trình phi tuyến có ích trong việc làm giảm nhiễu xung hoặc nhiễu muối-tiêu. Nó cũng có ích trong việc làm giảm nhiễu ngẫu nhiên mà bảo vệ các đường biên ảnh. Nhiễu xung hoặc nhiễu muối-tiêu xuất hiện do bít lỗi ngẫu nhiên trong kênh truyền thông. Ở bộ lọc trung vị có một cửa sổ trượt dọc theo ảnh và giá trị trung vị cường độ của các pixel bên trong cửa sổ trở thành cường độ đầu ra của pixel được xử lý. Chẳng hạn, giả sử các giá trị của các pixel trong một cửa sổ là 5, 6, 55, 10 và 15, còn pixel được xử lý có giá trị 55. Đầu ra của bộ lọc trung vị ở pixel đang xét là 10, tức là median của 5 giá trị trên.

$$\cdot \frac{1}{9} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$\cdot \frac{1}{10} \cdot \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline -2 & 5 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array}$$

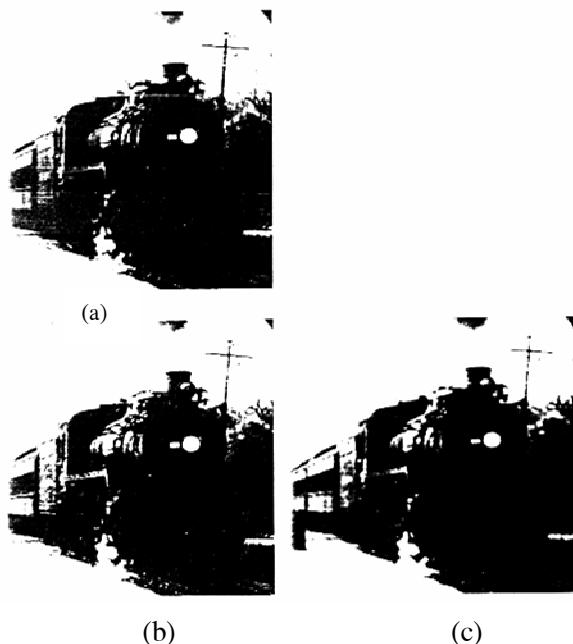
$$\cdot \frac{1}{16} \cdot \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline -2 & 19 & -2 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$



**Hình 2.14:** Các đáp ứng xung của bộ lọc thông thấp dùng cho cải thiện ảnh.

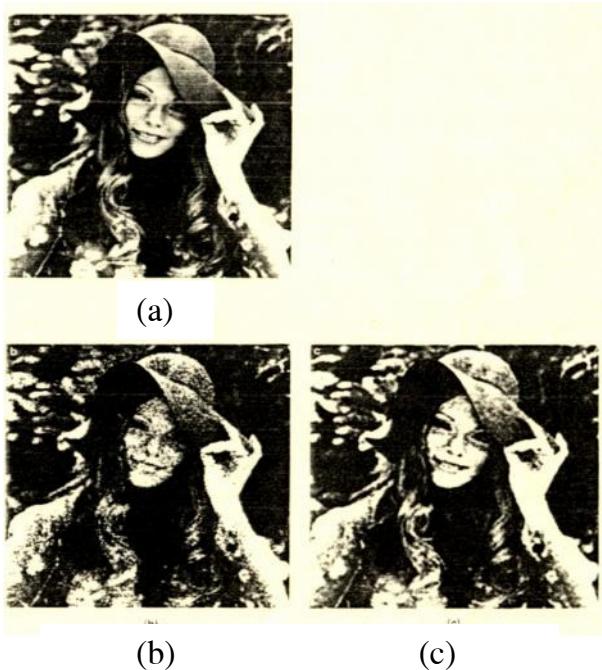
**Hình 2.15:** Ví dụ về làm giảm nhiễu bằng bộ lọc thông thấp.

- (a) Ảnh gốc 256 x 256 pixel;
- (b) Ảnh gốc bị xuống cấp bởi nhiễu ngẫu nhiên Gauss băng rộng ở mức SNR = 15 dB;
- (c) Kết quả xử lý ảnh trong hình (b) bằng bộ lọc thông thấp.

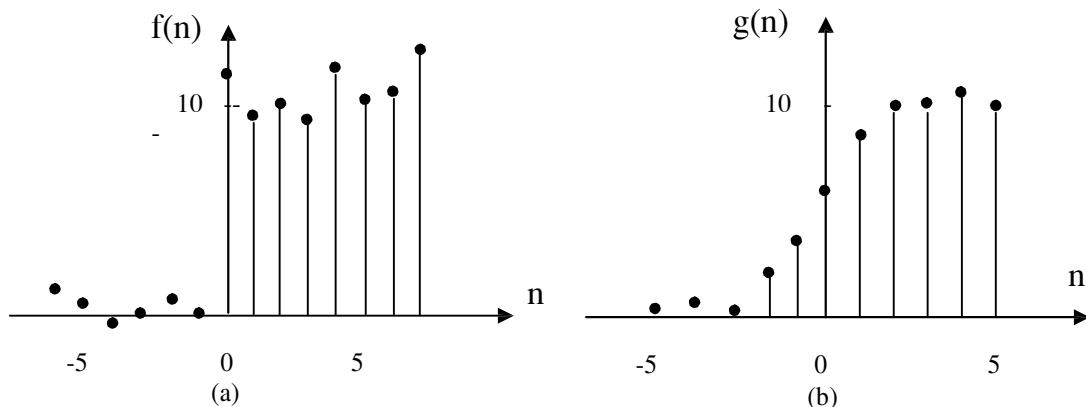


**Hình 2.16:** Ứng dụng của bộ lọc thông thấp trong việc mã hoá ảnh.

- (a) Ảnh gốc 256 x 256 pixel;
- (b) Ảnh trong hình (a) sau khi mã hoá bằng một hệ PCM với kỹ thuật nhiễu giả Robert 2 bít/pixel.
- (c) Ảnh trong hình (a) sau khi qua quy trình xử lý “lọc thông cao – mã hoá – lọc thông thấp”.



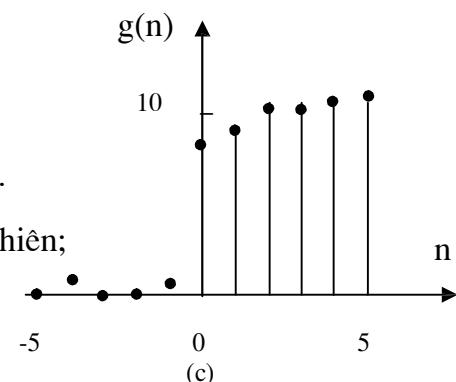
Tương tự bộ lọc thông thấp, bộ lọc trung vị làm tròn ảnh và nhờ đó có lợi cho việc làm giảm nhiễu. Khác với bộ lọc thông thấp, bộ lọc trung vị có thể bảo tồn những mất liên tục theo hàm bậc thang và có thể làm tròn một vài pixel có giá trị chênh lệch với những pixel lân cận khá xa mà không ảnh hưởng đến các pixel khác. Hình 2.17(a) là dãy 1-D bậc thang bị xuống cấp bởi một lượng nhỏ nhiễu ngẫu nhiên. Hình 2.17(b) là kết quả sau khi đi qua bộ lọc thông thấp mà đáp ứng xung là một cửa sổ chữ nhật 5 - điểm. Hình 2.17(c) là kết quả sau khi đi qua bộ lọc trung vị 5 -điểm. Từ hình này thấy rõ là bộ lọc trung vị bảo tồn mất liên tục kiểu bậc thang tốt hơn. Hình 2.18(a) là dãy một chiều (1-D) với hai giá trị chênh lệch khá xa với các điểm xung quanh. Hình 2.18(b) và (c) theo thứ tự là kết quả của bộ lọc thông thấp và bộ lọc trung vị. Các bộ lọc sử dụng ở Hình 2.18 cũng là những bộ lọc đã sử dụng ở Hình 2.17. Nếu hai giá trị xung là do nhiễu thì dùng bộ lọc trung vị sẽ làm giảm nhiễu. Nhưng nếu hai giá trị đó lại là bộ phận của tín hiệu thì sử dụng bộ lọc trung vị sẽ làm méo tín hiệu.

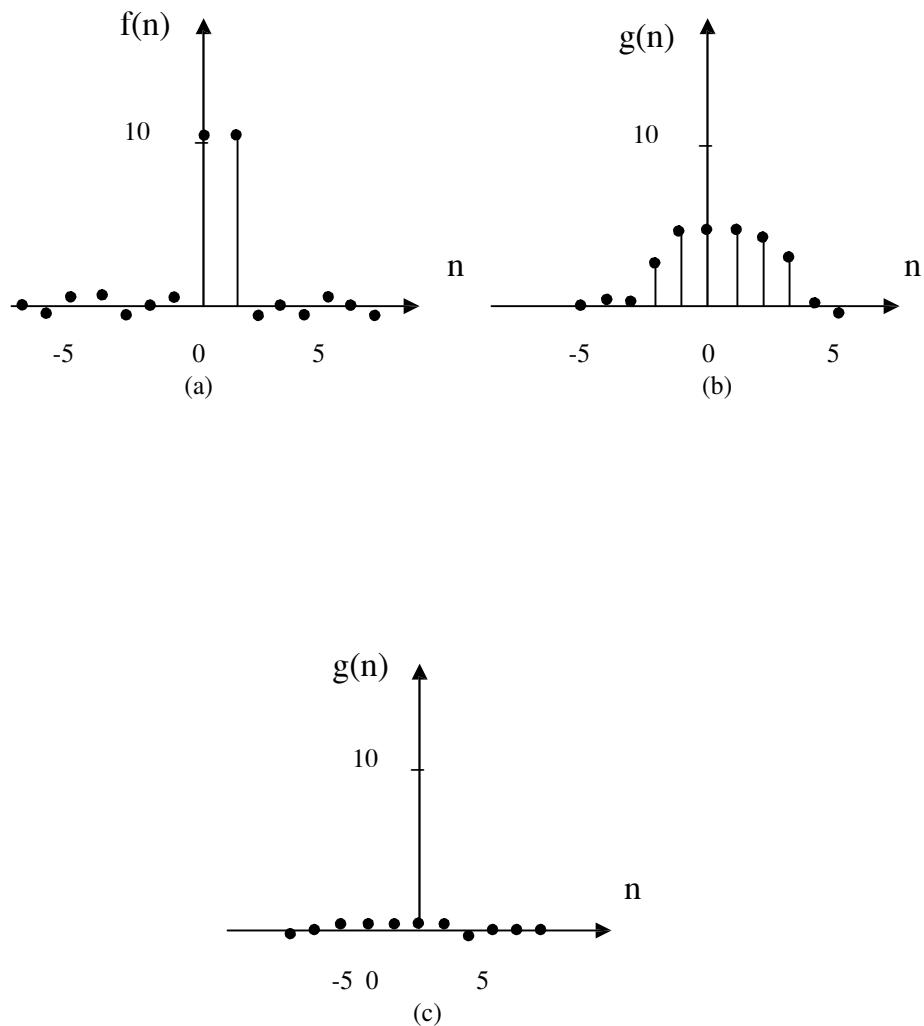


**Hình 2.17:** Minh họa về xu hướng của bộ lọc

trung vị bảo tồn những bất liên tục theo hàm bậc thang.

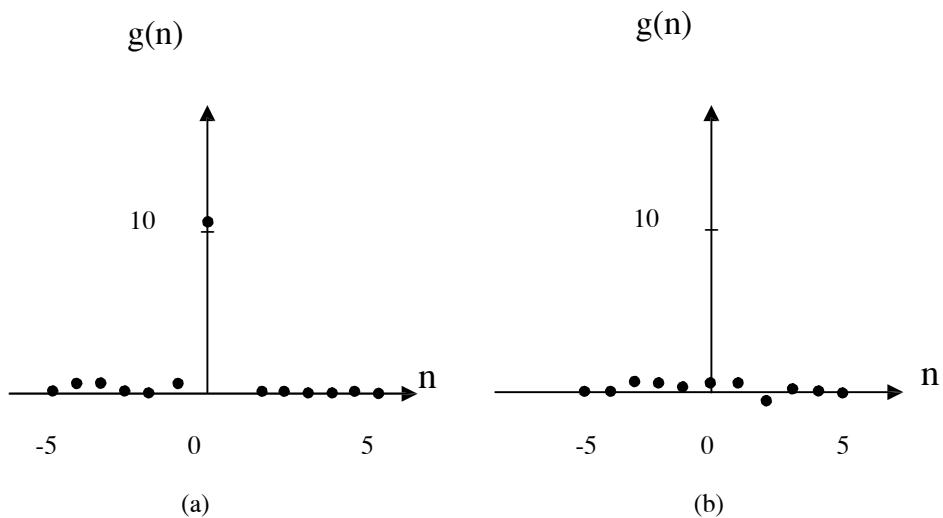
- Dãy bậc thang 1-D bị xuống cấp bởi nhiễu ngẫu nhiên;
- Dãy trong hình (a) sau khi qua bộ lọc thông thấp có đáp ứng xung là một cửa sổ chữ nhật 5 -điểm;
- Dãy trong hình (a) sau khi qua bộ lọc trung vị 5 -điểm.





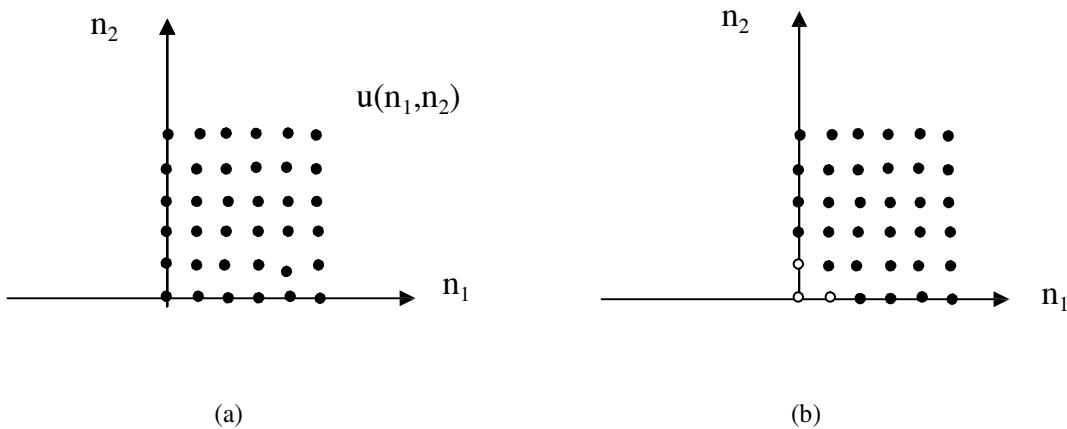
**Hình 2.18:** Minh họa khả năng loại bỏ các giá trị xung của bộ lọc trung vị.

- Dãy 1-D với hai mẫu liên tiếp chênh lệch khá xa với các mẫu xung quanh;
- Dãy trong hình (a) sau khi qua bộ lọc thôn g thấp có đáp ứng xung là hình chữ nhật 5 điểm;
- Dãy trong hình (a) sau khi qua bộ lọc trung vị 5 -điểm.



**Hình 2.19:** Kết quả áp dụng bộ lọc trung vị cho dãy trên Hình 2.18(a) theo các kích thước cửa sổ khác nhau. Kết quả này chứng minh rằng khả năng loại bỏ các giá trị xung của bộ lọc trung vị phụ thuộc vào kích thước cửa sổ.

- Kích thước cửa sổ = 3;
- Kích thước cửa sổ = 5;
- Kích thước cửa sổ = 7.



**Hình 2.20:** Minh họa việc bộ lọc trung vị 2 -D,  $N \times N$  điểm làm méo bất liên tục 2 -D theo hàm bậc thang.

- (a) Dãy bậc thang đơn vị  $u(n_1, n_2)$ ;
- (b) Kết quả lọc  $u(n_1, n_2)$  bằng bộ lọc trung vị  $5 \times 5$  điểm.

Một thông số quan trọng khi sử dụng bộ lọc trung vị là kích thước cửa sổ. Hình 1.19 biểu diễn kết quả lọc trung vị của tín hiệu trên Hình 2.18(a) theo hàm kích thước cửa sổ. Nếu cỡ cửa sổ nhỏ hơn 5 thì hai pixel ứng với hai giá trị xung hầm như không bị ảnh hưởng. Đối với cửa sổ lớn hơn thì bị ảnh hưởng đáng kể. Do đó việc chọn cỡ cửa sổ phụ thuộc vào bối cảnh. Bởi vì khó có thể chọn trước cỡ cửa sổ tối ưu, nên phải thử dùng nhiều bộ lọc trung vị có cỡ cửa sổ khác nhau và chọn lấy kết quả tốt nhất trong những ảnh nhận được.

Trong tiết trên ta thảo luận phép lọc trung vị 1 -D. Có thể mở rộng trực tiếp tính toán bộ lọc trung vị từ trường hợp 1 -D sang trường hợp 2 -D. Tuy nhiên, không phải tất cả các đặc tính của bộ lọc trung vị 1 -D đều áp dụng được cho bộ lọc trung vị 2 -D. Chẳng hạn, phép lọc trung vị 1 -D dãy bậc thang đơn vị  $u(n)$  bảo tồn được sự mất liên tục bậc thang và không phương hại gì tín hiệu  $u(n)$ . Nay giờ giả sử ta lọc trung vị 2 -D dãy bậc thang  $u(n_1, n_2)$  bằng bộ lọc trung vị hai chiều  $N \times N$  điểm. Hình 2.20(a) là  $u(n_1, n_2)$  và Hình 2.20(b) là kết quả lọc  $u(n_1, n_2)$  với bộ lọc trung vị hai chiều  $5 \times 5$  điểm. Từ Hình 2.20(b), thấy rằng sự mất liên tục về cường độ, được coi như là bậc thang 1 -D

(với  $n_1$  lớn, tại điểm  $n_2=0$ , và với  $n_2$  lớn, tại điểm  $n_1=0$ ) không bị ảnh hưởng. Tuy nhiên những mất liên tục thực sự là hàm bậc thang 2 -D ( $n_1=n_2=0$ ) bị méo nghiêm trọng. Một phương pháp có xu hướng bảo tồn mất liên tục dạng bậc thang 2 -D là đem lọc tín hiệu 2-D theo phương nằm ngang với bộ lọc trung vị 1 -D và sau đó lại cho kết quả qua bộ lọc theo phương thẳng đứng với bộ lọc trung vị 1 -D khác. Phương pháp này được gọi là phép lọc trung vị tách được, dùng trong những ứng dụng có bộ lọc trung vị 2 -D. Khi cho bộ lọc trung vị tách riêng tác dụng vào  $u(n_1,n_2)$  thì tín hiệu  $u(n_1,n_2)$  không bị ảnh hưởng.

**Hình 2.21:** Ví dụ về làm giảm nhiễu ngẫu

nhiên bằng rộng băng bộ lọc trung vị. (a)

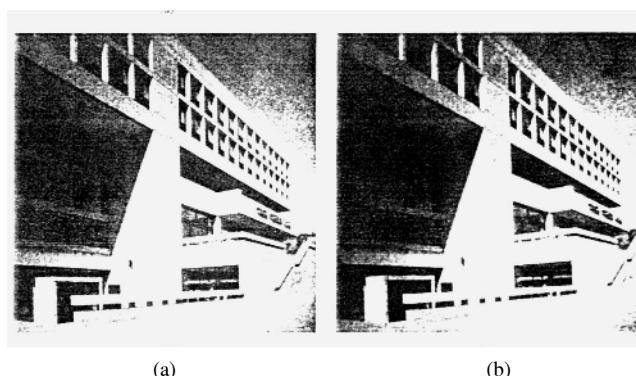
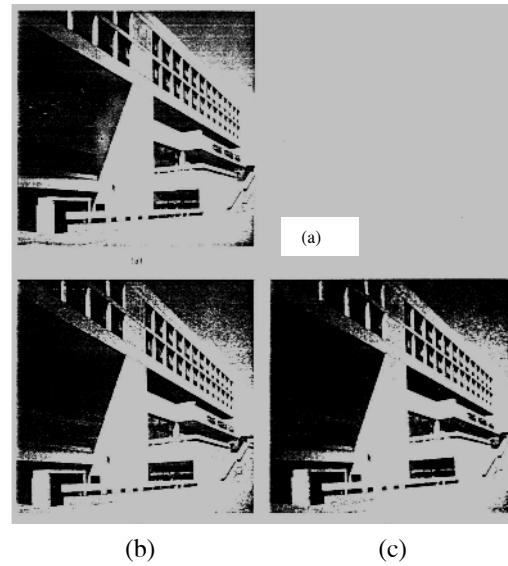
ảnh gốc 512 x 512 pixel;

(b) Ảnh bị xuống cấp bởi nhiễu ngẫu nhiên

Gauss băng rộng ở mức SNR bằng 7dB;

(c) Ảnh được xử lý bởi phép lọc trung vị tách riêng, cỡ cửa sổ cho cả hai bộ lọc trung vị 1 -D (theo phương ngang và

phương dọc) đều bằng 3.



**Hình 2.22:** Ví dụ về làm giảm nhiễu muối-tiêu bằng phép lọc trung vị.

(a) Ảnh trong Hình 2.21(a) bị xuống cấp vì nhiễu muối-tiêu

(b) Ảnh đã xử lý bằng bộ lọc trung vị tách riêng đã dùng ở Hình 2.21.

Bộ lọc trung vị là một hệ phi tuyến, cho nên nhiều kết quả lý thuyết về hệ tuyến tính không áp dụng vào đây được. Chẳng hạn, kết quả của phép lọc trung vị tách riêng phụ thuộc vào thứ tự các bộ lọc trung vị 1-D theo phương ngang và theo phương dọc. Tuy có khó khăn này, người ta cũng đã phát triển được một số kết quả lý thuyết về lọc trung vị [Gallagher and Wise; Nodes and Gallagher; Arce and McLoughlin]. Một kết quả nổi bật rằng: áp dụng lọc trung vị 1-D lặp lại nhiều lần cho dãy 1-D thì cuối cùng sẽ nhận được tín hiệu gốc (root signal), tín hiệu này bất biến với tất cả mọi lần lọc về sau.

Có hai ví dụ sau đây minh họa cho tính năng bộ lọc trung vị. Trong ví dụ đầu, ảnh gốc  $512 \times 512$  pixel là Hình 2.21(a) bị xuống cấp bởi nhiễu ngẫu nhiên Gauss bằng rộng ở mức SNR bằng 7dB. Ảnh bị xuống cấp biểu diễn trên Hình 2.21(b). Hình 2.21(c) là ảnh được xử lý bởi phép lọc trung vị tách riêng, cỡ cửa sổ cho cả hai bộ lọc trung vị 1-D (phương ngang và phương dọc) đều bằng 3. Mặc dù những đường biên ảnh rất sắc không bị mờ, phép lọc trung vị vẫn gây mờ toàn ảnh đáng kể. Trong ví dụ thứ hai, ảnh gốc lấy từ Hình 2.21(a) bị xuống cấp bởi nhiễu muối-tiêu. Hình 2.22(a) là ảnh đã xuống cấp. Hình 2.22(b) là ảnh được xử lý bằng bộ lọc trung vị tách riêng đã dùng ở Hình 2.21. Ví dụ này cho thấy rằng phép lọc trung vị rất hiệu quả trong việc khử nhiễu muối-tiêu.

### 2.3. LÀM TRƠN PIXEL NGOẠI CỠ

Giống như bộ lọc trung vị, làm trơn pixel ngoại cỡ là thuật toán phi tuyến và có ích trong việc làm giảm nhiễu muối-tiêu. Trong phương pháp này, cửa sổ dọc trượt theo ảnh, và ta nhận được giá trị trung bình của các pixel, ngoại trừ pixel đang xét. Nếu sự chênh lệch giữa giá trị trung bình và giá trị của pixel đang xét vượt quá một mức ngưỡng nào đó thì pixel đang xét được thay thế bằng giá trị trung bình. Nếu không thì cho qua. Vì rất khó xác định trước các giá trị thông số phù hợp, nên để xử lý ảnh phải dùng nhiều giá trị ngưỡng cũng như kích thước cửa sổ khác nhau và cuối cùng chọn kết quả tốt nhất.

Hình 2.23 minh họa hiệu quả của phép làm trơn pixel ngoại cỡ. Hình 2.23 là kết quả nhận được sau khi xử lý ảnh trong Hình 2.22(a) bằng phép làm trơn pixel ngoại cỡ với giá trị ngưỡng là 50 và cửa sổ  $3 \times 3$  điểm.



**Hình 2.23:** Ví dụ về làm giảm nhiễu muối-tiêu bằng phép làm tròn pixel ngoại cõi. Ảnh trên Hình 2.22(a) được xử lý bằng phép làm tròn pixel ngoại cõi với giá trị ngưỡng là 50 và cửa sổ  $3 \times 3$  điểm.

### 3. PHÁT HIỆN BIÊN

Biên trong ảnh là một đường biên giới hay một đường bao (contour) mà ở đó xuất hiện sự thay đổi đáng kể một vài tính chất vật lý của ảnh, như độ phản xạ bề mặt, độ chiếu sáng hoặc khoảng cách từ những bề mặt nhìn thấy được đến vị trí người xem. Thay đổi tính chất vật lý biểu hiện bằng nhiều cách, bao gồm thay đổi cường độ, màu, kết cấu bề mặt. Ở đây, ta chỉ xét về sự thay đổi cường độ ảnh.

Phát hiện đường biên rất có lợi trong một số bối cảnh, chẳng hạn trong nhiệm vụ lý giải ảnh như nhận dạng đối tượng, một bước quan trọng là phân đoạn ảnh thành từng vùng khác nhau, ứng với những đối tượng khác nhau trong cảnh. Phát hiện đường biên thường là bước đầu tiên trong phân đoạn ảnh. Một ví dụ khác, là phương pháp mã hoá ảnh với nhịp bít thấp bằng cách chỉ mã hoá những đường biên phát hiện ra. Ta biết rằng, khi một ảnh chỉ gồm toàn đường biên thì độ đẽ hiểu rất cao.

Ý nghĩa của sự thay đổi tính chất vật lý trong ảnh phụ thuộc vào ứng dụng: điểm thay đổi cường độ được coi là điểm biên ở ứng dụng này, nhưng trong ứng dụng khác không phải là điểm biên. Trong hệ nhận dạng đối tượng, đường bao toàn bộ đối tượng đã đủ cho việc nhận dạng, còn những đường bao khác đại diện những chi tiết trên đối tượng cũng có thể không coi là đường biên. Như vậy, không thể định nghĩa đường biên ngoài bối cảnh của ứng dụng. Tuy nhiên những algorit phát hiện đường biên có ích trong nhiều tập ứng dụng lớn đã được phát triển. Trong tiết này ta thảo luận về một số algorit phát hiện đường biên tiêu biểu.

### 3.1. PHƯƠNG PHÁP GRADIENT

Xét một hàm tương tự  $f(x)$  biểu diễn đường biên 1-D như trên Hình 2.24(a). Trong những bài toán điển hình, coi giá trị  $x_0$  trong hình là một điểm biên. Một cách để xác định  $x_0$  là tính đạo hàm bậc nhất  $f'(x)$  hoặc đạo hàm bậc hai  $f''(x)$ . Các Hình 2.24(b) và 1.24(c) biểu diễn  $f'(x)$  và  $f''(x)$ . Từ hình vẽ có thể xác định giá trị  $x_0$  bằng cách tìm cực trị cục bộ (min hoặc max) của  $f'(x)$  hoặc bằng cách tìm điểm  $f''(x)$  đi qua giá trị không, ở đó  $f''(x)$  đổi dấu. Trong tiết này ta chỉ thảo luận về những phương pháp khai thác các đặc tính của  $f'(x)$ . Trong tiết sau, thảo luận về những phương pháp khai thác các đặc tính của  $f''(x)$ .

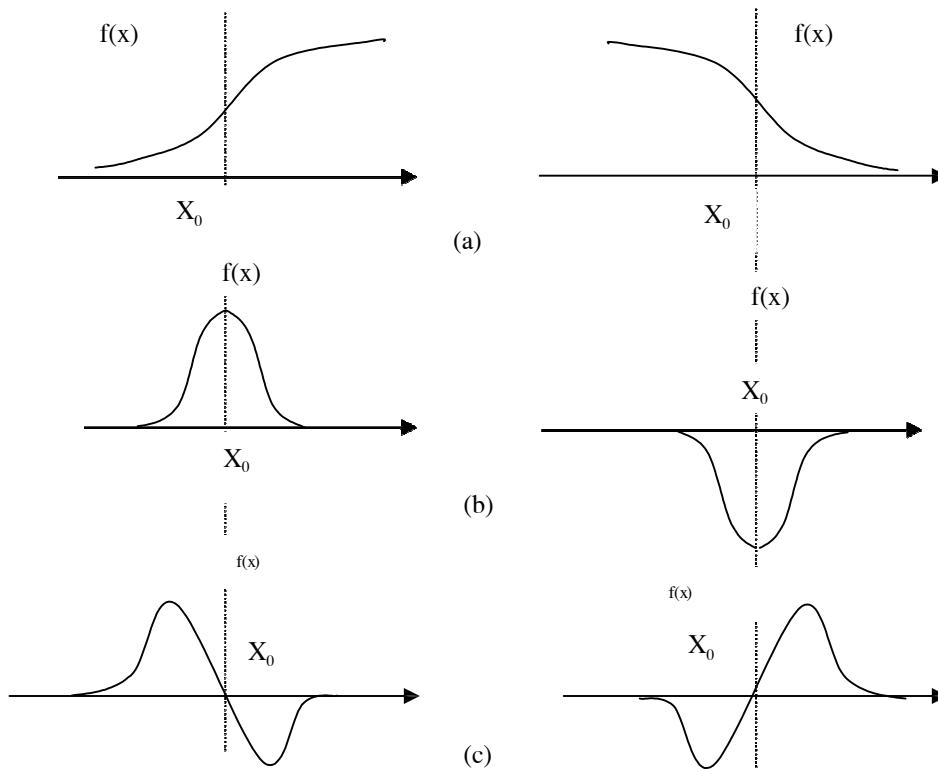
Ngoài việc xác định điểm biên  $x_0$ ,  $f'(x)$  cũng có thể sử dụng để ước lượng cường độ và hướng của đường biên. Nếu  $|f'(x)|$  là lớn thì  $f(x)$  biến thiên rất nhanh đồng thời cường độ cũng thay đổi nhanh. Nếu  $|f'(x)|$  dương thì  $f(x)$  tăng. Dựa trên những nhận xét trên suy ra một cách tiếp cận để phát hiện đường biên là sử dụng hệ ở Hình 2.25. Trong hệ này trước hết từ  $f(x)$  tính  $|f'(x)|$ . Nếu  $|f'(x)|$  lớn hơn một ngưỡng nào đó thì nó có thể là một “ứng viên” điểm biên. Nếu tất cả các giá trị của  $x$  sao cho  $|f'(x)|$  lớn hơn một ngưỡng nào đó được phát hiện là điểm biên thì biên sẽ xuất hiện dưới dạng một đường chéo không phải một điểm. Để tránh vấn đề này ta yêu cầu thêm là  $|f'(x)|$  có giá trị cực đại cục bộ ở những điểm biên. Cũng cần xác định xem  $f(x)$  là tăng hoặc giảm tại  $x=x_0$ . Thông tin cần thiết chưa đựng trong  $f'(x)$  tại  $x=x_0$ . Việc chọn ngưỡng phụ thuộc vào ứng dụng. Khi ngưỡng tăng chỉ những giá trị của  $x$  ở đây  $f(x)$  tăng nhanh mới được ghi lại làm “ứng viên” điểm biên. Việc chọn ngưỡng tối ưu không phải dễ, cho nên phải thử mò mẫm một số lần. Cũng có thể chọn ngưỡng theo phương pháp thích nghi. Hệ trên Hình 2.25 dựa vào loại biên đặc biệt cho trong Hình 2.24(a), nhưng vẫn có thể ứng dụng để phát hiện các loại biên khác.

Dạng suy rộng  $f'(x)$  vào trường hợp hàm hai chiều  $f(x,y)$  là gradient

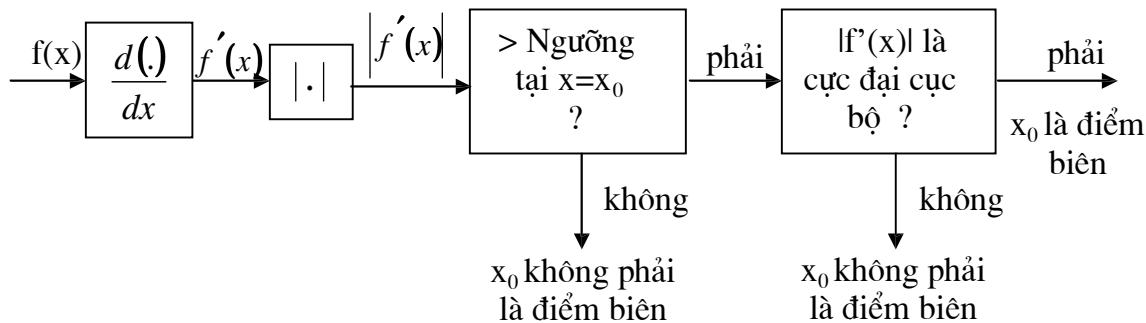
$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \hat{i}_x + \frac{\partial f(x, y)}{\partial y} \hat{i}_y \quad (2.6)$$

trong đó  $\hat{i}_x$  là vector đơn vị theo hướng  $x$ ,  $\hat{i}_y$  là vector đơn vị theo hướng  $y$ . Việc dựa vào  $|\nabla f(x,y)|$  để suy rộng hệ phát hiện đường biên trên Hình 2.25 được biểu diễn trên Hình 2.26. Thoạt tiên tính biên độ của  $|\nabla f(x,y)|$  sau đó đem so sánh với một ngưỡng để

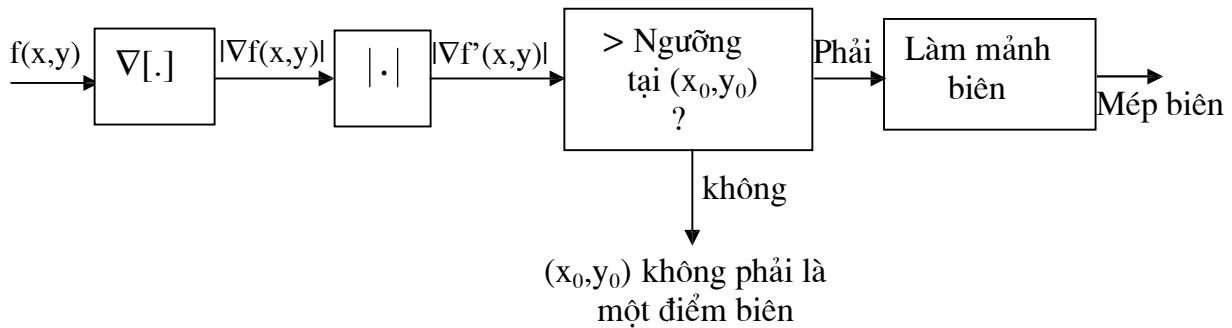
xác định các “ứng viên” điểm biên. Nếu tất cả giá trị  $(x,y)$  sao cho  $|\nabla f(x,y)|$  lớn hơn một ngưỡng nào đó đều được phát hiện là những điểm biên thì biên sẽ xuất hiện dưới dạng dải chứ không chỉ là đường. Quá trình xác định ra một đường biên từ một dải các “ứng viên” điểm biên được gọi là làm mảnh dải biên. Trong một algorit làm mảnh biên đơn giản, điểm biên được chọn bằng cách kiểm tra xem có phải  $|\nabla f(x,y)|$  là giá trị cực đại cục bộ, - ít ra cũng là cực đại trên một hướng. Thuộc tính  $|\nabla f(x,y)|$  đạt được giá trị cực đại cục bộ ít nhất là trên một hướng thường được kiểm tra theo một vài hướng xác định. Trong phần lớn trường hợp chỉ cần kiểm tra theo hướng nằm ngang và hướng thẳng đứng. Nếu  $|\nabla f(x,y)|$  là một giá trị cực đại cục bộ theo bất kỳ một hướng xác định nào tại điểm có khả năng là điểm biên, thì điểm này được coi là điểm biên. Một khó khăn với algorit làm mảnh dải biên đơn giản này là nó tạo một số đường biên giả nhỏ trong vùng lân cận đường biên mạnh. Một phương pháp đơn giản để khử hầu hết những đường biên giả nhỏ này là áp đặt thêm những điều kiện ràng buộc sau đây:



**Hình 2.24:** (a)  $f(x)$ ; (b)  $f'(x)$ ; (c)  $f''(x)$  cho một loại biên 1 -D điển hình.



**Hình 2.25:** Hệ phát hiện biên 1-D.



**Hình 2.26:** Hệ phát hiện biên 2-D.

(a) nếu  $|\nabla f(x,y)|$  có một giá trị cực đại cục bộ ở điểm  $(x_0,y_0)$  theo phương nằm ngang chứ không phải theo phương thẳng đứng, thì điểm  $(x_0,y_0)$  là điểm biên khi

$$\left| \frac{\partial f(x,y)}{\partial x} \right|_{x=x_0, y=y_0} > k \left| \frac{\partial f(x,y)}{\partial y} \right|_{x=x_0, y=y_0} \quad \text{với } k \text{ thường chọn cỡ bằng 2}$$

(b) nếu  $|\nabla f(x,y)|$  có giá trị cực đại cục bộ ở điểm  $(x_0,y_0)$  theo phương thẳng chéo chứ không phải theo phương nằm ngang, thì điểm  $(x_0,y_0)$  là điểm biên khi

$$\left| \frac{\partial f(x,y)}{\partial y} \right|_{x=x_0, y=y_0} > k \left| \frac{\partial f(x,y)}{\partial x} \right|_{x=x_0, y=y_0} \quad \text{với } k \text{ thường chọn cỡ bằng 2}$$

Khi  $|\nabla f(x,y)|$  có giá trị cực đại cục bộ tại điểm  $(x_0, y_0)$  theo phương nằm ngang, chứ không phải theo phương thẳng đứng, điều kiện (a) yêu cầu tốc độ biến thiên của cường độ theo phương nằm ngang phải lớn hơn theo phương thẳng đứng nhiều. Điều kiện (b) cũng như điều kiện (a) chỉ cần hoán vị  $x$  với  $y$ .

Hệ phát hiện đường biên dựa trên hàm  $|\nabla f(x,y)|$  gọi là bộ dò biên vô hướng bởi vì những hàm như vậy không định thiên theo một hướng đặc biệt nào. Nếu hệ dò biên dựa trên hàm có định thiên theo một hướng đặc biệt thì đó là một bộ phát hiện có hướng.

Nếu ta sử dụng  $\partial f(x,y)/\partial x$  thay cho  $|\nabla f(x,y)|$ , chẳng hạn trong hệ Hình 2.26, hệ sẽ dò biên theo phương thẳng đứng, mà không có đáp ứng với những biên trên phương nằm ngang.

Đối với một dãy hai chiều  $f(n_1, n_2)$ , đạo hàm riêng  $\partial f(x,y)/\partial x$  và  $\partial f(x,y)/\partial y$  có thể được thay thế bởi một hiệu, chẳng hạn  $\partial f(x,y)/\partial x$  có thể được thay thế bởi

$$\frac{\partial f(x, y)}{\partial x} \leftrightarrow [f(n_1, n_2) - f(n_1-1, n_2)]/T, \quad (2.7a)$$

$$[f(n_1+1, n_2) - f(n_1, n_2)]/T, \quad (2.7b)$$

$$\text{hoặc} \quad [f(n_1+1, n_2) - f(n_1-1, n_2)]/(2T). \quad (2.7c)$$

Vì các đạo hàm tính ra được so sánh với một ngưỡng, và ngưỡng này có thể điều chỉnh nên có thể bỏ qua các hệ số tỷ lệ xích  $1/T$  và  $1/2T$ . Thường lấy giá trị trung bình các biểu thức (2.7) trên nhiều mẫu để tăng độ tin cậy và tính liên tục của giá trị ước lượng của  $\partial f(x,y)/\partial x$ . Những ví dụ về các giá trị ước lượng  $\partial f(x,y)/\partial x$  “đã cải thiện” là

$$\frac{\partial f(x, y)}{\partial x} \leftrightarrow [f(n_1+1, n_2+1) - f(n_1-1, n_2+1)] + [f(n_1+1, n_2) - f(n_1-1, n_2)] +$$

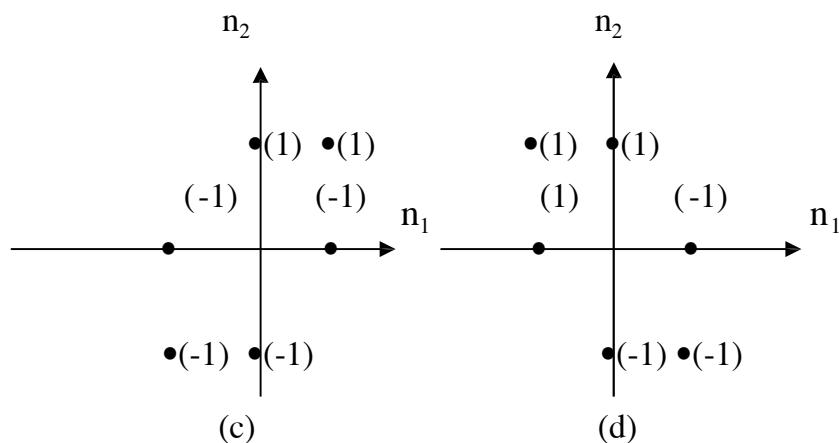
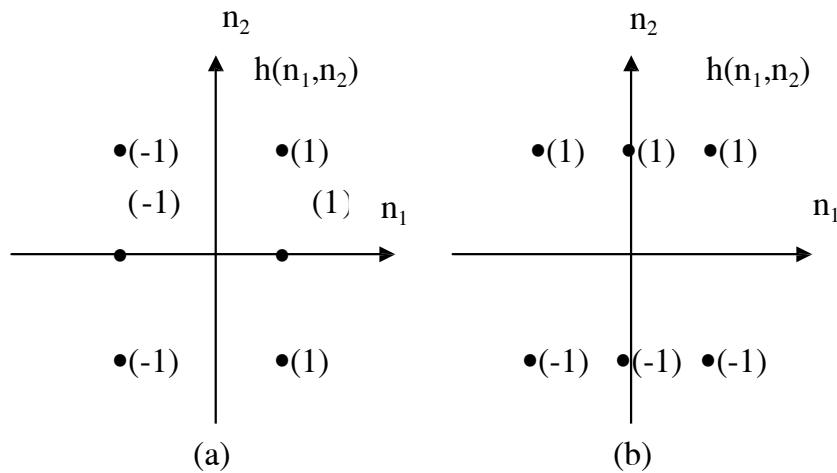
$$[f(n_1+1, n_2-1) - f(n_1-1, n_2-1)] \quad (2.8a)$$

$$\text{hoặc} \quad [f(n_1+1, n_2+1) - f(n_1-1, n_2+1)] + 2[f(n_1+1, n_2) - f(n_1-1, n_2)] + \\ [f(n_1+1, n_2-1) - f(n_1-1, n_2-1)] \quad (2.8b)$$

Trong (2.8) đã bỏ các hệ số tỷ lệ xích.

Thuật toán hiệu (differencing operation) trong (2.7) và (2.8) có thể được xem như là tích chập của  $f(n_1, n_2)$  với đáp ứng xung của bộ lọc  $h(n_1, n_2)$ . Những ví dụ đáp ứng xung có thể sử dụng để phát triển các bộ dò biên có hướng được trình bày Hình 2.27. Các bộ lọc  $h(n_1, n_2)$  ở Hình 2.27(a) và 1.27(b) phát hiện đường biên theo phương

thẳng đứng và phương nằm ngang theo thứ tự có thể xem như là các phép lấy xấp xỉ  $\partial f(x,y)/\partial x$  và  $\partial f(x,y)/\partial y$ . Các bộ lọc  $h(n_1, n_2)$  ở Hình 2.27(c) và 1.27(d) phát hiện đường biên theo hướng hai đường chéo. Gradient  $\nabla f(x,y)$  trong (2.6) cũng có thể biểu diễn dưới dạng các đạo hàm riêng bậc nhất trong hệ toạ độ quay. Khi quay một góc 45 độ thì hướng các đạo hàm riêng theo hướng hai đường chéo.



**Hình 2.27:** Đáp ứng xung của các bộ lọc có thể dùng cho phát hiện biên định hướng g.

- Phát hiện biên theo phương thẳng đứng;
- Phát hiện biên theo phương nằm ngang;
- và (d) phát hiện biên theo phương đường chéo.

Có thể triển khai các bộ phát hiện vô hướng bằng cách lấy xấp xỉ rời rạc (discrete approximation)  $|\nabla f(x,y)|$  trong hệ ở Hình 2.26. Từ (2.6)

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}(x,y)\right)^2 + \left(\frac{\partial f}{\partial y}(x,y)\right)^2} \quad (2.9)$$

Từ (2.9) có thể triển khai các bộ dò biên vô hướng bằng cách tổ hợp phi tuyến các số hạng dùng trong triển khai bộ dò biên có hướng. Một ví dụ về lấy xấp xỉ rời rạc của (2.9) có thể đem sử dụng cho bộ dò biên có hướng là

$$|\nabla f(x,y)| \leftrightarrow \sqrt{(f_x(n_1, n_2))^2 + (f_y(n_1, n_2))^2} \quad (2.10)$$

$$\text{trong đó } f_x(n_1, n_2) = f(n_1, n_2) * h_x(n_1, n_2)$$

$$f_y(n_1, n_2) = f(n_1, n_2) * h_y(n_1, n_2)$$

$h_x(n_1, n_2)$  và  $h_y(n_1, n_2)$  được biểu diễn trên Hình 2.28. Phương pháp được Sobel phát triển dựa trên (2.10) với  $h_x(n_1, n_2)$  và  $h_y(n_1, n_2)$  trong Hình 2.28. Một ví dụ khác là phương pháp do Roberts phát triển, cũng dựa trên (2.10) với  $h_x(n_1, n_2)$  và  $h_y(n_1, n_2)$  vẽ trên Hình 2.29. Tuỳ theo  $|\nabla f(x,y)|$  được tính xấp xỉ chính xác như thế nào trong miền rời rạc, có thể phát triển nhiều phương án khác nhau.

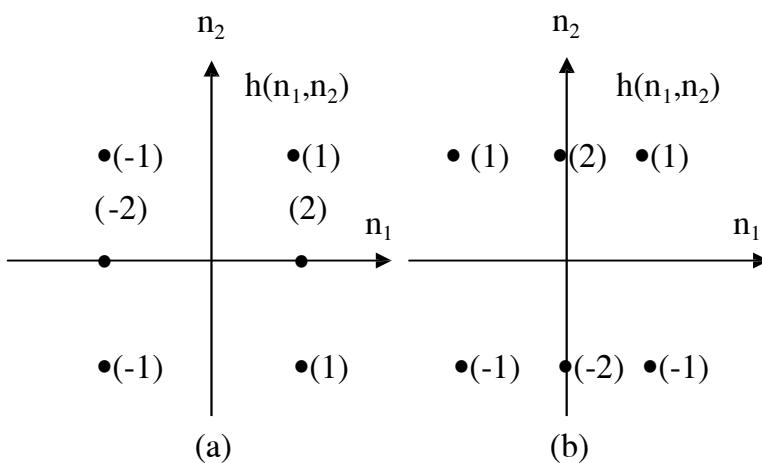
Hình 2.30 là kết quả dò biên khi sử dụng bộ dò biên có hướng. Hình 2.30(a) là ảnh gốc 512x512 pixel. Hình 2.30(b) và 1.30(c) theo thứ tự cho kết quả bộ dò theo phương thẳng đứng và phương nằm ngang. Các bộ dò theo phương thẳng đứng và phương nằm ngang dựa vào  $h(n_1, n_2)$  trong Hình 2.27(a) và 1.27(b). Hình 2.31(a) và 1.31(b) là kết quả sử dụng các bộ dò biên Sobel và Robert đối với ảnh trong Hình 2.30(a). Cả hai đều thuộc lớp các bộ dò vô hướng và các phương pháp xác định giá trị ngưỡng và cũng như kiểm tra tính cực đại cục bộ của biên đều là những phương pháp đã sử dụng ở Hình 2.30.

Có nhiều phương án khác nhau của những phương pháp phát hiện đường biên đã thảo luận trong tiết này. Ví dụ ta có thể sử dụng một tổ hợp phi tuyến khác của  $\partial f(x,y)/\partial x$  và  $\partial f(x,y)/\partial y$  thay cho

$$\sqrt{(\partial f(x,y)/\partial x)^2 + (\partial f(x,y)/\partial y)^2}$$

trong hệ ở Hình 2.26. Cũng có nhiều phương pháp khác nhau để làm mảnh dải biên.

Có thể cải thiện các phương pháp phát hiện đường biên đã nói đến trong tiết này bằng nhiều cách. Các phương pháp dựa vào tính toán một vài dạng của gradient hoặc lấy sai phân thường nhạy với nhiễu. Một số điểm biên lẻ loi xuất hiện ngẫu nhiên trên bản đồ biên trong Hình 2.31 hầu hết đều là kết quả của nhiễu nền hoặc chi tiết ảnh rất nhỏ. Nên sử dụng một vài phương pháp làm trơn nhiễu đã được thảo luận trong tiết 1.2 hoặc những phương pháp làm giảm nhiễu tinh xảo hơn sẽ nói đến ở chương 4 trước khi áp dụng algorit phát hiện đường biên. Cũng có thể khử các điểm biên lẻ loi xuất hiện ngẫu nhiên bằng cách xử lý đơn giản các bản đồ biên. Các phương pháp phát hiện đường biên dựa trên gradient có nhược điểm là có thể tạo nên một vài điểm mất liên tục trên đường biên như trên Hình 2.31.

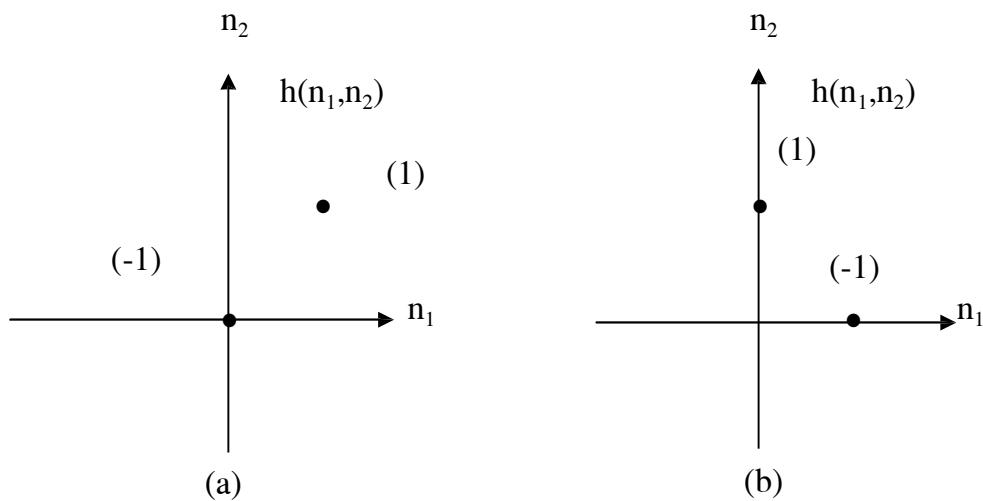


**Hình 2.28:** Phép lấy xấp xỉ

(a)  $\partial f(x, y)/\partial x$  với  $f(n_1, n_2) * h_x(n_1, n_2)$ ;

(b)  $\partial f(x, y)/\partial y$  với  $f(n_1, n_2) * h_y(n_1, n_2)$ ;

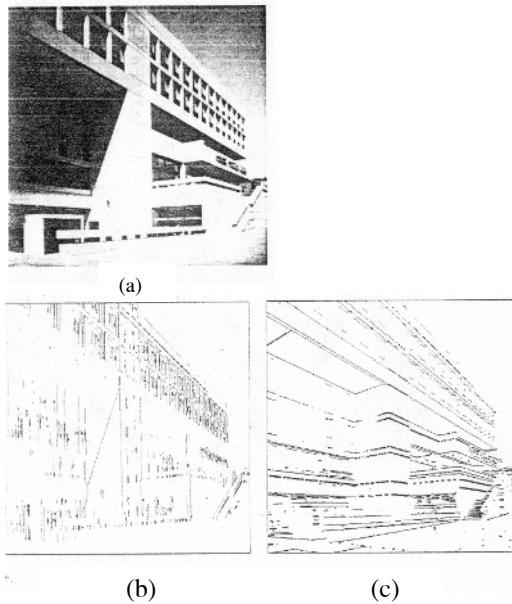
phương pháp phát hiện đường biên của Sobel dựa trên sự so sánh  $\sqrt{(f(n_1, n_2) * h_x(n_1, n_2))^2 + (f(n_1, n_2) * h_y(n_1, n_2))^2}$  với một ngưỡng.



**Hình 2.29:** Đáp ứng xung của các bộ lọc sử dụng phương pháp phát hiện đường biên Robert.

Phương pháp này dựa trên sự so sánh

$$\sqrt{(f(n_1, n_2) * h_x(n_1, n_2))^2 + (f(n_1, n_2) * h_y(n_1, n_2))^2} \text{ với một ngưỡng.}$$

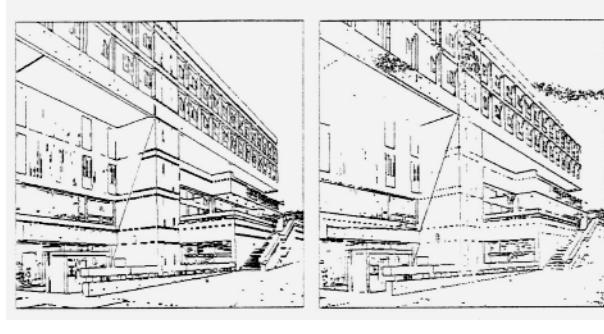


**Hình 2.30:** Các bản đồ đường biên nhận được bằng các bộ phát hiện có hướng.

- (a) Ảnh gốc 512 x 512 pixel;
- (b) Kết quả nhận được khi áp dụng bộ phát hiện theo phương thẳng đứng;
- (c) Kết quả nhận được khi áp dụng bộ phát hiện theo phương nằm ngang.

**Hình 2.31:** Kết quả nhận được khi:

- (a) Áp dụng bộ dò biên Sobel
- (b) Áp dụng bộ dò biên Robert cho ảnh trong Hình 2.30(a).



### 3.2. CÁC PHƯƠNG PHÁP DỰA TRÊN LAPLACIAN

Mục đích của algorit phát hiện đường biên là xác định ra những vùng ở đó có sự thay đổi đột ngột về cường độ. Trong trường hợp hàm một chiều  $f(x)$ , tìm những vùng có sự thay đổi đột ngột về cường độ tức là tìm những vùng ở đó  $f'(x)$  lớn. Đối với các phương pháp dựa trên gradient,  $f'(x)$  được coi là lớn khi biên độ  $|f'(x)|$  lớn hơn một ngưỡng. Một cách khác để kết luận  $f'(x)$  là lúc nó đạt một giá trị cực đại cục bộ, nghĩa là lúc đạo hàm bậc hai  $f''(x)$  qua điểm không. Điều này được minh họa trên Hình 2.24. Nếu tuyên bố điểm đi qua giá trị không là điểm biên, thì kết quả là sẽ có một số lượng lớn điểm được công nhận là điểm biên. Vì không có sự kiểm tra biên độ  $f'(x)$  cho nên bất kỳ một gợn sóng nhỏ nào của  $f(x)$  cũng đủ để tạo ra điểm biên. Do tính chất nhạy cảm với nhiễu này nên khi xử lý ảnh với nhiễu nền, cần cho qua một hệ giảm nhiễu trước khi đưa vào phát hiện biên.

Một dạng suy rộng của  $\partial^2 f(x,y)/\partial x^2$  dùng để dò biên hàm hai chiều  $f(x,y)$  là Laplacian  $\nabla^2 f(x,y)$ :

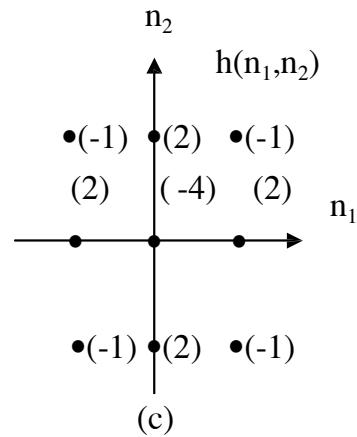
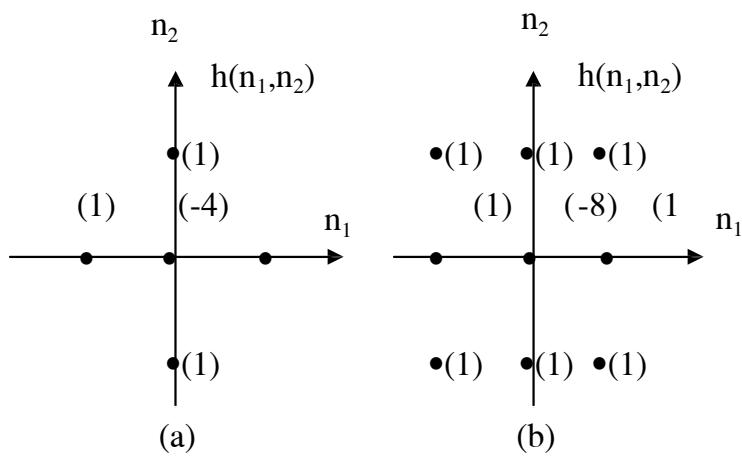
$$\nabla^2 f(x,y) = \nabla(\nabla f(x,y)) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (2.11)$$

đối với dãy hai chiều  $f(n_1, n_2)$ , các đạo hàm bậc hai  $\partial^2 f(x,y)/\partial x^2$  và  $\partial^2 f(x,y)/\partial y^2$  có thể thay thế bằng một dạng nào đó của sai phân bậc hai. Sai phân bậc hai có thể biểu diễn bằng tích chập của  $f(n_1, n_2)$  với đáp ứng xung của bộ lọc  $h(n_1, n_2)$ . Hình 2.32 cho những ví dụ về các  $h(n_1, n_2)$  có thể sử dụng được. Để minh chứng rằng  $f(n_1, n_2) * h(n_1, n_2)$  có thể xem như phép lấy xấp xỉ rời rạc  $\nabla^2 f(x,y)$ , ta xét  $h(n_1, n_2)$  Hình 2.32(a). Giả sử ta lấy xấp xỉ  $\partial f(x,y)/\partial x$  bằng

$$\frac{\partial f(x, y)}{\partial x} \rightarrow f_x(n_1, n_2) = f(n_1 + 1, n_2) - f(n_1, n_2) \quad (2.12)$$

Ta lại bỏ qua các hệ số tỷ lệ xích, bởi vì nó không ảnh hưởng tới các điểm đi qua giá trị không. Vì trong (2.12) sử dụng hiệu hướng thuận, nên khi lấy xấp xỉ  $\partial^2 f(x, y)/\partial x^2$  ta có thể sử dụng hiệu hướng ngược:

$$\frac{\partial^2 f(x, y)}{\partial x^2} \rightarrow f_{xx}(n_1, n_2) = f_x(n_1, n_2) - f_x(n_1 - 1, n_2) \quad (2.13)$$



**Hình 2.32:** Các ví dụ về  $h(n_1, n_2)$  có thể sử dụng trong việc lấy xấp xỉ  $\nabla^2 f(x, y)$  bằng  $f(n_1, n_2) * h(n_1, n_2)$ .

từ (2.12) và (2.13)

$$\frac{\partial^2 f(x, y)}{\partial x^2} \rightarrow f_{xx}(n_1, n_2) = f(n_1+1, n_2) - 2f(n_1, n_2) + f(n_1-1, n_2) \quad (2.14)$$

từ (2.11), (2.14) và lấy xấp xỉ  $\partial^2 f(x, y)/\partial y^2$  theo cách tương tự ta nhận được

$$\begin{aligned} \nabla^2 f(x, y) \rightarrow \nabla^2 f(n_1, n_2) &= f_{xx}(n_1, n_2) + f_{yy}(n_1, n_2) \\ &= f(n_1+1, n_2) + f(n_1-1, n_2) + f(n_1, n_2+1) + f(n_1, n_2-1) - 4f(n_1, n_2). \end{aligned} \quad (2.15)$$

Kết quả  $\nabla^2 f(n_1, n_2)$  là  $f(n_1, n_2)*h(n_1, n_2)$  với  $h(n_1, n_2)$  trong Hình 2.32(a). Tuỳ theo cách lấy xấp xỉ các đạo hàm bậc hai có thể nhận được nhiều đáp ứng xung  $h(n_1, n_2)$  khác nhau, bao gồm cả những cái trên Hình 2.32(b) và 1.32(c).

Hình 2.33 cho ví dụ về phát hiện đường biên bằng cách tìm điểm đi qua giá trị không của  $\nabla^2 f(n_1, n_2)$ . Hình 2.33(a) là ảnh gốc  $512 \times 512$  pixel. Hình 2.33(b) biểu diễn điểm đi qua giá trị không của  $\nabla^2 f(n_1, n_2)$ , nhận được từ công thức (2.15) và sử dụng ảnh trong Hình 2.33(a) là hàm hai chiều  $f(n_1, n_2)$ . Vì các đường bao(chu tuyến) đi qua giá trị không là những đường biên giới giữa các vùng, chúng có xu hướng là những đường liên tục.

Kết quả là việc làm mảnh dải biên, vốn là rất cần thiết trong phương pháp gradient, thì phương pháp Laplacian không cần đến. Ngoài ra, dùng phương pháp Laplacian thì các algorit bắt buộc đường biên phải liên tục sẽ không còn có ích nhiều như trong phương pháp gradient. Xem Hình 2.33(b) có thể thấy rõ điều đó, tuy nhiên việc chọn tất cả các điểm đi qua giá trị không làm điểm biên có xu thế tạo ra số lượng điểm biên quá lớn.

Phương pháp Laplacian được thảo luận trên tạo ra nhiều đường biên giả, thường xuất hiện ở những vùng ở đó phương sai cục bộ của ảnh nhỏ. Trường hợp một vùng có nền đồng nhất, nghĩa là ở đó  $f(n_1, n_2)$  là hằng số, coi như một trường hợp đặc biệt. Vì  $\nabla^2 f(n_1, n_2) = 0$  và ta phát hiện đường biên bằng các điểm  $\nabla^2 f(n_1, n_2)$  đi qua giá trị không, nên mọi biến thiên nhỏ của  $f(n_1, n_2)$  đều gây ra đường biên giả. Một phương pháp để khử đi số lớn đường biên giả là yêu cầu phương sai cục bộ phải đủ lớn ở điểm biên, như trên Hình 2.34. Phương sai cục bộ  $\sigma_f^2(n_1, n_2)$  có thể ước lượng từ:

$$\sigma_f^2(n_1, n_2) = \frac{I}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} [f(k_1, k_2) - m_f(k_1, k_2)]^2 \quad (2.16a)$$

$$\text{trong đó } m_f(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} f(k_1, k_2) \quad (2.16b)$$

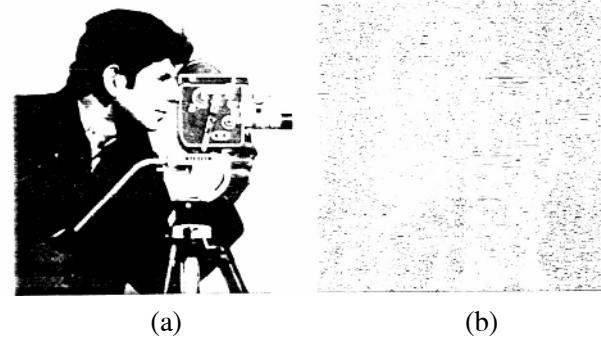
M thường chọn khoảng bằng 2. Bởi vì phươn g sai  $\sigma_f^2(n_1, n_2)$  được so sánh với một ngưỡng, có thể bỏ hệ số tỷ lệ xích  $1/(2M+1)^2$  trong (2.16a). Ngoài ra chỉ cần tính phươn g sai cục bộ  $\sigma_f^2$  cho  $(n_1, n_2)$  ở các điểm đi qua giá trị không của  $\nabla^2 f(n_1, n_2)$ . Hình 2.35 là kết quả áp dụng hệ ở Hình 2.34 cho ảnh trong Hình 2.33(a). So sánh các Hình 2.33(b) và 1.35 thấy giảm được nhiều đường biên giả. Hệ ở Hình 2.34 có thể coi như một phươn g pháp dựa trên gradient.

**Hình 2.33:** Bản đồ đường biên nhận được

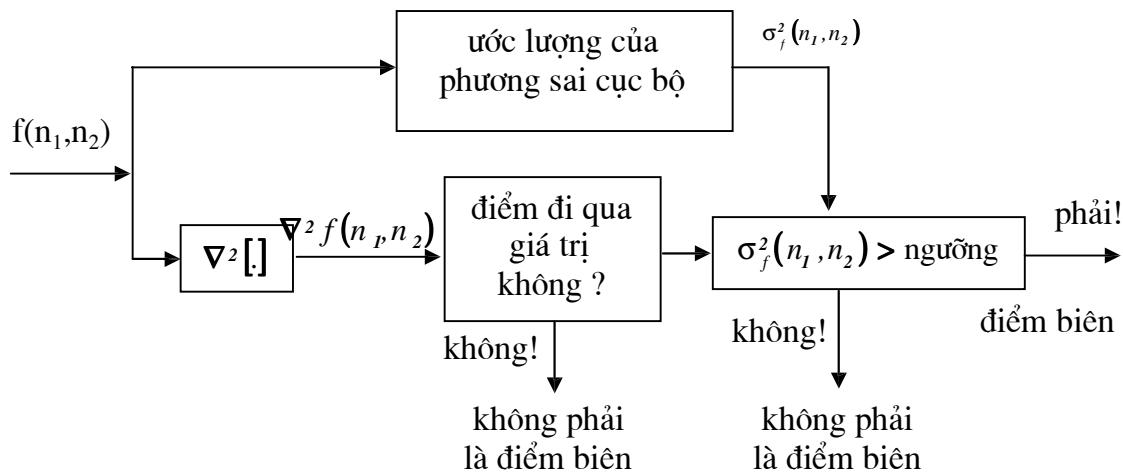
bằng bộ dò đường biên dựa trên Laplacian.

- (a) Ảnh gốc  $512 \times 512$  pixel;
- (b) Kết quả của tích chập ảnh trên hình (a) với  $h(n_1, n_2)$  trên Hình 2.32(a) rồi sau đó

tìm các điểm đi qua giá trị không.



(a) (b)



**Hình 2.34:** Hệ phát hiện đường biên dựa trên Laplacian mà không tạo ra nhiều đường biên giả.



**Hình 2.35:** Bản đồ đường biên nhận được bằng cách áp dụng hệ trên Hình 2.34 vào ảnh trên Hình 2.33(a).

Phương sai cục bộ  $\sigma_f^2(n_1, n_2)$  liên quan tới độ lớn gradient. So sánh  $\sigma_f^2(n_1, n_2)$  với một ngưỡng tương tự như so sánh độ lớn gradient với một ngưỡng. Yêu cầu  $\nabla^2 f(n_1, n_2)$  đi qua giá trị không ở biên có thể coi như như làm mảnh biên. Với cách hiểu này ta có thể thực hiện hệ ở Hình 2.34 bằng cách thoát tiên tính phương sai  $\sigma_f^2(n_1, n_2)$  và sau đó chỉ phát hiện điểm  $\nabla^2 f(n_1, n_2)$  đi qua giá trị không ở những điểm mà phương sai  $\sigma_f^2(n_1, n_2)$  ở trên một ngưỡng đã chọn.

### 3.3. PHÁT HIỆN ĐƯỜNG BIÊN BẰNG PHƯƠNG PHÁP MARR VÀ HI LDRETH

Trong hai tiết trước, ta đã thảo luận về các algorit phát hiện đường biên có thể từ ảnh đầu vào tạo ra bản đồ biên. Mar và Hildreth nhận thấy rằng có sự thay đổi cường độ rõ rệt xuất hiện ở những độ phân giải khác nhau trên hình. Chẳng hạn, trong cùng một ảnh có thể tồn tại cả vùng bóng mờ và vùng các chi tiết tinh tế được hội tụ sắc nét.

Sự phát hiện tối ưu những thay đổi cường độ đáng kể cần đến những toán tử đáp ứng được ở những độ phân giải khác nhau. Mar và Hildreth gợi ý rằng ảnh gốc cần phải giới hạn bằng tần số cắt khác nhau và algorit phát hiện đường biên được áp dụng cho từng ảnh riêng. Bản đồ tổng hợp đường biên sẽ có những đường biên ứng với từng độ phân giải.

Mar và Hildreth lập luận rằng bản đồ đường biên ở các độ phân giải khác nhau chứa thông tin quan trọng về các thông số có ý nghĩa về mặt vật lý. Thế giới thị giác được tạo ra bởi các thành phần như đường biên, vết xước, và bóng, mỗi thứ tập trung cao ở trong độ phân giải của chúng. Sự khu trú này cũng được phản ánh trong những thay đổi vật lý quan trọng như sự thay đổi độ chiếu sáng và hệ số phản xạ. Nếu cùng một đường biên xuất hiện trong một loạt bản đồ đường biên ứng với những độ phân giải

khác nhau, thì có nghĩa là nó đại biểu cho sự thay đổi cường độ của ảnh là do một hiện tượng vật lý duy nhất. Nếu biến chỉ xuất hiện trong một bản đồ đường biên, thì lý do có thể là sự thay đổi cường độ trong cùng một vùng ảnh do hai hiện tượng vật lý độc lập gây ra.

Để giới hạn bằng tần của ảnh ở các tần số cắt khác nhau, đáp ứng xung  $h(x,y)$  và đáp ứng tần số  $H(\Omega_x, \Omega_y)$  của bộ lọc thông thấp do Mar và Hildreth đề nghị có dạng Gauss và biểu diễn sau:

$$h(x,y) = \exp(-(x^2+y^2)/(2\pi\sigma^2)) \quad (2.17a)$$

$$H(\Omega_x, \Omega_y) = (2\pi^2\sigma^2)\exp(-\pi\sigma^2(\Omega_x^2+\Omega_y^2)/2) \quad (2.17b)$$

trong đó  $\sigma$  xác định tần số cắt, ứng với giá trị  $\sigma$  càng lớn tần số cắt càng thấp. Việc chọn dạng Gauss là căn cứ vào thực tế nó làm tròn và khu trú cả trong miền không gian và miền tần số. Một  $h(x,y)$  tron hiếm khi đưa ra sự thay đổi nào không tồn tại trong ảnh gốc.  $h(x,y)$  càng có tính khu trú cao thì càng ít xảy ra khả năng dịch chuyển vị trí biên.

Từ ảnh được làm tròn, có thể xác định các biên bằng những algorit phát hiện đường biên đã thảo luận trong hai tiết trên. Tuỳ theo phương pháp nào được sử dụng, có thể kết hợp phép toán lọc thông thấp trong công thức (2.17) và phép toán đạo hàm riêng. Chẳng hạn, lưu ý rằng  $\nabla^2[.]$  và nhân chập (\*) đều là tuyến tính, ta nhận được:

$$\begin{aligned} \nabla^2(f(x,y)*h(x,y)) &= f(x,y)*[\nabla^2h(x,y)] \\ &= f(x,y)*\left[\frac{\partial^2 h(x,y)}{\partial x^2} + \frac{\partial^2 h(x,y)}{\partial y^2}\right] \end{aligned} \quad (2.18)$$

Đối với hàm Gauss  $h(x,y)$  trong (2.17),  $\nabla^2h(x,y)$  và biến đổi Fourier của nó là

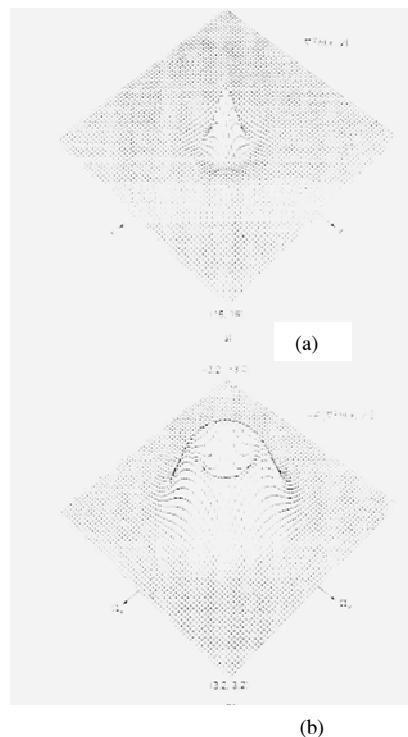
$$\nabla^2h(x,y) = \frac{e^{-(x^2+y^2)/(2\pi\sigma^2)}}{(\pi\sigma^2)^2}(x^2+y^2-2\pi\sigma^2) \quad (2.19a)$$

$$F[\nabla^2h(x,y)] = -(2\pi^2\sigma^2)\exp(-\pi\sigma^2(\Omega_x^2+\Omega_y^2))/2(\Omega_x^2+\Omega_y^2). \quad (2.19b)$$

Để đơn giản Mar và Hildreth chọn phương pháp phát hiện đường biên bằng cách tìm điểm không của  $\nabla^2f(x,y)$ . Việc giới hạn bằng tần của  $f(x,y)$  có xu thế làm giảm nhiễu, nhờ đó giảm vấn đề nhạy cảm với nhiễu được khi tìm các điểm đi qua giá trị không. Các hàm  $\nabla^2h(x,y)$  và  $-F[\nabla^2h(x,y)]$  trong (2.19) được biểu diễn trên Hình 2.36. Rõ ràng là phép tính  $f(x,y)*\nabla^2h(x,y)$  tương đương với lọc thông dải  $f(x,y)$ , trong đó  $\sigma^2$  trong

(2.19) là thông số điều khiển độ rộng của bộ lọc thông dải. Đối với một dãy  $f(n_1, n_2)$ , một cách tiếp cận đơn giản là thay thế  $x$  và  $y$  trong (2.19) bằng  $n_1$  và  $n_2$ .

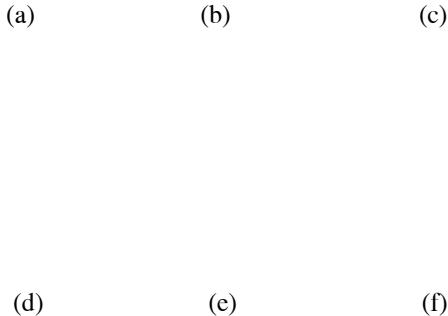
Hình 2.37 là một ví dụ của cách tiếp cận đang thảo luận. Các Hình 2.37(a), (b) và (c) là ba ảnh nhận được bằng cách làm mờ ảnh gốc trong Hình 2.33(a) bởi  $h(n_1, n_2)$ , - nhận được khi thay thế  $x$  và  $y$  của  $h(x, y)$  trong (2.17) bằng  $n_1$  và  $n_2$ , theo thứ tự với  $\sigma^2 = 4, 16$  và  $36$ . Các Hình 2.37(d), (e) và (f) là ảnh nhận được bằng cách tìm điểm đi qua giá trị không của  $f(n_1, n_2) * \nabla^2 h(x, y) |_{x=n_1, y=n_2}$ , với  $\nabla^2 h(x, y)$  nhận được từ (2.19a) khi tuân tự cho  $\sigma^2 = 4, 16$  và  $36$ . Mar và Hildreth sử dụng các bản đồ đường biên với các thang khác nhau để mô tả đối tượng, như trong các Hình 2.37(d),(e) và (f) trong công trình lý giải ảnh của họ.



**Hình 2.36:** Phác thảo theo phương trình (2.19), với  $\sigma^2 = 1$ , của:

- (a)  $\nabla^2 h(x, y)$
- (b)  $-F[\nabla^2 h(x, y)]$





**Hình 2.37:** Các bản đồ đường biên nhận được từ ảnh đã đi qua bộ lọc thông thấp. Các ảnh nhoè ứng với:

$$(a) \sigma^2 = 4; \quad (b) \sigma^2 = 16; \quad (c) \sigma^2 = 36.$$

Kết quả áp dụng algorit dựa trên Laplacian cho các ảnh nhoè ứng với:

$$(d) \sigma^2 = 4; \quad (e) \sigma^2 = 16; \quad (f) \sigma^2 = 36.$$

### 3.4. PHÁT HIỆN ĐƯỜNG BIÊN DỰA TRÊN MÔ HÌNH TÍN HIỆU

Các algorit phát hiện đường biên được thảo luận trên là các phư ơng pháp chung, theo nghĩa là chúng được phát triển độc lập của các bối cảnh ứng dụng. Một cách tiếp cận khác là phát triển một algorit phát hiện đường biên riêng cho một ứng dụng đặc biệt.

Chẳng hạn nếu ta biết dạng của một đường biên, thông tin này có thể được kết hợp để phát triển algorit phát hiện đường biên. Để minh họa một algorit phát hiện đường biên dùng riêng cho một ứng dụng, ta hãy xét bài toán phát hiện đường biên của động mạch vành trong một phim chụp X quang (angiogram).

Động mạch vành là hệ mạch máu bao quanh trái tim và cung cấp máu cho cơ tim. Sự hẹp động mạch vành hạn chế việc cung cấp đầy đủ máu tới trái tim, gây đau và làm tổn thương cơ tim. Sự tổn thương này được gọi là bệnh động mạch vành. Để xác định mức độ nặng nhẹ của bệnh động mạch vành, phải dựa vào phim chụp X quang (angiogram). Angiogram là ảnh chụp động mạch bằng tia X sau khi đã tiêm vào mạch máu một chất phản xạ tia X, thường là Iodine. Hình 2.38 một phim chụp X quang của

động mạch vành. Chỉ căn cứ vào cảm nhận thị giác cá nhân thì những người đọc phim khác nhau sẽ có những đánh giá khác nhau về mức độ trầm trọng của bệnh.

Độ đo về sự tắc nghẽn thường dùng nhất là tỉ lệ phần trăm tắc nghẽn, được định nghĩa là tỷ lệ co hẹp cực đại của động mạch trên một độ dài xác định. Một trong những cách tiếp cận để đánh giá tỉ lệ phần trăm tắc nghẽn bắt đầu bằng việc xác định đường biên động mạch từ một angiogram. Ta sẽ xem xét vấn đề phát hiện các đường biên động mạch.

Một mô hình hợp lý của angiogram  $f(n_1, n_2)$  là

$$f(n_1, n_2) = (v(n_1, n_2) + p(n_1, n_2)) * g(n_1, n_2) + w(n_1, n_2) \quad (2.20)$$

trong đó  $v(n_1, n_2)$  biểu thị mạch máu,  $p(n_1, n_2)$  biểu thị nền,  $g(n_1, n_2)$  biểu thị sự nhoè,  $w(n_1, n_2)$  là nhiễu nền. Hàm  $v(n_1, n_2)$  nhận được từ một mẫu hình nón suy rộng của mạch máu 3-D, liên tục và có các tiết diện ellip. Chọn hình dạng ellip vì số lượng tham số để mô tả đặc tính của nó ít và theo kinh nghiệm thì nó dẫn đến những kết quả đánh giá tỉ lệ phần trăm tắc nghẽn chính xác.

Tiết diện (1-D) của  $v(n_1, n_2)$  gồm một mạch máu được xác định toàn bộ bởi ba thông số, hai thông số đại biểu đường biên mạch máu và cái thứ ba liên quan tới hệ số suy hao tia X của Iodine. Tính liên tục của mạch máu được đảm bảo bởi một hàm bậc ba lót vào đường biên mạch máu. Nên  $p(n_1, n_2)$  được hình hoá bằng một đa thức hai chiều(2-D) bậc thấp. Các đa thức bậc thấp là những hàm rất trơn, và chọn chúng là vì những đối tượng ở nền như mô và bộ xương lớn hơn mạch máu nhiều lần. Hàm nhoè  $g(n_1, n_2)$  được mô hình hoá là dạng hàm Gauss hai chiều(2 -D), dùng nó để xét đến sự nhoè xảy ra ở những giai đoạn khác nhau của quá trình chụp. Nhiều  $w(n_1, n_2)$  là nhiễu nền ngẫu nhiên và giả thiết là nhiễu trắng. Các tham số trong mô hình của  $f(n_1, n_2)$  là những tham số mạch, các hệ số của đa thức  $p(n_1, n_2)$  và phương sai nhiễu.

Mạch, mô, xương và quá trình chụp X quang trong thực tế phức tạp hơn nhiều so với mô hình đơn giản đưa ra ở trên. Tuy nhiên, kinh nghiệm cho thấy mô hình này đã cho phép ước lượng tốt tình trạng đường biên mạch và tỉ lệ phần trăm tắc nghẽn tương ứng. Có thể ước lượng các tham số của mô hình bởi những quy trình khác nhau. Một khả năng là dùng phép ước lượng hợp lý tối đa để ước lượng tham số. Trong phương pháp gần đúng nhất, các tham số chưa biết ký hiệu là  $\theta$  được ước lượng bằng cách cực đại hoá hàm mật độ xác suất  $p_{f(n_1, n_2)|\theta}(f_0(n_1, n_2)|\theta_0)$ , trong đó  $f(n_1, n_2)$  là angiogram và  $\theta$  là tất cả các tham số chưa biết cần phải ước lượng. Phương pháp hợp lý tối đa đem áp

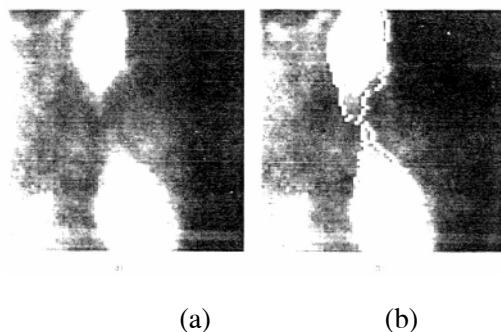
dụng để phát hiện đường biên mạch là một bài toán phi tuyến, nhưng đã được Pappas và Lim giải gần đúng. Hình 2.39 và 1.40 là kết quả minh họa ứng dụng phương pháp định giá hợp lý tối đa để phát hiện đường biên mạch máu dùng phiên bản (version) 1-D của mô hình 2-D trong Hình 2.20.

Trong phiên bản 1-D, thì  $f(n_1, n_2)$  trong Hình 2.20 được coi là dãy 1-D với biến  $n_1$  cho từng giá trị  $n_2$ . Tính toán được đơn giản đi rất nhiều khi sử dụng mô hình 1-D. Hình

2.39(a) là ảnh gốc angiogram  $80 \times 80$  pixel và Hình 2.39(b) là các đường biên mạch phát hiện được và in chồng lên ảnh gốc để so sánh. Hình 2.40 là một ví dụ khác. Phát triển một algorit phát hiện đường biên chuyên dùng cho một bài toán ứng dụng phức tạp hơn là áp dụng các algorit chung đã được thảo luận trong tiết trước, nhưng nó có khả năng phát hiện đường biên chính xác hơn.



- Hình 2.40:** Một ví dụ khác về phát hiện
- thành mạch máu trên phim X-quang
  - bằng mô hình hoá tín hiệu.
  - (a) Phim X-quang  $80 \times 80$  pixel;
  - (b) Phát hiện thành mạch máu.



**Hình 2.38:** Ảnh chụp X-quang động mạch vành.

## 4. PHÉP NỘI SUY ẢNH VÀ SỰ ĐÁNH GIÁ CHUYỂN ĐỘNG

Trong phép nội suy tín hiệu, ta phục hồi tín hiệu liên tục từ các mẫu. Phép nội suy tín hiệu có nhiều ứng dụng, có thể sử dụng trong việc thay đổi kích cỡ của ảnh số để cải thiện thể hiện của ảnh khi xem trên thiết bị hiển thị. Xét ảnh số cỡ  $64 \times 64$  pixel. Nếu thiết bị hiển thị hiện hình bậc không, từng cá thể pixel đều được nhìn thấy, ảnh thể hiện thành các khối. Nếu tăng kích thước ảnh bằng nội suy và lấy mẫu lại trước khi hiển thị thì ảnh xuất hiện sẽ mịn hơn và nhìn đẹp mắt hơn. Một dãy khung hình cũng có thể nội suy theo thứ nguyên thời gian. Ảnh động 24 khung/sec cũng có thể đổi thành tín hiệu truyền hình NTSC 60 mành / giây bằng phép nội suy. Cũng có thể dùng phép nội suy thời gian để cải thiện thể hiện của ảnh video chuyển động chậm.

Phép nội suy cũng có thể sử dụng trong các ứng dụng khác như mã hoá ảnh. Chẳng hạn, một cách tiếp cận đơn giản để làm giảm tốc độ bít là loại bỏ một số pixel hoặc một vài khung và tạo lại chúng từ pixel và khung đã mã hoá.

### 4.1. PHÉP NỘI SUY KHÔNG GIAN

Xét một dãy  $f(n_1, n_2)$  trong không gian 2-D, là kết quả của phép lấy mẫu tín hiệu tương tự (analog)  $f(x, y)$  bằng một bộ chuyển đổi A/D lý tưởng:

$$f(n_1, n_2) = f_c(x, y) \Big|_{x=n_1 T_1, y=n_2 T_2} \quad (2.21)$$

nếu  $f_c(x, y)$  bị giới hạn bằng tần và các tần số lấy mẫu  $1/T_1$  và  $1/T_2$  đều cao hơn tốc độ Nyquist, thì có thể phục hồi  $f_c(x, y)$  từ  $f(n_1, n_2)$  bằng bộ chuyển đổi D/A lý tưởng là

$$f_c(x, y) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f(n_1, n_2) h(x - n_1 T_1, y - n_2 T_2) \quad (2.22)$$

trong đó  $h(x, y)$  là đáp ứng xung của một bộ lọc tương tự (analog) thông thấp lý tưởng:

$$h(x, y) = \frac{\sin \frac{\pi}{T_1} x \sin \frac{\pi}{T_2} y}{\frac{\pi}{T_1} x \frac{\pi}{T_2} y} \quad (2.23)$$

Có nhiều khó khăn trong việc sử dụng (2.22) và (2.23) để nội suy ảnh. Ảnh tương tự (analog)  $f_c(x, y)$  ngay cả khi dùng bộ lọc chống chùng phổ (antialiasing) cũng

không hoàn toàn là bị giới hạn bằng tần, cho nên khi lấy mẫu sẽ bị chồng phỏ. Ngoài ra  $h(x,y)$  trong (2.23) là một hàm mở rộng vô hạn vì vậy sử dụng (2.22) để đánh giá  $f_c(x,y)$  trong thực tế không thể thực hiện được. Để lấy xấp xỉ phép nội suy bằng (2.22) và (2.23) có thể sử dụng bộ lọc thông thấp  $h(x,y)$  vốn bị giới hạn không gian. Đối với bộ lọc  $h(x,y)$  giới hạn không gian thì phép tổng trong (2.22) có một số hữu hạn số hạng khác không. Nếu  $h(x,y)$  là một hàm cửa sổ hình chữ nhật là:

$$h(x,y) = 1, \quad -\frac{T_1}{2} \leq x \leq \frac{T_1}{2}, \quad -\frac{T_2}{2} \leq y \leq \frac{T_2}{2} \quad (2.24)$$

thì được gọi là phép nội suy bậc không. Trong phép nội suy bậc không,  $\hat{f}_c(x,y)$  được chọn là  $f(n_1, n_2)$  ở pixel gần điểm  $(x,y)$  nhất. Những ví dụ khác của  $h(x,y)$  được sử dụng phổ cập hơn là những hàm cho hình dạng trơn hơn như hàm Gauss giới hạn không gian hoặc bộ lọc cửa sổ thông thấp lý tưởng.

Một phương pháp đơn giản khác được sử dụng nhiều trong thực tế là phép nội suy song tuyến tính. Trong phương pháp này  $\hat{f}_c(x,y)$  được định giá bằng phép tổ hợp tuyến tính của  $f(n_1, n_2)$  ở 4 pixel gần nhất. Giả sử ta muốn định giá  $f(x,y)$  cho  $n_1 T_1 \leq x \leq (n_1+1)T_1$  và  $n_2 T_2 \leq y \leq (n_2+1)T_2$  biểu diễn trên Hình 2.41. Giá trị  $\hat{f}_c(x,y)$  trong phương pháp phép nội suy song tuyến tính là:

$$\begin{aligned} \hat{f}_c(x,y) &= (1-\Delta_x)(1-\Delta_y)f(n_1, n_2) + \Delta_y(1-\Delta_x)f(n_1, n_2+1) + \Delta_x(1-\Delta_y)f(n_1+1, n_2) \\ &\quad + \Delta_x\Delta_yf(n_1+1, n_2+1) \end{aligned} \quad (2.25a)$$

$$\text{trong đó } \Delta_x = (x - n_1 T_1) / T_1 \quad (2.25b)$$

$$\text{và } \Delta_y = (y - n_2 T_2) / T_2 \quad (2.25c)$$

Một phương pháp khác là phép nội suy đa thức. Xét một vùng không gian cục bộ  $3 \times 3$  hoặc  $5 \times 5$  pixel qua đó  $f(x,y)$  được lấy xấp xỉ bằng một đa thức. Ảnh nội suy  $\hat{f}_c(x,y)$  là

$$\hat{f}_c(x,y) = \sum_{i=1}^N S_i \phi_i(x,y) \quad (2.26)$$

trong đó  $\phi_i(x,y)$  là một số hạng của đa thức. Một ví dụ của  $\phi_i(x,y)$  khi  $N = 6$  là

$$\phi_i(x,y) = 1, x, y, x^2, y^2, xy. \quad (2.27)$$

hệ số  $S_i$  có thể được xác định bằng cách lấy tối thiểu của

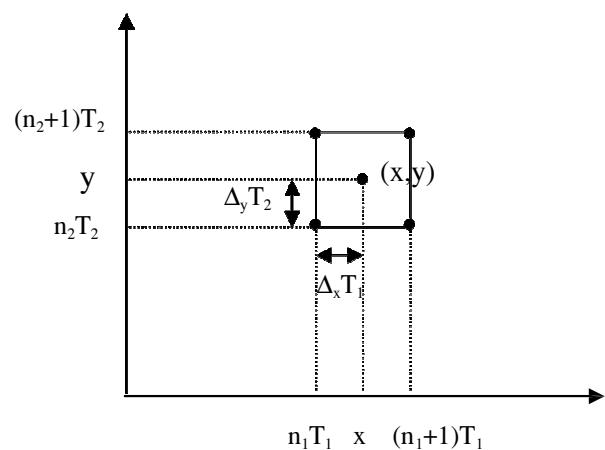
$$\text{Error} = \sum_{(n_1, n_2) \in \Psi} \left[ f(x, y) - \sum_{i=1}^N S_i \phi_i(x, y) \right]^2 \Big|_{x=n_1 T_1, y=n_2 T_2} \quad (2.28)$$

Trong đó  $\Psi$  là những pixel trên đó  $f(x, y)$  được lấy xấp xỉ. Giải phương trình (2.28) là bài toán tuyến tính đơn giản, bởi vì  $\phi_i(x, y)$  cố định. Ưu điểm của phép nội suy đa thức là  $\hat{f}_c(x, y)$  trơn và việc định giá  $\partial \hat{f}_c(x, y)/\partial x$  và  $\partial \hat{f}_c(x, y)/\partial y$  đơn giản.  $\partial \hat{f}_c(x, y)/\partial x$  và  $\partial \hat{f}_c(x, y)/\partial y$  là các đạo hàm riêng được dùng trong những ứng dụng như phát hiện đường biên và ước lượng di động. Ngoài ra đem khớp một đa thức với số hệ số ít hơn số pixel trong vùng  $\Psi$  ở công thức (2.28), có thể làm trơn nhiễu ở một mức độ nhất định. Việc làm trơn nhiễu đặc biệt có lợi trong các ứng dụng có sử dụng đạo hàm riêng  $\partial \hat{f}_c(x, y)/\partial x$  và  $\partial \hat{f}_c(x, y)/\partial y$ .

Cũng có thể sử dụng algorit ước lượng di động sẽ thảo luận trong tiết sau để phát triển những sơ đồ nội suy không gian.

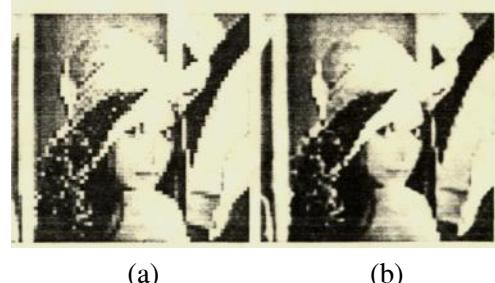
Hình 2.42 là một ví dụ phép nội suy ảnh. Hình 2.42(a) là ảnh  $265 \times 256$  pixel được tạo ra từ phép nội suy bậc không một ảnh gốc  $64 \times 64$  pixel. Hình 2.42(b) là ảnh  $256 \times 256$  pixel nhận được từ phép nội suy song tuyến tính một ảnh gốc  $64 \times 64$  pixel.

**Hình 2.41:** Vùng ở đó  $f_c(x, y)$  được nội suy từ 4 pixel lân cận  $f_c(n_1 T_1, n_2 T_2)$ ,  $f_c((n_1+1)T_1, n_2 T_2)$ ,  $f_c(n_1 T_1, (n_2+1)T_2)$ ,  $f_c((n_1+1)T_1, (n_2+1)T_2)$ .



**Hình 2.42:** Ví dụ về phép nội suy không gian.

- (a) Ảnh 265 x 256 pixel được tạo ra từ phép nội suy bậc-không một ảnh gốc 64 x 64 pixel;
- (b) Ảnh 256 x 256 pixel nhận được từ phép nội suy song tuyến tính ảnh gốc đã sử dụng trong hình (a).



## 4.2. UỐC LƯỢNG CHUYỂN ĐỘNG

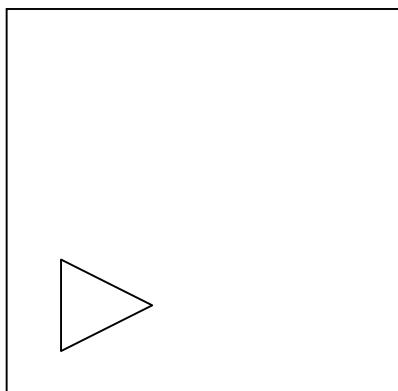
Có thể từ những khung hình đã có tạo ra các khung hình mới bằng nội suy thời gian. Khác với nội suy không gian, nội suy thời gian yêu cầu lưu trữ một khối lượng lớn.

Vì vậy, thường phải tạo ra một khung mới giữa hai khung liền kề, một cái thuộc về quá khứ, cái kia thuộc về tương lai so với khung được tạo ra.

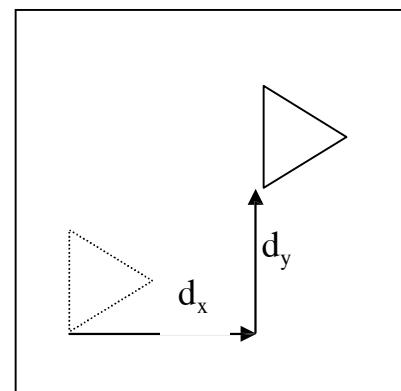
Phương pháp đơn giản nhất thường dùng trong thực tế là phương pháp bậc - không (zero-order hold method) tạo ra một khung mới bằng cách lặp lại khung đã có ở thời điểm gần kề nhất.

Khi biến đổi ảnh động 24 khung/sec sang tín hiệu NTSC 60 trường/sec, từ một khung ảnh động tạo ra 3 trường kế tiếp nhau, sau đó từ khung ảnh động kế theo lại tạo ra 2 trường kế tiếp nhau nữa, quá trình cứ như thế lặp lại cho đến khi hết các ảnh động. Đó là *phương pháp gỡ dần 3:2* (3:2 pull-down method). Với đa số cảnh (scene) không có chuyển động toàn bộ (global motion) lớn thì kết quả khá tốt. Tuy vậy, khi có chuyển động toàn bộ lớn, sẽ có hiện tượng giật giật. Một cách để cải thiện là bù chuyển động.

Ảnh động hoặc truyền hình quảng bá là một dãy những khung tĩnh được hiển thị liên tiếp nhau với tốc độ cao. Tốc độ (hay nhịp hiển thị) cần thiết để gây cảm nhận như chuyển động tự nhiên thường phải khá cao, đủ đảm bảo một độ dư thừa về thời gian giữa các khung kề. Phần lớn biến thiên cường độ từ khung này sang khung kế theo sau là do đối tượng chuyển động. Quá trình xác định vận động (movement) của các đối tượng trong một dãy khung ảnh gọi là *ước lượng chuyển động*. Xử lý ảnh có xét đến sự tồn tại của chuyển động gọi là *xử lý ảnh có bù chuyển động*.



(a)  $f(x, y, t_{-1})$



(b)  $f(x, y, t_0)$

**Hình 2.43:** Ảnh tĩnh tiến với di chuyển ( $d_x, d_y$ ).

- (a)  $f(x, y, t_{-1})$ ;
- (b)  $f(x, y, t_0)$ .

Xử lý ảnh có bù chuyển động có nhiều ứng dụng. Một ứng dụng là nội suy ảnh. Bằng cách ước lượng các thông số chuyển động ta có thể tạo ra một khung mới giữa hai khung đã có. Ứng dụng phép xử lý có bù chuyển động cho nội suy ảnh sẽ được thảo luận ở tiết sau. Một ứng dụng nữa là phục hồi ảnh. Nếu có thể ước lượng các thông số chuyển động và nhận biết các vùng trong những khung khác nhau ở đó cường độ ảnh bằng hoặc gần bằng nhau, thì có thể lọc thời gian ở những vùng đó. Những ứng dụng cho phục hồi ảnh sẽ được thảo luận ở chương 3. Xử lý ảnh có bù chuyển động cũng có thể ứng dụng cho mã hoá ảnh. Bằng cách dựa vào các khung trước đó dự báo cường độ khung đương thời, có thể giới hạn mã hoá ở các hiệu số cường độ giữa khung đương thời và khung đương thời dự báo. Ngoài ra, chúng ta cũng có thể loại bỏ vài khung và xây dựng lại những khung bị loại bỏ bằng nội suy từ các khung đã mã hoá. Những ứng dụng cho mã hoá ảnh sẽ thảo luận ở chương 3.

Bài toán ước lượng chuyển động mà ta xét ở đây là chuyển động tĩnh tiến của các đối tượng. Đặt  $f(x, y, t_{-1})$  và  $f(x, y, t_0)$  theo thứ tự là cường độ ảnh ở những thời điểm  $t_{-1}$  và  $t_0$ . Ta gọi các khung  $f(x, y, t_{-1})$  và  $f(x, y, t_0)$  là khung quá khứ và khung đang xét. Giả thiết rằng:

$$f(x, y, t_0) = f(x - d_x, y - d_y, t_{-1}) \quad (2.29)$$

trong đó  $d_x$  và  $d_y$  là dịch chuyển ngang và thẳng đứng giữa  $t_{-1}$  và  $t_0$ . Hình 2.43 biểu diễn một ví dụ của  $f(x, y, t_{-1})$  và  $f(x, y, t_0)$  thoả mãn phương trình (2.29). Giả sử có chuyển động đều giữa  $t_{-1}$  và  $t_0$ ,

$$f(x, y, t) = f\left(x - v_x(t - t_{-1}), y - v_y(t - t_{-1}), t_{-1}\right), \quad t_{-1} \leq t \leq t_0 \quad (2.30)$$

trong đó  $v_x$  và  $v_y$  là các tốc độ theo phương nằm ngang và phương thẳng đứng.

Một hệ quả trực tiếp của (2.30) là một phương trình vi phân liên hệ  $v_x$  và  $v_y$  với  $\delta f(x, y, t)/\delta x$ ,  $\delta f(x, y, t)/\delta y$ ,  $\delta f(x, y, t)/\delta t$  có hiệu lực trên miền không - thời gian trong đó giả thiết chuyển động là tịnh tiến đều. Để suy ra hệ thức này, ta ký hiệu  $f(x, y, t_{-1})$  bằng  $s(x, y)$ :

$$s(x, y) = f(x, y, t_{-1}). \quad (2.31)$$

Từ (2.30) và (2.31):

$$f(x, y, t) = s(\alpha(x, y, t), \beta(x, y, t)), \quad t_{-1} \leq t \leq t_0 \quad (2.32)$$

trong đó  $\alpha(x, y, t) = x - v_x(t - t_{-1})$  (2.32b)

và  $\beta(x, y, t) = y - v_y(t - t_{-1})$  (2.32c)

Từ (2.32), giả thiết rằng  $\delta f(x, y, t)/\delta x$ ,  $\delta f(x, y, t)/\delta y$ ,  $\delta f(x, y, t)/\delta t$  tồn tại, ta nhận được:

$$\frac{\partial f(x, y, t)}{\partial x} = \frac{\partial s}{\partial \alpha} \frac{\partial \alpha}{\partial x} + \frac{\partial s}{\partial \beta} \frac{\partial \beta}{\partial x} = \frac{\partial s}{\partial \alpha} \quad (2.33a)$$

$$\frac{\partial f(x, y, t)}{\partial y} = \frac{\partial s}{\partial \alpha} \frac{\partial \alpha}{\partial y} + \frac{\partial s}{\partial \beta} \frac{\partial \beta}{\partial y} = \frac{\partial s}{\partial \beta} \quad (2.33b)$$

$$\frac{\partial f(x, y, t)}{\partial t} = \frac{\partial s}{\partial \alpha} \frac{\partial \alpha}{\partial t} + \frac{\partial s}{\partial \beta} \frac{\partial \beta}{\partial t} = -v_x \frac{\partial s}{\partial \alpha} - v_y \frac{\partial s}{\partial \beta}. \quad (2.33c)$$

Từ (2.33),

$$v_x \frac{\partial f(x, y, t)}{\partial x} + v_y \frac{\partial f(x, y, t)}{\partial y} + \frac{\partial f(x, y, t)}{\partial t} = 0. \quad (2.34)$$

Phương trình (2.34) gọi là phương trình ràng buộc không - thời gian và có thể được mở rộng để xét đến những loại chuyển động khác, như phóng to, thu nhỏ ống kính (zooming).

Giả thiết tịnh tiến đơn giản dẫn đến phương trình (2.29) và giả thiết bổ sung coi phép tịnh tiến có tốc độ đều dẫn đến (2.30) đã đặt ra nhiều hạn chế. Chẳng hạn, không được cho đối tượng quay, không phóng to, thu nhỏ ống kính, các vùng không bị trùm phủ bởi sự tịnh tiến đối tượng, các đối tượng không được chuyển động với những tốc độ  $v_x$  và  $v_y$  khác nhau. Tuy vậy, bằng cách giả thiết chỉ có chuyển động tịnh tiến đều ở cục bộ, và chỉ ước lượng 2 thông số chuyển động ( $d_x, d_y$ ) hay ( $v_x, v_y$ ) ở mỗi pixel hay ở mỗi hình con, các công thức (2.29) và (2.34) vẫn có hiệu lực ở những vùng nền không bị ảnh hưởng bởi chuyển động đối tượng.

Những vùng này chiếm một phần đáng kể của dãy khung hình. Ngoài ra, nếu ta nhận biết những vùng ở đó ước lượng chuyển động không chính xác, có thể loại bỏ phép xử lý bù chuyển động ở những vùng đó. Chẳng hạn trong nội suy ảnh, ta có thể giả thiết  $v_x = v_y = 0$ .

Có thể phân các phương pháp ước lượng chuyển động thành hai nhóm lớn, là các phương pháp thích ứng vùng (region matching) và các phương pháp ràng buộc không-thời gian. Các phương pháp thích ứng vùng dựa trên (2.29) và các phương pháp ràng buộc không-thời gian dựa trên (2.34). Ta hãy thảo luận về các phương pháp thích ứng vùng trước.

#### 4.2.1. CÁC PHƯƠNG PHÁP THÍCH ỨNG VÙNG

Nội dung các phương pháp thích ứng vùng là xét một vùng nhỏ trong khung hình và tìm sự chuyển động có vẻ “khớp nhất” trong số những vùng có thể của khung hình lân cận. Trong các phương pháp thích ứng vùng, vector chuyển động ( $d_x, d_y$ ) được ước lượng bằng cách lấy cực tiểu:

$$\text{Error} = \iint_{(x,y) \in R} C[f(x, y, t_0), f(x - d_x, y - d_y, t_{-1})] dx dy \quad (2.35)$$

trong đó  $R$  là vùng không gian cục bộ sử dụng để ước lượng ( $d_x, d_y$ ), và  $C[.,.]$  là một độ đo để phân biệt sự khác nhau giữa hai argumen. Các tích phân trong công thức (2.35)

có thể thay thế bằng những tổng nếu được lấy mẫu ở biến không gian ( $x, y$ ). Nếu ta ước lượng ( $d_x, d_y$ ) ở thời điểm  $t_0$ , vùng  $R$  là vùng không gian cục bộ bao quanh vị trí không gian đặc biệt ở đó tiến hành ước lượng ( $d_x, d_y$ ). Kích cỡ của  $R$  phụ thuộc nhiều yếu tố. Nếu chọn  $R$  quá lớn thì giả thiết ( $d_x, d_y$ ) gần như không đổi trên vùng  $R$  sẽ không còn hiệu lực và sẽ phải tính toán nhiều để ước lượng sai số. Nếu chọn  $R$  quá nhỏ thì các giá trị ước lượng có thể rất nhạy với tạp âm. Dựa theo những nhận định trên thì sự lựa chọn phù hợp là  $5 \times 5$  pixel. Có nhiều cách chọn hàm phân biệt  $C[\cdot, \cdot]$  khác nhau. Thông thường sử dụng hai cách chọn là hiệu bình phương và hiệu tuyệt đối giữa hai acgumen.

Với những cách chọn đó (2.35) rút gọn lại là:

$$\text{Error} = \iint_{(x,y) \in R} [f(x, y, t_0) - f(x - d_x, y - d_y, t_{-1})]^2 dx dy \quad (2.36)$$

$$\text{Error} = \iint_{(x,y) \in R} |f(x, y, t_0) - f(x - d_x, y - d_y, t_{-1})| dx dy \quad (2.37)$$

Hàm  $f(x, y, t_0) - f(x - d_x, y - d_y, t_{-1})$  được gọi là hàm dịch chuyển. Trong các ứng dụng điển hình của phép xử lý bù chuyển động, sự lựa chọn hàm phân biệt không ảnh hưởng nhiều đến hiệu năng hệ thống. Trong phạm vi mà công thức (2.29) còn hiệu lực, thì biểu thức sai số trong (2.36) hoặc (2.37) sẽ bằng 0 ở một giá trị thích đáng của  $(d_x, d_y)$ .

Cực tiểu hóa Error trong (2.36) hoặc (2.37) là một bài toán phi tuyến. Có nhiều nỗ lực để giải quyết bài toán phi tuyến đó, dẫn đến nhiều phương pháp giải khác nhau, nhưng tựu trung có thể phân thành hai loại chính là phương pháp thích ứng khối và phương pháp đệ quy. Ta thảo luận phương pháp thích ứng khối trước.

Một cách tiếp cận trực tiếp để giải quyết vấn đề cực tiểu trên là định giá Error cho mọi vị trí  $(d_x, d_y)$  trên một phạm vi phù hợp và chọn  $(d_x, d_y)$  ở đó Error là cực tiểu. Trong cách tiếp cận đó, một khối các cường độ pixel ở thời gian  $t_0$  được khớp phù hợp trực tiếp (matched directly) với một khối ở thời gian  $t_{-1}$ . Đó là cơ sở của các phương pháp thích ứng khối. Vì biểu thức Error phải được định giá ở nhiều giá trị của  $(d_x, d_y)$ , phương pháp ước lượng  $(d_x, d_y)$  này chi phí tính toán quá lớn và nhiều phương pháp đã được khai triển để làm giảm tính toán. Trong phương pháp giản hóa đầu tiên, ta giả thiết  $(d_x, d_y)$  là hằng số trên một khối  $7 \times 7$  pixels. Với giả thiết đó, ta chia ảnh thành nhiều khối và ước lượng  $(d_x, d_y)$  cho mỗi khối. Mặc dù ta thường chọn kích thước khối như là kích cỡ của  $R$  trong (2.35), nhưng cũng không nhất thiết phải làm như vậy. Trong một phương pháp giản hóa khác, ta có thể giới hạn khoảng tìm kiếm ở các giá trị nguyên của  $(d_x, d_y)$ . Ngoài việc làm giảm khoảng tìm kiếm từ các biến  $(d_x, d_y)$  liên tục

sang các biến rời rạc, sự giới hạn khoảng tìm kiếm ở các giá trị nguyên cho phép ta xác định  $f(n_1 - d_x, n_2 - d_y, t_{-1})$  cần thiết để ước lượng biểu thức Error, mà không nội suy. Tuy nhiên các giá trị ước lượng ( $d_x, d_y$ ) chỉ hạn chế ở những giá trị rời rạc.

Ta có thể làm giảm đòi hỏi tính toán ở các phương pháp thích ứng khối hơn nữa bằng cách sử dụng thủ tục tìm kiếm hiệu quả hơn. Một phương pháp được gọi là phương pháp tìm kiếm ba bước được minh họa trong Hình 2.44. Trong bước đầu tiên của phương pháp này, biểu thức sai số được định giá ở 9 giá trị của  $(d_x, d_y)$ , được đánh dấu là “1” và dùng dấu chấm tròn ghi vị trí trên hệ toạ độ. Trong 9 giá trị  $(d_x, d_y)$  đó, ta chọn cái ứng với Error nhỏ nhất. Giả sử đạt Error nhỏ nhất ở  $(d_x = 3, d_y = -3)$ . Trong bước thứ hai, ta định giá biểu thức sai số ở 8 giá trị khác của  $(d_x, d_y)$ , được đánh dấu là “2” và dùng dấu chấm vuông ghi vị trí trên hệ toạ độ. Bây giờ ta chọn  $(d_x, d_y)$  từ 9 giá trị (8 giá trị mới và  $(d_x = 3, d_y = -3)$ ). Thủ tục này được lặp lại thêm một lần nữa. Đến cuối bước ba, ta có ước lượng của  $(d_x, d_y)$ . Thủ tục này có thể mở rộng ra trên ba bước để làm tăng phạm vi có thể của những vị trí  $(d_x, d_y)$ . Một phương pháp tìm kiếm khác là thoát tiên tìm  $d_x$  bằng cách tìm  $(d_x, 0)$ . Khi đã ước lượng được  $d_x$ , chẳng hạn đó là  $\hat{d}_x$ , thì ước lượng  $d_y$  bằng cách tìm  $(\hat{d}_x, d_y)$ . Nếu ta muốn cải thiện ước lượng hơn nữa, thì có thể ước lượng lại  $d_x$  bằng cách tìm  $(d_x, \hat{d}_y)$ , trong đó  $\hat{d}_y$  là ước lượng của  $d_y$  nhận được trong bước trước. Mỗi bước trong thủ tục đó, ta chỉ ước lượng một tham số, đơn giản hơn ước lượng hai tham số đồng thời. Các phương pháp kinh nghiệm (heuristic) làm giảm khối lượng tính toán bằng cách làm giảm số các giá trị của  $(d_x, d_y)$  ở đó biểu thức sai số được định giá. Tuy nhiên Error ở giá trị ước lượng  $(d_x, d_y)$  có thể không phải là một cực tiểu toàn bộ (global minimum). Trong các phương pháp thích ứng khối, ta ước lượng  $(d_x, d_y)$  bằng cách định giá Error ở một vài bộ  $(d_x, d_y)$  xác định. Có một biến thể của phương pháp, là sử dụng các algorit giảm như các phương pháp Newton - Raphson và Davidon - Fletcher-Powell để giải quyết bài toán phi tuyến tìm cực tiểu của Error đối với  $(d_x, d_y)$ . Trong loại này của các algorit, thủ tục đệ quy được sử dụng để cải thiện ước lượng trong mỗi lần lặp, vì thế chúng được gọi là các phương pháp đệ quy.

Gọi  $(\hat{d}_x(k), \hat{d}_y(k))$  là ước lượng của  $(d_x, d_y)$  sau  $k$  lần lặp. Ước lượng  $(d_x, d_y)$  sau  $k+1$  lần lặp là  $(\hat{d}_x(k+1), \hat{d}_y(k+1))$  biểu diễn là:

$$\hat{d}_x(k+1) = \hat{d}_x(k) + u_x(k) \quad (2.38a)$$

$$\hat{d}_y(k+1) = \hat{d}_y(k) + u_y(k) \quad (2.38b)$$

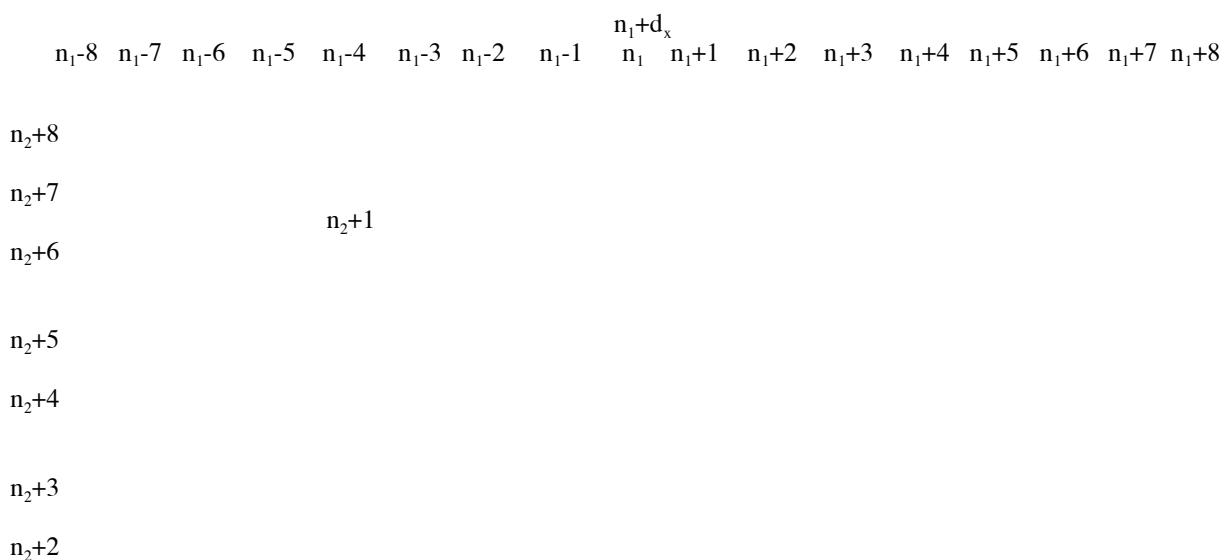
trong đó  $u_x(k)$ ,  $u_y(k)$  là các số hạng cập nhật hoặc số hạng hiệu chỉnh. Các số hạng cập nhật thay đổi theo phương pháp giảm được sử dụng. Chẳng hạn, nếu ta dùng phương pháp giảm nhanh nhất, thì (2.38) trở thành :

$$\hat{d}_x(k+1) = \hat{d}_x(k) - \varepsilon \frac{\partial Error(d_x, d_y)}{\partial d_x} \Big|_{d_x = \hat{d}_x(k), d_y = \hat{d}_y(k)} \quad (2.39a)$$

$$\hat{d}_y(k+1) = \hat{d}_y(k) - \varepsilon \frac{\partial Error(d_x, d_y)}{\partial d_y} \Big|_{d_x=\hat{d}_x(k), d_y=\hat{d}_y(k)} \quad (2.39b)$$

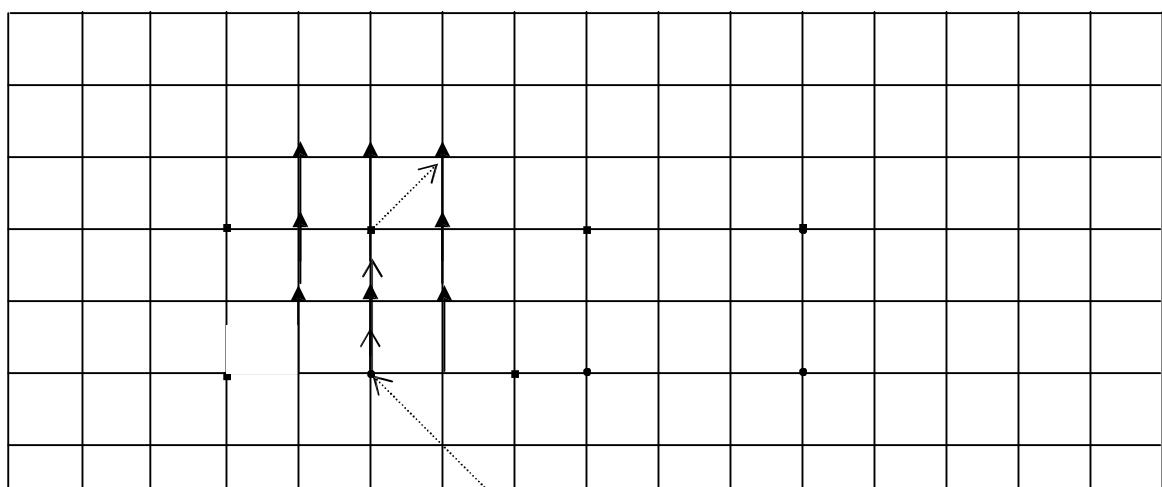
trong đó  $\epsilon$  là kích cỡ một bước, có thể điều chỉnh được, và  $\text{Error}(d_x, d_y)$  là Error trong công thức (2.35), hàm của  $d_x$  và  $d_y$  với mỗi giá trị đã cho R. Các phương pháp đệ quy thường liên quan tới đạo hàm riêng và có khuynh hướng là nhạy cảm với sự tồn tại nhiều hoặc những chi tiết ảnh tinh vi. Làm tròn ảnh trước khi ước lượng chuyển động thường cải thiện hiệu năng các phương pháp đệ quy.

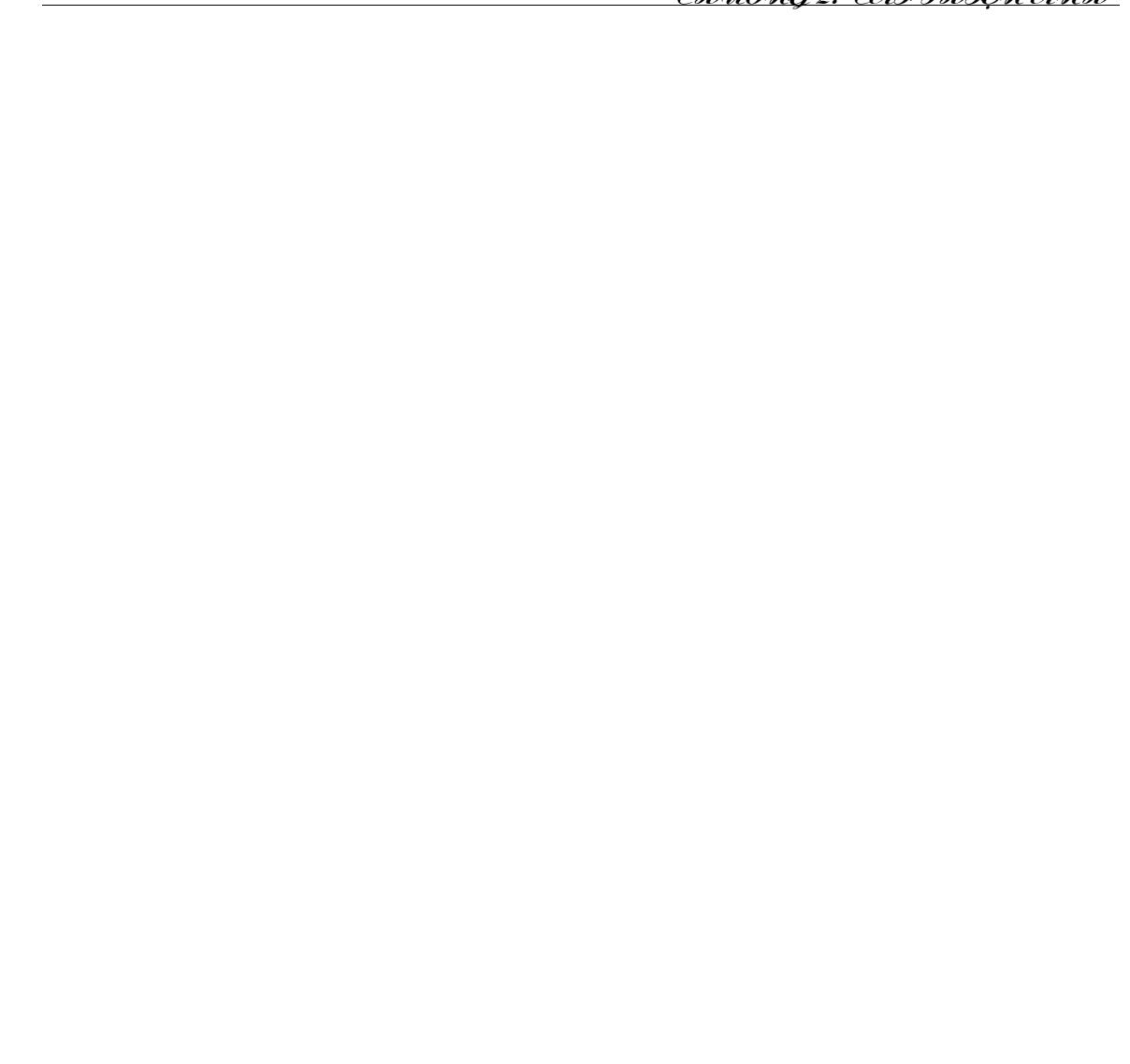
Trong các phương pháp đệ quy,  $(d_x, d_y)$  không bị giới hạn ở các giá trị nguyên và có thể được ước lượng chính xác trong điền ảnh con (subpixel). Các số hạng cập nhật thường liên quan đến sự định giá các đạo hàm riêng của  $\text{Error}(d_x, d_y)$ , gồm việc định giá  $f(x, y, t_{-1})$  và các đạo hàm riêng của nó ở một điểm tùy ý trong không gian. Trong thực tế  $f(x, y, t_{-1})$  chỉ được biết khi  $x = n_1 T_1$  và  $y = n_2 T_2$ . Để định giá các đại lượng cần thiết ở điểm không gian tùy ý  $(x, y)$ , ta có thể sử dụng kỹ thuật nội suy không gian được thảo luận trong tiết 1.4.1



EXERCISE 2: EDIT THE GRAPH

3  
3  
3  
  
2                    3    2  
3  
2  
3  
2  
2  
  
3  
3  
3  
  
2        1            2 2 1            1  
  
**9**  
**5**  
  
2        2            2





Trong các phương pháp đê quy,  $(d_x, d_y)$  thường được ước lượng ở mỗi pixel. Khi sử dụng phép đê quy liên quan tới công thức (2.38)  $(\hat{d}_x(\theta), \hat{d}_y(\theta))$  thường nhận được ước lượng ở pixel kề trong cùng một đường quét ngang, trong đường quét ngang kề theo hoặc trong khung kề theo. Các phương pháp này được gọi là ước lượng đê quy pel phân tử ảnh (picture element), theo thứ tự là đê quy ngang, dọc và đê quy thời gian. Khi cho  $(\hat{d}_x(k), \hat{d}_y(k))$ , ta chỉ có thể sử dụng phép đê quy liên quan công thức (2.38) một lần cho mỗi pixel rồi chuyển động tới pixel tiếp theo. Hoặc là, ta cũng có thể sử dụng phép đê quy nhiều hơn một lần để ước lượng  $(d_x, d_y)$  chính xác hơn trước khi chuyển động sang pixel tiếp theo.

Mặc dù ta phân loại các phương pháp thích ứng vùng thành phương pháp thích ứng khối và phương pháp đệ quy như thường làm trên báo chí khoa học, nhưng đường ranh giới giữa hai loại chỉ là ranh giới mờ. Bằng cách chọn lưới (grid) định giá biểu thức sai số dày hơn, ta cũng có thể định giá ( $d_x$ ,  $d_y$ ) trên từng subpixel với các phương pháp thích ứng khối. Ngoài ra thủ tục tìm kiếm ba bước ở trên có thể được xem như phương pháp đệ quy ở đó ước lượng được cải thiện bằng phép lặp. Nhược điểm chính của các phương pháp thích ứng vùng là yêu cầu khối lượng tính toán lớn. Tuy chỉ phải ước lượng hai tham số  $d_x$  và  $d_y$ , việc giải bài toán phi tuyến ở mỗi pixel hoặc ở mỗi ảnh con đều có thể đòi hỏi chi phí tính toán lớn.

#### 4.2.2. CÁC PHƯƠNG PHÁP RÀNG BUỘC KHÔNG - THỜI GIAN

Các algorit của loại này được dựa trên phương trình ràng buộc không - thời gian (2.34), có thể xem như một phương trình tuyến tính cho hai số chưa biết  $v_x$  và  $v_y$  dưới giả thiết là  $\frac{\partial f(x, y, t)}{\partial x}$ ,  $\frac{\partial f(x, y, t)}{\partial y}$  và  $\frac{\partial f(x, y, t)}{\partial t}$  đã biết. Bằng cách ước lượng  $\frac{\partial f(x, y, t)}{\partial x}$ ,  $\frac{\partial f(x, y, t)}{\partial y}$  và  $\frac{\partial f(x, y, t)}{\partial t}$  ở nhiều điểm  $(x_i, y_i, t_i)$  với  $1 \leq i \leq N$ , ở đó  $v_x$  và  $v_y$  được giả định là hằng số, ta có thể nhận được một tập hợp phương trình tuyến tính quá xác định (overdetermined):

$$v_x \frac{\partial f(x, y, t)}{\partial x} \Big|_{(x_i, y_i, t_i)} + v_y \frac{\partial f(x, y, t)}{\partial y} \Big|_{(x_i, y_i, t_i)} + \frac{\partial f(x, y, t)}{\partial t} \Big|_{(x_i, y_i, t_i)} \approx 0, \quad 1 \leq i \leq N \quad (2.40)$$

các ước lượng chuyển động có thể nhận được bằng cách lấy cực tiểu:

$$Error = \sum_{i=1}^N \left[ v_x \frac{\partial f(x, y, t)}{\partial x} \Big|_{(x_i, y_i, t_i)} + v_y \frac{\partial f(x, y, t)}{\partial y} \Big|_{(x_i, y_i, t_i)} + \frac{\partial f(x, y, t)}{\partial t} \Big|_{(x_i, y_i, t_i)} \right]^2 \quad (2.41)$$

vì biểu thức sai số trong (2.41) là dạng toàn phương của hai số chưa biết  $v_x$  và  $v_y$ , phải giải hai phương trình tuyến tính để có đáp số. Tổng quát hơn, thường giả thiết (2.34) là đúng trong vùng không - thời gian cục bộ ký hiệu là  $\psi$ . Để ước lượng  $v_x$  và  $v_y$  ta lấy cực tiểu:

$$Error = \iiint_{(x,y,t) \in \Psi} \left( v_x \frac{\partial f(x, y, t)}{\partial x} + v_y \frac{\partial f(x, y, t)}{\partial y} + \frac{\partial f(x, y, t)}{\partial t} \right)^2 dx dy dt. \quad (2.42)$$

Các tích phân trong (2.42) có thể được thay thế bằng các tổng. (2.41) là một ví dụ như thế. Lấy vi phân Error trong (2.42) đối với  $v_x$  và  $v_y$  và cho kết quả bằng không dẫn đến:

$$Wv = \gamma \quad (2.43a)$$

trong đó

$$W = \begin{bmatrix} \iiint_{(x,y,t) \in \Psi} \left( \frac{\partial f(x, y, t)}{\partial x} \right)^2 dx dy dt. & \iiint_{(x,y,t) \in \Psi} \frac{\partial f(x, y, t)}{\partial x} \frac{\partial f(x, y, t)}{\partial y} dx dy dt. \\ \iiint_{(x,y,t) \in \Psi} \frac{\partial f(x, y, t)}{\partial x} \frac{\partial f(x, y, t)}{\partial y} dx dy dt. & \iiint_{(x,y,t) \in \Psi} \left( \frac{\partial f(x, y, t)}{\partial y} \right)^2 dx dy dt. \end{bmatrix} \quad (2.43b)$$

$$v = [v_x, v_y]^T \quad (2.43c)$$

$$\gamma = - \begin{bmatrix} \iiint_{(x,y,t) \in \Psi} \frac{\partial f(x, y, t)}{\partial x} \frac{\partial f(x, y, t)}{\partial t} dx dy dt. \\ \iiint_{(x,y,t) \in \Psi} \frac{\partial f(x, y, t)}{\partial y} \frac{\partial f(x, y, t)}{\partial t} dx dy dt. \end{bmatrix} \quad (2.43d)$$

Hai phương trình tuyến tính trong (2.43) có thể có nhiều lời giải. giả sử  $f(x, y, t)$  là hằng số trong vùng không gian thời gian  $\Psi$ , thì  $\partial f(x, y, t)/\partial x$ ,  $\partial f(x, y, t)/\partial y$  và  $\partial f(x, y, t)/\partial t$  tất cả đều bằng không và mọi phần tử trong  $W$  và  $\gamma$  trong (2.43) cũng bằng không, nhờ đó bất kỳ  $(v_x, v_y)$  nào cũng sẽ thoả mãn (2.43a). Mọi tốc độ chuyển động trong vùng độ cường độ đồng đều sẽ không ảnh hưởng đến  $f(x, y, t)$ , do đó không thể từ  $f(x, y, t)$  ước lượng được tốc độ chuyển động thực tế. Giả sử  $f(x, y, t)$  là một mép bậc thang lý tưởng. Tốc độ chuyển động theo hướng song song với mép bậc thang sẽ không ảnh hưởng đến  $f(x, y, t)$  và do đó không thể ước lượng được. Những bài toán này đã được

nghiên cứu và đã khai triển được một nghiêm [Martinez]. Gọi  $\lambda_1$  và  $\lambda_2$  là các giá trị riêng của  $W$  và gọi  $\alpha_1$  và  $\alpha_2$  là các vectơ riêng trực chuẩn tương ứng. Một nghiêm hợp lý của phương trình (2.43) là:

$$\text{Trường hợp 1} \quad v = 0, \quad \lambda_1, \lambda_2 < \text{ngưỡng} \quad (2.44a)$$

$$\text{Trường hợp 2} \quad v = \frac{[\alpha_1^T \gamma] \alpha_1}{\lambda_1}, \quad \lambda_1 \gg \lambda_2 \quad (2.44b)$$

$$\text{Trường hợp 3} \quad v = W^{-1}\gamma, \quad \text{với các giá trị khác (otherwise)} \quad (2.44c)$$

Trường hợp 1 bao gồm cả vùng cùn g độ đều, ở đó đặt tốc độ bằng không.

Trường hợp 2 bao gồm vùng mép biên lý tưởng và ước lượng tốc độ trong (2.44b) đi theo hướng vuông góc với mép biên.

Việc giải phương trình tuyến tính (2.43) yêu cầu định giá  $\partial f(x, y, t)/\partial x$ ,  $\partial f(x, y, t)/\partial y$  và  $\partial f(x, y, t)/\partial t$  ở các vị trí không - thời gian tùy ý. Điều đó có thể được hoàn thành bằng cách mở rộng phương pháp nội suy đa thức không gian ra 3 -D, -- có ưu điểm so với các cách tiếp cận khác là tính toán đơn giản và không nhạy cảm với tạp âm.

Trong phép nội suy đa thức 3-D, hàm nội suy  $\hat{f}(x, y, t)$  là:

$$\hat{f}(x, y, t) = \sum_{i=1}^N S_i \phi_i(x, y, t). \quad (2.45)$$

một ví dụ về  $\phi_i(x, y, t)$  khi  $N = 9$  là.

$$\phi_i(x, y, t) = 1, x, y, t, x^2, y^2, xy, xt, yt. \quad (2.46)$$

Hệ số  $S_i$  có thể được xác định bằng cách lấy cực tiểu:

$$\text{Error} = \sum_{(n_1, n_2, n_3) \in \Psi} \sum \left( f(x, y, t) - \sum_{i=1}^N S_i \phi_i(x, y, t) \right)^2 \Big|_{x=n_1 T_1, y=n_2 T_2, t=n_3 T_3}. \quad (2.47)$$

Một lựa chọn hợp lý là vùng  $\Psi$  chứa đựng 50 pixels: 5 cho  $n_1$ , 5 cho  $n_2$  và 2 cho  $t$ . Để lấy cực tiểu sai số trong (2.47) với  $S_i$  phải giải một hệ phương trình tuyến tính, Lưu ý rằng các đạo hàm riêng  $\partial f(x, y, t)/\partial x$ ,  $\partial f(x, y, t)/\partial y$  và  $\partial f(x, y, t)/\partial t$  dùng trong (2.43) có thể được tính trước thành hàm của các số hạng  $S_i$ .

Các algorit ước lượng chuyển động được thảo luận trên yêu cầu xác định các vùng không - thời gian ký hiệu là  $\Psi$  trên đó có thể giả thiết là chuyển động đều. Vì

vùng không gian cục bộ trong một khung vào cỡ 5 x 5 pixel, để xác định một  $\psi$  hợp lý cần có một ước lượng chuyển động ban đầu chỉ chênh lệch vài pixel so với dịch chuyển thực. Trong thực tế, chuyển động giữa hai khung kề nhau thường không quá 10 pixels. Một cách tiếp cận để nhận được chuyển động (hay tốc độ) ban đầu là dùng tốc độ ở lân cận đã tính trước đó rồi xác định ra vùng  $\psi$  thích hợp như trên Hình 2.45. Một cách tiếp

cận khác là dùng phương pháp cấp bậc (hierarchical) hoặc phương pháp nhiều lưới (multigrid). Phương pháp nhiều lưới bắt đầu với chuyển động được ước lượng trên lưới thô.

Lưới thô nhận được từ khung gốc bằng lọc thông thấp và lấy mẫu down-sampling. Phép lấy mẫu down-sampling làm cho chuyển động (hoặc tốc độ) co lại. Tốc độ

chuyển động lớn trong khung gốc sẽ bị giảm xuống theo một hệ số tỉ lệ bằng hệ số down-sampling. Tốc độ chuyển động trong các khung down-sampling có thể ước

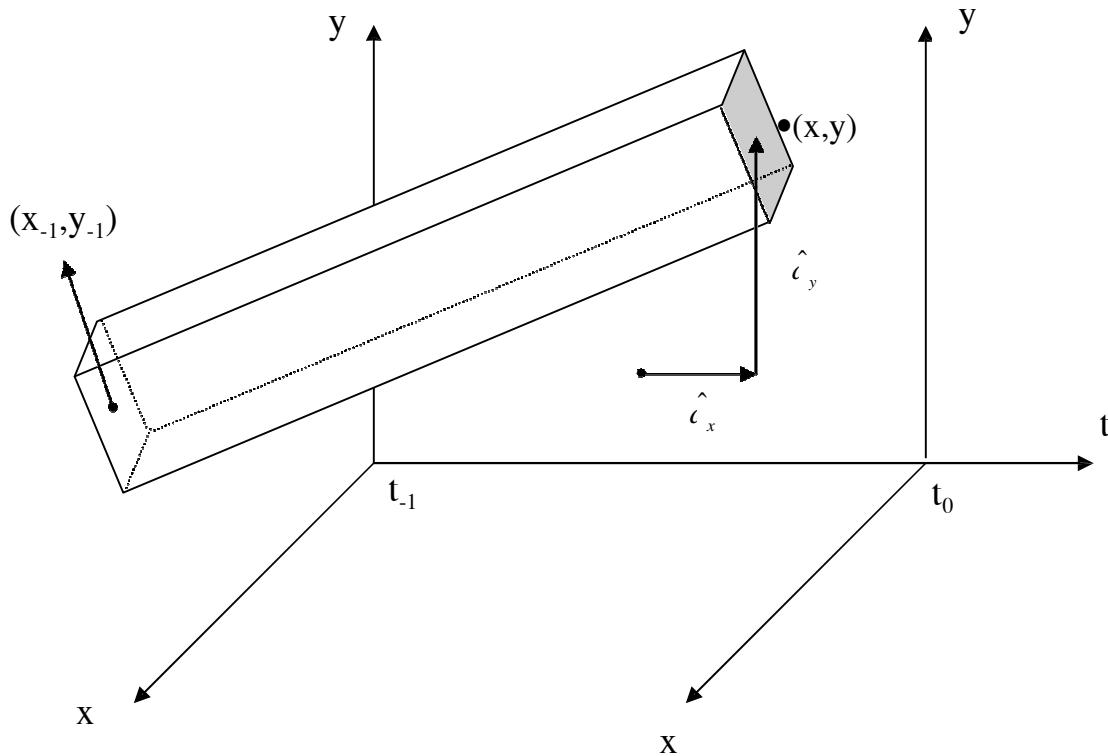
lượng bằng cách sử dụng phương pháp ràng buộc không - thời gian với giả thiết tốc độ ban đầu bằng không. Các tốc độ ước lượng được trên lưới thô sẽ đem nội suy để sinh ra các

tốc độ ban đầu trên một lưới tinh hơn. Có thể sử dụng phép nội suy song tuyến tính để nội suy tốc độ. Phương pháp nhiều lưới có thể xem như một ví dụ về xử lý hình chóp (pyramid processing), khai thác một cấu trúc dữ liệu gọi là hình chóp. Một hình chóp liên tiếp cung cấp những thông tin cô đọng của ảnh. Một ví dụ về hình chóp là ảnh phân giải cao và những ảnh độ phân giải thấp hơn tiếp theo. Xử lý hình chóp có lợi trong các ứng dụng khác nhau bao gồm kỹ thuật mã hoá ảnh sẽ được thảo luận trong chương 4 .

Một ưu điểm chính của các phương pháp ràng buộc không - thời gian so với các phương pháp thích ứng vùng là tính toán đơn giản . Ngoài ra, sự nghiên cứu sơ bộ dựa vào cả dữ liệu tổng hợp và thực tế chỉ ra rằng phương pháp ràng buộc không - thời gian với phép nội suy đa thức cho  $f(x, y, t)$  cho kết quả không kém gì phương pháp thích ứng vùng cả ở các khung ảnh có nhiều và không nhiều.

Cả phương pháp thích ứng vùng và phương pháp ràng buộc không - thời gian, đều có thể đưa ra các sai số đáng kể trong ước lượng chuyển động ở một vài vùng , có thể là do tín hiệu  $f(x, y, t)$  không thể mô hình hoá bằng chuyển động đều hoặc là do algorit ước lượng chuyển động không hoàn hảo khi có nhiều. Một phương tiện để phát hiện những sai số lớn là đem so sánh biểu thức sai số cho các phương pháp thích ứng vùng (2.35), hoặc biểu thức sai số cho các phương pháp ràng buộc không - thời gian (2.42) với một ngưỡng nào đó. Các vùng ở đó sai số vượt trên ngưỡng được tuyên bố là

vùng ước lượng chuyển động không đáng tin và phép xử lý bù chuyển động ở các vùng này phải huỷ bỏ.



**Hình 2.45:** Vùng  $\psi$  dùng trong (2.42) để ước lượng  $(\hat{d}_x, \hat{d}_y)$  tại vị trí không gian  $(x_0, y_0)$  và thời điểm  $t_0$ . Giá trị ước lượng  $(\hat{d}_x, \hat{d}_y)$  đại diện cho sự dịch chuyển tính ra trước đây ở vị trí lân cận

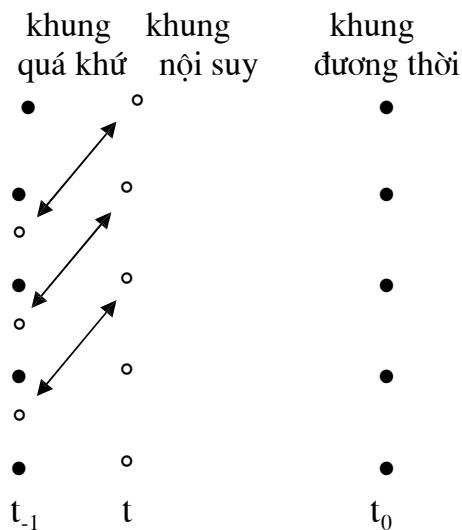
#### 4.3. PHÉP NỘI SUY THỜI GIAN CÓ BÙ CHUYỂN ĐỘNG

Giả sử ta có hai khung liên tiếp  $f(n_1, n_2, t_{-1})$  và  $f(n_1, n_2, t_0)$  như trong Hình 2.46. Ta muốn tạo ra một khung mới  $f(n_1, n_2, t)$  trong đó  $t_{-1} < t < t_0$ . Một cách tiếp cận đơn giản là chọn một khung gốc gần kề về thời gian với khung mong muốn. Một vấn đề với cách tiếp cận này là, nếu dãy khung có một chuyển động tổng lớn thì sẽ có hiện tượng giật giật.

Một cách khác là sử dụng phép nội suy bù chuyển động, các algorit ước lượng đã thảo luận ở tiết trước. Trong phép nội suy thời gian có bù chuyển động, ta giả thiết

là chuyển động tịnh tiến đều trên một vùng không - thời gian .Từ  $f(n_1, n_2, t_{-1})$  và  $f(n_1, n_2, t_0)$  ta tính ra các tốc độ ở  $f(n_1, n_2, t)$ . Sau đó ta chiếu các tốc độ đến khung ở  $t_{-1}$  hoặc  $t_0$  về mặt thời gian gần với thời điểm mong muốn  $t$ , như trong Hình 2.46 .Vì điểm chiếu không gian thường không nằm trên lưới lấy mẫu gốc, nên cần phải nội suy không gian để nhận được khung nội suy. Nếu tốc độ được ước lượng tại một pixel đặc biệt trong  $f(n_1, n_2, t_{-1})$  không được coi là đủ độ chính xác, thì giả thiết là tốc độ bằng 0. Trong trường hợp này, giá trị pixel được nội suy có trị số giống như ở pixel cùng vị trí trong  $f(n_1, n_2, t_{-1})$  hoặc  $f(n_1, n_2, t_0)$  tùy theo khung nào về thời gian gần với thời điểm mong muốn  $t$  hơn.

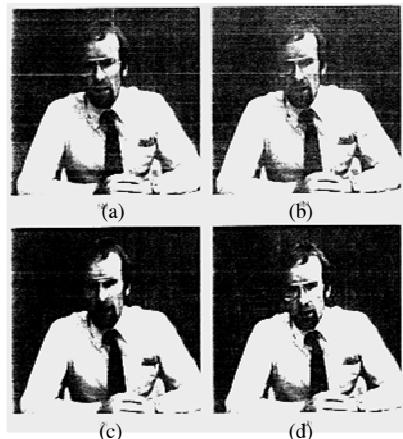
Không thể dùng ảnh tĩnh để minh họa các đặc trưng chuyển động của phép nội suy khung có bù chuyển động.Tuy nhiên chúng ta vẫn có thể có một khái niệm thông bằng cách xem một khung tĩnh được tạo ra từ 2 khung ảnh bằng phương pháp này.Hình 2.47 trình bày một bộ 4 khung: 2 khung gốc trên Hình 2.47(a), 1.47(d) và 2 khung được nội suy trên Hình 2.47(b) và 1.47(c). Khung được nội suy trong (b) là có bù chuyển động. Khung trong (c) nhận được bằng cách lấy trung bình của 2 khung gốc. Khung này cho thấy lượng chuyển động xuất hiện giữa 2 khung gốc. 4 khung có độ phân giải không gian là 512 x 512 pixel. Khung được nội suy ứng với thời điểm ở chính giữa 2 khung gốc. Chú ý rằng trong ví dụ này khi sử dụng bù chuyển động thì chất lượng của những ảnh nội suy về cơ bản giống như 2 khung gốc. Phương pháp ước lượng chuyển động dùng ở đây là phương pháp ràng buộc không - thời gian với phép nội suy đa thức đã thảo luận trong phần 2.4.2.



**Hình 2.46:** Tạo ra  $f(n_1, n_2, t)$  bằng cách nội suy  $f(n_1, n_2, t_{-1})$  và  $f(n_1, n_2, t_0)$ . Trong ví dụ này dịch chuyển ( $d_x, d_y$ ) nhận được từ  $f(n_1, n_2, t_{-1})$  và  $f(n_1, n_2, t_0)$  ở từng pixel ( $n_1, n_2$ ) tại thời điểm  $t$ . Mỗi pixel tại thời điểm  $t$  được chiếu vào vị trí không gian tương ứng tại thời điểm  $t_{-1}$  (trong ví dụ này  $t$  gần  $t_{-1}$  hơn  $t_0$ ) và cường độ pixel xác định theo  $f(n_1, n_2, t_{-1})$  tại vị trí pixel được chiếu đến. Lưu ý rằng để thực thi thuật toán này thường cần đến nội suy không gian của  $f(n_1, n_2, t_{-1})$ .

**Hình 2.47:** Ví dụ về nội suy khung theo thời gian

- (a) Khung gốc 1;
- (b) Khung nội suy bằng bù chuyển động;
- (c) Khung nội suy bằng lấy trung bình giữa hai khung;
- (d) Khung gốc 2.



Phép nội suy có bù chuyển động đã được sử dụng trong việc thay đổi nhịp khung (frame rate). Sự thay đổi nhịp khung có thể phối hợp với sự thay đổi thang thời gian của âm thanh [Lim] để thay đổi độ dài của ảnh động hoặc của chương trình TV. Kinh nghiệm với cảnh điển hình (typical scene) cho thấy rằng sự thay đổi nhịp khung của video qua phép nội suy bù chuyển động có thể tạo ra tín hiệu video với chất lượng so sánh được với tín hiệu gốc, ngoại trừ những nhịp chuyển động không tự nhiên đối với một số động tác như đi bộ và nói, - xuất hiện khi hệ số thay đổi nhịp đủ cao [Martinez].

#### 4.4. ÚNG DỤNG CỦA CÁC PHƯƠNG PHÁP ƯỚC LƯỢNG CHUYỂN ĐỘNG VÀO PHÉP NỘI SUY KHÔNG GIAN

Ý tưởng chung về nội suy bù chuyển động có thể được sử dụng để phát triển các algorit mới cho nội suy không gian. Để thử các algorit mới, ta xét một bài toán nội suy không gian cụ thể. Như đã thảo luận trong tiết 1.4 hệ thống truyền hình NTSC sử dụng format 2:1 ken dòng, nhịp quét 30 khung /giây. Một khung bao gồm 525 dòng quét ngang và được chia thành 2 trường, trường lẻ bao gồm số dòng lẻ và trường chẵn chứa các dòng chẵn .Tạo ra một khung tại thời điểm t từ một trường tại thời điểm t thông qua phép nội suy không gian có thể có ích trong một số các ứng dụng, bao gồm truyền hình 60 khung/giây mà không tăng độ rộng băng tần và cải thiện độ phân giải theo phương thẳng đứng của các khung cố định.

Kỹ thuật nội suy không gian đã thảo luận trong chương 1.4.1 có thể được sử dụng để tạo ra 1 khung từ 1 trường, nhưng gộp thêm một vài hiểu biết về ảnh có thể cải thiện hiệu năng của các algorit nội suy không gian. Nhiều phần tử trong thế giới thị giác như các đường biên và vết xước có tính liên tục trong không gian. Thông tin này có thể được khai thác để tạo ra 1 khung từ 1 trường, Gọi  $f(x, y_{-1})$  và  $f(x, y_0)$  là cường độ ảnh của 2 dòng quét ngang kề nhau trong một trường. Ta muốn tạo ra 1 dòng mới quét giữa  $f(x, y_{-1})$  và  $f(x, y_0)$ . Một mô hình có xét đến tính liên tục không gian của các phần tử như các đường biên và vết xước là:

$$f(x, y_0) = f(x - d_x, y_{-1}) \quad (2.48)$$

trong đó  $d_x$  là chuyển động theo phương ngang giữa  $y_{-1}$  và  $y_0$ . Phương trình (2.48) có thể được xem như là một trường hợp đặc biệt trong mô hình tịnh tiến với tốc độ đều của công thức (2.29). Biến không gian  $y$  trong (2.48) có chức năng tương tự với biến thời gian  $t$  trong (2.29), và chỉ có một biến không gian  $x$  trong (2.48) trong khi có đến 2 biến không gian  $x, y$  trong (2.29). Như vậy, mọi điều chúng ta đã thảo luận trong tiết 1.4.2 đều áp dụng được cho bài toán ước lượng  $d_x$  trong (2.48). Ví dụ, giả thiết rằng chuyển động có tốc độ đều, thì (2.48) có thể được biểu diễn như sau:

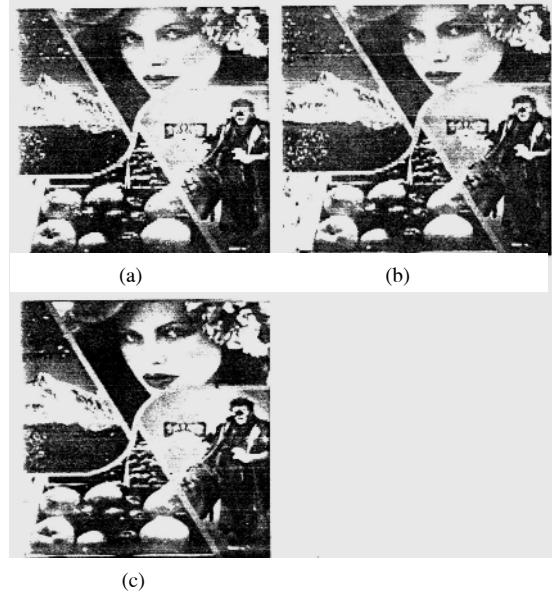
$$f(x, y) = f(x - v_x(y - y_{-1}), y_{-1}), y_{-1} \leq y \leq y_0 \quad (2.49)$$

Điều kiện ban đầu là

$$\nu_x \frac{\partial f(x, y)}{\partial x} + \frac{\partial f(x, y)}{\partial y} = 0 \quad (2.50)$$

**Hình 2.48:** Tạo ra một khung từ một trường bằng phép nội suy không gian.

- (a) Ảnh 256 x 256 pixel nhận được từ một ảnh 256 x 128 pixel bằng nội suy bậc -không
- (b) Cũng như (a) nhưng dùng nội suy song tuyến tính
- (c) Cũng như (a) nhưng áp dụng algorit ước lượng chuyển động.



Phương trình (2.48) có thể dùng để phát triển các phương pháp thích ứng vùng và (2.50) có thể dùng để phát triển các phương pháp ràng buộc không - thời gian cho việc ước lượng  $d_x$  hoặc  $\nu_x$ .

Một khi chuyển động theo phương ngang (hoặc tốc độ) đã được ước lượng, thì có thể đem vào sử dụng trong nội suy không gian theo kiểu tương tự như phép nội suy thời gian đã thảo luận trong tiết 2.4.3.

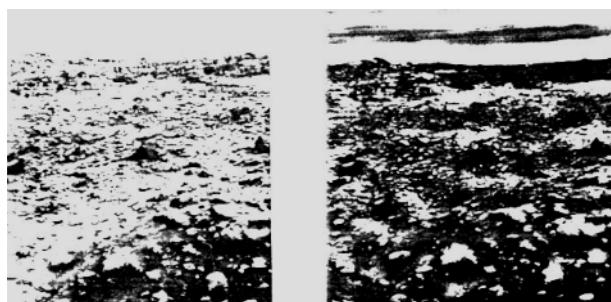
Hình 2.48 minh họa hiệu năng của algorit nội suy không gian dựa tr ên (2.50). Hình 2.48(a) biều diẽn khung 256 x 256 pixels nhận được bằng sự lặp lại từng dòng theo phương ngang ảnh 256 x 128 pixels. Hình 2.48(b) là khung nhận được bằng phép nội suy không gian song tuyến tính,Hình 2.48(c) là khung nhận được bằng ước lượng chuyển động theo phương ngang dựa trên (2.50) và tiếp đó dùng ước lượng cho phép nội suy không gian. Ảnh trong Hình 2.48(c) bảo tồn tính liên tục không gian của các đường cong và các vòng đường biên tốt hơn 2 ảnh trong Hình 2.48(a) và (b).

## 5. GIẢ MÀU VÀ MÀU TÔ

Ta biết rằng thị giác con người rất nhạy cảm với màu. Chẳng hạn, số lượng cường độ có thể phân biệt nhỏ hơn nhiều so với số lượng màu có thể phân biệt và cường độ. Ngoài ra ảnh màu đẹp mắt người xem hơn ảnh trắng đen. Tính thẩm mỹ của màu có thể sử dụng cho cải thiện ảnh. Trong một vài ứng dụng như trên truyền hình thương mại, có thể sử dụng giả màu làm nổi bật một đối tượng đặc biệt trên ảnh. Ví dụ, một quả chuối màu đỏ đặt giữa những hoa quả khác có màu tự nhiên sẽ nhận được sự chú ý của con người nhiều hơn. Trong những ứng dụng khác, những dữ liệu không đại biểu cho một bức ảnh vẫn có thể biểu diễn dưới dạng ảnh màu. Trong trường hợp này, màu được dùng gọi là màu tô. Ví dụ, phổ tiếng nói biểu diễn năng lượng tiếng nói theo hàm thời gian và tần số có thể biểu dưới dạng ảnh màu, với những đoạn lặng, đoạn thoại, đoạn phi thoại được phân biệt bằng các màu khác nhau và năng lượng mạnh yếu thể hiện bằng cường độ màu.

Việc dùng màu để cải thiện ảnh là theo trí tưởng tượng nghệ thuật, tuyệt đối không có nguyên tắc chỉ đạo hoặc luật lệ nào phải theo cả. Do đó trong tiết này ta tập trung vào ba ví dụ để minh họa các loại hình dùng màu cải thiện ảnh. Trong ví dụ đầu tiên ta biến đổi ảnh trắng -đen thành ảnh màu bằng một quy trình rất đơn giản. Để nhận được ảnh màu từ một ảnh trắng -đen, trước hết cho ảnh trắng đen qua bộ lọc thông thấp, bộ lọc thông dải và bộ lọc thông cao. Kết quả nhận được sau khi qua bộ lọc thông thấp được coi như là thành phần màu xanh lam. Ảnh đã qua bộ lọc thông dải coi như là thành phần màu xanh lá cây, còn ảnh đã qua bộ lọc thông cao coi như là thành phần màu đỏ.

Ba thành phần đỏ-xanh lá cây-xanh lam được kết hợp thành ảnh màu. Hình 2.49(a) là ảnh nguồn trắng đen 512x512 pixels, Hình 2.49(b) là ảnh màu nhận được bằng quy trình này(xem phụ bản màu). Màu rất dễ coi, nhưng quy trình này không thể tạo ra ảnh màu giống với tự nhiên được. Đem phim trắng -đen chuyển thành phim màu như là Casablanca hoặc It's a wonderful life cần đến những phép xử lý tinh xảo hơn và nhiều pha can thiệp của con người.

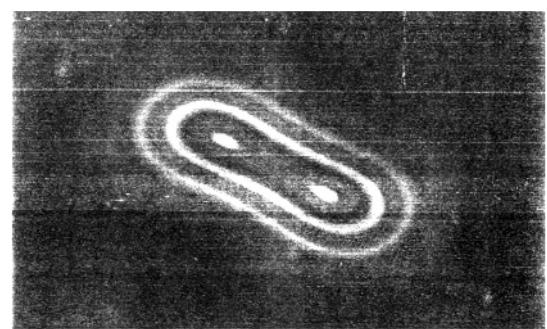
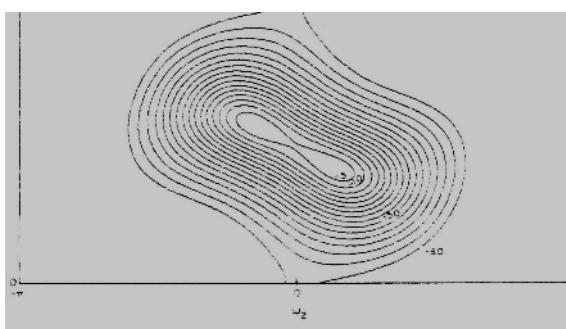


(a) (b)

**Hình 2.49:** Tạo ra ảnh mẫu từ một ảnh trắng đen bằng cách ánh xạ kết quả lọc thông thấp ảnh trắng đen vào thành phần màu xanh, ánh xạ kết quả lọc thông dải ảnh trắng đen vào thành phần màu xanh lá cây, và ánh xạ kết quả lọc thông thấp ảnh trắng đen vào thành phần màu đỏ.

Trong ví dụ thứ hai, ta xét sự hiển thị ước lượng  $P_x(\omega_1, \omega_2)$  của phổ không gian 2-D trên màn hình ống tia điện tử (CRT).  $P_x(\omega_1, \omega_2)$ , - tính bằng dB, thường được hiển thị bằng những đường bao. Một ví dụ ước lượng phổ hợp lẽ tối đa cho dữ liệu của hai tín hiệu hình sin trong tạp âm trắng được biểu diễn trên Hình 2.50(a). Cực đại ứng với 0 dB, còn các đường bao là những mức giảm dần từng 0.5dB so với điểm cực đại. Như đã thảo luận trước đây, trong những ứng dụng như phát hiện máy bay ở tầm thấp bằng một dãy cảm biến microphone, ta muốn xác định số lượng tín hiệu hình sin hiện diện và tần số của chúng. Một cách khác để biểu diễn ước lượng phổ là tô màu. Hình 2.50(b) cho ví dụ, ở đó các biên độ khác nhau của  $P_x(\omega_1, \omega_2)$  được ánh xạ vào các màu khác nhau. So sánh hai hình thấy rằng ở Hình 2.50(b) hai đỉnh và vị trí của chúng trong ước lượng phổ nổi bật hơn.

Ví dụ thứ ba là sử dụng màu để hiển thị cự ly. Trong những ứng dụng như radar hồng ngoại, có thông tin về cự ly và và cường độ sáng. Hình 2.51(a) (xem phụ bản màu) là cường độ ảnh của nhiều tòa nhà ở cách trạm radar từ 2 tới 4 km, thông tin về cự ly bị che lấp. Hình 2.51(b) là ảnh sử dụng màu để hiển thị cự ly. Giá trị cự ly xác định màu sắc (hue), còn cường độ quyết định mức sáng của màu. Tính ưu việt của kỹ thuật được chứng minh khi ta nhìn thấy trong Hình 2.51(b) một đường nằm ngang ở cự ly gần (ảnh của một đường dây điện thoại), còn trong Hình 2.51(a) thì hoàn toàn không nhìn thấy được.



**Hình 2.50:** Hiển thị đánh giá phổ bằng tô màu.

- (a) Đánh giá hợp lý tối đa phổ 2-D được biểu diễn bằng bản đồ đường bao
- (b) Đánh giá phổ ở hình (b) được biểu diễn bằng màu (xem ảnh màu kèm theo).



(a)



(b)

**Hình 151:** Hiển thị thông tin về cự ly bằng màu sắc

- (a) Cường độ ảnh trên một ảnh radar hồng ngoại với thông tin về cự ly bị loại bỏ;
- (b) Ảnh trong hình (a) với thông tin về cự ly được hiển thị bằng màu sắc.

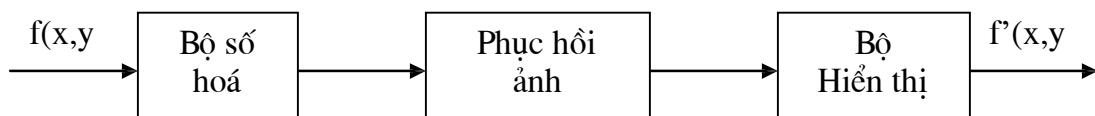
### CHƯƠNG 3

#### PHỤC HỒI ẢNH.

##### v GIỚI THIỆU

Trong phục hồi ảnh, ảnh bị xuống cấp một cách nào đó và mục đích phục hồi là làm giảm bớt hoặc loại bỏ sự xuống cấp. Các algorit cải thiện ảnh đơn giản và mang tính kinh nghiệm (heuristic) để làm giảm sự xuống cấp đã được thảo luận trong chương 2. Trong chương này, ta nghiên cứu các algorit phục hồi ảnh. Các algorit phục hồi ảnh thường tính toán phức tạp hơn algorit cải thiện ảnh. Ngoài ra, chúng được thiết kế để khai thác các đặc tính chi tiết của tín hiệu và sự xuống cấp.

Một môi trường điển hình cho hệ phục hồi ảnh được biểu diễn trên hình 3.1. Nếu bộ số hoá (digitizer) và bộ hiển thị (display) là lý tưởng thì cường độ ảnh đầu ra  $f'(x,y)$  sẽ đồng nhất cường độ đầu vào  $f(x,y)$ , không phải phục hồi tí nào. Trong thực tiễn, có nhiều loại xuống cấp khác nhau có thể xảy ra trong bộ số hoá và bộ hiển thị. Với hệ phục hồi ảnh ta giải quyết sự xuống cấp để làm cho ảnh đầu ra  $f'(x,y)$  gần giống như ảnh đầu vào  $f(x,y)$ .



**Hình 3.1:** Môi trường điển hình cho phục hồi ảnh.

Để nghiên cứu phục hồi ảnh, ta giả thiết rằng tất cả sự xuống cấp đều xảy ra trước khi áp dụng hệ phục hồi ảnh, như trên hình 3.2. Điều này cho phép ta xét toàn bộ vấn đề phục hồi ảnh trong miền không gian rời rạc (đường chấm trong hình 3.2). Ta có thể coi  $f(n_1, n_2)$  là ảnh số gốc,  $g(n_1, n_2)$  là ảnh số bị giảm chất lượng và  $p(n_1, n_2)$  là ảnh số đã xử lý. Mục đích của phục hồi ảnh là làm cho ảnh đã xử lý  $p(n_1, n_2)$  gần giống như

ảnh ban đầu  $f(n_1, n_2)$ . Không phải giả thiết cho rằng “tất cả sự xuống cấp đều xảy ra trước khi áp dụng hệ phục hồi ảnh” bao giờ cũng hợp lý. Một ví dụ là sự xuống cấp do nhiễu cộng ngẫu nhiên trong bộ hiển thị. Trong trường hợp này, nên xử lý ảnh trước để đề phòng sự xuống cấp về sau. Tuy nhiên, với nhiều loại i xuống cấp khác nhau, như nhoè trong bộ số hoá và bộ hiển thị, có thể lập mô hình là xảy ra trước khi áp dụng hệ phục hồi ảnh. Trong chương này, ta giả sử rằng ảnh gốc  $f(n_1, n_2)$  bị xuống cấp, và được đưa vào hệ phục hồi để từ ảnh đã xuống cấp  $g(n_1, n_2)$  phục hồi lại ảnh  $f(n_1, n_2)$  như ta thấy trên hình 3.2.

Sự lựa chọn hệ phục hồi ảnh phụ thuộc vào loại hình xuống cấp. Các algorit làm giảm nhiễu cộng ngẫu nhiên khác với các algorit làm giảm nhoè ảnh. Các loại hình xuống cấp ta xét trong chương này là nhiễu cộng ngẫu nhiên, nhoè và nhiễu phụ thuộc tín hiệu, như nhiễu nhân. Chọn những loại hình xuống cấp này là vì chúng thường xảy ra trong thực tiễn và được đề cập đến trong nhiều tài liệu. Ngoài việc trình bày về các hệ phục hồi ảnh chuyên trị những loại hình xuống cấp nói đến trong chương này, còn đề cập đến các cách tiếp cận chung dùng cho việc khai triển các hệ làm giảm các loại xuống cấp khác.

Xuyên qua toàn chương đưa ra nhiều ví dụ minh họa hiệu năng của các algorit khác nhau. Các ví dụ chỉ có tính chất minh họa chứ không thể dùng để so sánh hiệu năng của các algorit khác nhau. Hiệu năng của algorit xử lý ảnh phụ thuộc vào nhiều yếu tố, như mục tiêu xử lý và loại ảnh cụ thể. Một hoặc hai ví dụ không đủ chứng minh hiệu năng của algorit.

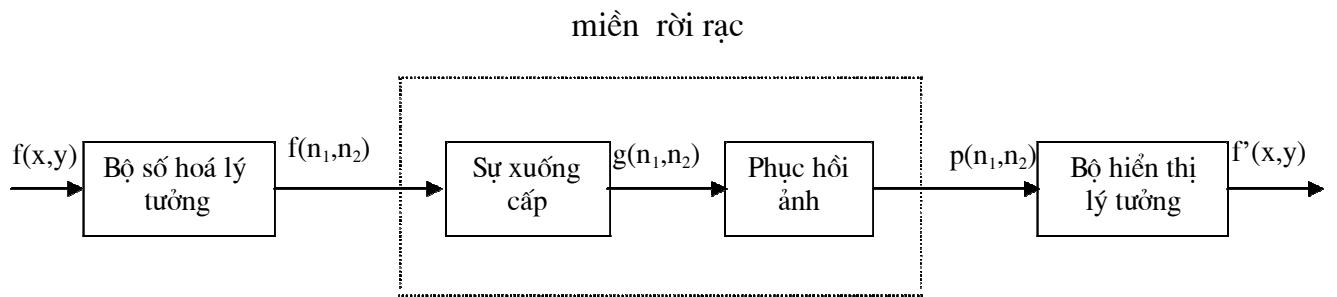
Trong tiết 3.1, ta thảo luận cách lấy thông tin về sự xuống cấp. Sự hiểu biết chính xác bản chất của sự xuống cấp rất quan trọng trong việc phát triển thành công các algorit phục hồi ảnh. Trong tiết 3.2, ta thảo luận vấn đề phục hồi ảnh bị xuống cấp bởi nhiễu cộng ngẫu nhiên. Tiết 3.3 bàn về phục hồi ảnh bị xuống cấp bởi nhoè. Tiết 3.4, bàn về phục hồi ảnh bị xuống cấp bởi cả nhoè và nhiễu cộng ngẫu nhiên, và về vấn đề chung hơn là làm giảm xuống cấp cho ảnh bị nhiều loại hình xuống cấp cùng tác động. Trong tiết 3.5 ta khai triển các algorit phục hồi dùng làm giảm nhiễu phụ thuộc tín hiệu. Tiết 3.6, bàn về xử lý trong miền thời gian để phục hồi ảnh. Trong tiết 3.7, ta miêu tả cách đặt bài toán phục hồi ảnh bằng kí hiệu ma trận và cách dùng các công cụ của đại số học tuyến tính để giải những bài toán phục hồi ảnh.

## 1. ƯỚC LƯỢNG SỰ XUỐNG CẤP

Vì các algorit phục hồi ảnh được thiết kế để khai thác các đặc tính của tín hiệu và sự xuống cấp, nên sự hiểu biết tường tận bản chất của sự xuống cấp là rất quan trọng để khai triển thành công algorit phục hồi ảnh. Có hai cách tiếp cận để có thông tin về sự xuống cấp. Một cách tiếp cận là thu thập thông tin từ chính ảnh bị xuống cấp. Nếu ta có thể tìm ra các vùng cường độ xấp xỉ đồng đều trong ảnh, chẳng hạn bầu trời, thì có thể ước lượng phổ công suất hoặc hàm mật độ xác suất của nhiều nền ngẫu nhiên từ sự thăng giáng cường độ trong các vùng có nền đồng đều. Một ví dụ khác như, khi ảnh bị nhòe nếu ta tìm được trong ảnh đã xuống cấp một vùng mà tín hiệu gốc đã biết, thì có thể ước lượng hàm nhòe  $b(n_1, n_2)$ . Ký hiệu tín hiệu ảnh gốc ở một vùng đặc biệt của ảnh là  $f(n_1, n_2)$  và ảnh bị xuống cấp trong vùng đó là  $g(n_1, n_2)$ , thì quan hệ gần đúng giữa  $g(n_1, n_2)$  và  $f(n_1, n_2)$  là

$$g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2) \quad (3.1)$$

Theo giả thiết  $f(n_1, n_2)$  và  $g(n_1, n_2)$  đều đã biết, nên có thể được ước lượng được  $b(n_1, n_2)$  từ (3.1). Nếu  $f(n_1, n_2)$  là đáp ứng xung  $\delta(n_1, n_2)$  thì  $g(n_1, n_2) = b(n_1, n_2)$ . Một ví dụ của trường hợp này là ảnh một ngôi sao trong bầu trời đêm.



**Hình 3.2:** Phục hồi ảnh dựa trên giả thiết rằng tất cả sự xuống cấp đều xảy ra trước khi áp dụng phục hồi ảnh. Điều này cho phép ta xét vấn đề phục hồi ảnh trong miền không gian rời rạc.

Một cách tiếp cận khác để hiểu biết về sự xuống cấp là nghiên cứu cơ chế gây ra xuống cấp. Ví dụ, xét một ảnh tương tự (analog)  $f(x, y)$  bị nhòe bởi sự dịch chuyển phẳng của máy ảnh lúc chớp. Giả thiết không có sự xuống cấp nào khác ngoại trừ nhòe vì máy ảnh chuyển động, ta có thể biểu diễn ảnh bị xuống cấp  $g(x, y)$  là:

$$g(x, y) = \frac{1}{T} \int_{t=-T/2}^{T/2} f(x - x_0(t), y - y_0(t)) dt \quad (3.2)$$

trong đó  $x_0(t)$  và  $y_0(t)$  theo thứ tự đại biểu cho sự tịnh tiến theo phương ngang và dọc của  $f(x, y)$  ở thời điểm  $t$  và  $T$  là thời gian chớp. Trong miền biến đổi Fourier, (3.2) có thể biểu diễn là:

$$\begin{aligned} G(\Omega_x, \Omega_y) &= \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} g(x, y) \exp(-j\Omega_x x) \exp(-j\Omega_y y) dx dy \\ &= \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} \left[ \frac{1}{T} \int_{t=-T/2}^{T/2} f(x - x_0(t), y - y_0(t)) dt \right] \exp(-j\Omega_x x) \exp(-j\Omega_y y) dx dy \end{aligned} \quad (3.3)$$

trong đó  $G(\Omega_x, \Omega_y)$  là hàm biến đổi Fourier của  $g(x, y)$ . Ước lược (3.3) ta nhận được

$$G(\Omega_x, \Omega_y) = F(\Omega_x, \Omega_y) B(\Omega_x, \Omega_y) \quad (3.4a)$$

trong đó  $B(\Omega_x, \Omega_y) = \frac{1}{T} \int_{t=-T/2}^{T/2} e^{-j\Omega_x x_o(t)} e^{-j\Omega_y y_o(t)} dt.$  (3.4b)

Từ (3.4), thấy rằng nhoè vì chuyển động có thể được xem như một phép nhân chập  $f(x, y)$  với  $b(x, y)$ , mà biến đổi Fourier là  $B(\Omega_x, \Omega_y)$  tính theo công thức (3.4b). Đôi khi gọi hàm  $b(x, y)$  là hàm nhoè, vì  $b(x, y)$  thường có đặc tính thông thấp và làm nhoè ảnh. Cũng có thể gọi nó là *hàm trải rộng điểm* vì nó trải rộng xung. Khi không có chuyển động  $x_0(t) = 0$  và  $y_0(t) = 0$ ,  $B(\Omega_x, \Omega_y) = 1$  và  $g(x, y)$  là  $f(x, y)$ . Nếu có chuyển động tuyến tính theo hướng  $x$  để  $x_0(t) = kt$  và  $y_0(t) = 0$ ,  $B(\Omega_x, \Omega_y)$  trong công thức (3.4) rút gọn lại.

$$B(\Omega_x, \Omega_y) = \frac{\sin \frac{\Omega_x}{2} kT}{\frac{\Omega_x}{2} kT} \quad (3.5)$$

Mô hình gần đúng của ảnh rời rạc  $g(n_1, n_2)$  là

$$g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2) \quad (3.6)$$

trong đó  $B(\omega_1, \omega_2)$  là hàm biến đổi Fourier trong không gian rời rạc của  $b(n_1, n_2)$ , là một dạng của  $B(\Omega_x, \Omega_y)$  trong (3.4b). Một ví dụ khác ở đó sự xù ống cấp có thể được ước

lượng tử cơ chế của nó là nhiều hạt của phim, làm nhòe ảnh là do nhiều xạ quang và gây ra nhiều lốm đốm.

## 2. LÀM GIẢM NHIỀU CỘNG NGẦU NHIÊN

Mô hình ảnh bị xuống cấp bởi nhiều cộng ngẫu nhiên như sau

$$g(n_1, n_2) = f(n_1, n_2) + v(n_1, n_2) \quad (3.7)$$

trong đó  $v(n_1, n_2)$  biểu diễn nhiều cộng ngẫu nhiên độc lập với tín hiệu. Ví dụ về sự xuống cấp do nhiều cộng ngẫu nhiên bao gồm nhiều ở mạch điện tử và nhiều lượng tử hóa biến độ. Trong tiết này ta thảo luận về một số algorit làm giảm nhiều cộng ngẫu nhiên trong ảnh.

### 2.1. BỘ LỌC WIENER

Một trong những phương pháp đầu tiên được triển khai để làm giảm nhiều cộng ngẫu nhiên trong ảnh là phép lọc Wiener. Nếu ta giả thiết rằng  $f(n_1, n_2)$  và  $v(n_1, n_2)$  là những mẫu độc lập tuyến tính của quá trình ngẫu nhiên dừng trung vị bằng không, và phổ công suất  $P_f(\omega_1, \omega_2)$  và  $P_v(\omega_1, \omega_2)$  của chúng đã biết, thì có thể nhận được ước lượng tuyến tính tối ưu sai số quân phương tối thiểu của  $f(n_1, n_2)$  bằng cách cho  $g(n_1, n_2)$  qua bộ lọc Wiener mà đáp ứng tần số như sau.

$$H(\omega_1, \omega_2) = \frac{P_f(\omega_1, \omega_2)}{P_f(\omega_1, \omega_2) + P_v(\omega_1, \omega_2)} \quad (3.8)$$

Nếu ta thêm điều kiện ràng buộc rằng  $f(n_1, n_2)$  và  $v(n_1, n_2)$  là những mẫu của quá trình ngẫu nhiên Gauss thì bộ lọc Wiener trong công thức (3.8) là bộ ước lượng (estimator) tuyến tính tối ưu sai số quân phương tối thiểu của tín hiệu trong những bộ ước lượng tuyến tính và phi tuyến. Bộ lọc Wiener được dùng để phục hồi ảnh lần đầu tiên vào đầu thập kỷ 60. Nó cũng ảnh hưởng đến sự phát triển nhiều hệ phục hồi ảnh khác.

Bộ lọc Wiener trong (3.8) được thiết lập với giả thiết rằng  $f(n_1, n_2)$  và  $v(n_1, n_2)$  là mẫu của những quá trình trung vị bằng không. Nếu  $f(n_1, n_2)$  có giá trị trung vị là  $m_f$  và  $v(n_1, n_2)$  có giá trị trung vị là  $m_v$  thì thoát tiên đem ảnh bị xuống cấp  $g(n_1, n_2)$  trừ đi  $m_f$  và  $m_v$ . Sau đó cho kết quả  $g(n_1, n_2) - (m_f + m_v)$  qua bộ lọc Wiener. Đầu ra bộ lọc được cộng với giá trị trung bình  $m_f$  của tín hiệu. Điều này được biểu diễn trên hình 3.3. Việc xử lý những giá trị trung vị khác không như trên hình 3.3 làm giảm đến tối thiểu sai số quân phương giữa  $f(n_1, n_2)$  và  $p(n_1, n_2)$  đối với các quá trình ngẫu nhiên Gauss  $f(n_1, n_2)$ .

và  $v(n_1, n_2)$ . Nó cũng đảm bảo rằng  $p(n_1, n_2)$  sẽ là một ước lượng không thiên (unbiased) của  $f(n_1, n_2)$ . Nếu  $m_v = 0$  thì  $m_f$  đồng nhất với giá trị trung vị của  $g(n_1, n_2)$ . Trong trường hợp này, có thể từ  $g(n_1, n_2)$  ước lượng được  $m_f$ .

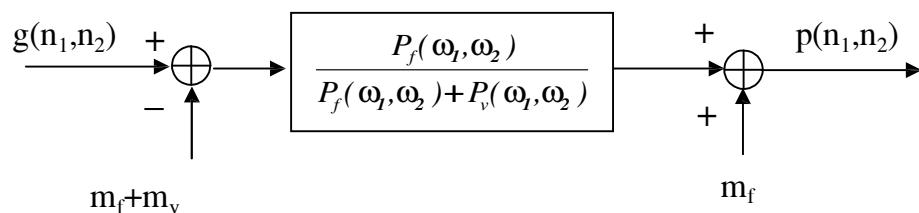
Bộ lọc Wiener trong (3.8) là lọc pha-không. Vì các phổ công suất  $P_f(\omega_1, \omega_2)$  và  $P_v(\omega_1, \omega_2)$  là thực và không âm nên  $H(\omega_1, \omega_2)$  cũng là thực không âm, nhờ đó bộ lọc Wiener chỉ ảnh hưởng tới biên độ phổ nhưng không ảnh hưởng pha. Bộ lọc Wiener giữ nguyên SNR(tỉ số tín hiệu trên nhiễu) của các phân hợp thành tần số cao nhưng làm giảm SNR của các phân hợp thành tần số thấp. Nếu ta cho  $P_f(\omega_1, \omega_2)$  tiến dần tới 0 thì  $H(\omega_1, \omega_2)$  sẽ tiến dần tới 1, cho thấy là bộ lọc có khuynh hướng giữ nguyên SNR của các phân hợp thành tần số cao. Nếu ta cho  $P_v(\omega_1, \omega_2)$  tiến dần tới  $\infty$ ,  $H(\omega_1, \omega_2)$  sẽ tiến dần tới 0, cho thấy là bộ lọc có khuynh hướng làm giảm SNR của các phân hợp thành tần số thấp.

Bộ lọc Wiener dựa vào giả thiết là phổ công suất  $P_f(\omega_1, \omega_2)$  và  $P_v(\omega_1, \omega_2)$  đã biết hoặc có thể ước lượng được. Trong những bài toán thường gặp, ước lượng phổ công suất nhiễu  $P_v(\omega_1, \omega_2)$  bằng các phương pháp đã thảo luận tương đối dễ làm, nhưng ước lượng phổ công suất ảnh  $P_f(\omega_1, \omega_2)$  thì không đơn giản. Một phương pháp được sử dụng là lấy trung bình  $|F(\omega_1, \omega_2)|^2$  cho nhiều ảnh  $f(n_1, n_2)$  khác nhau. Điều này tương tự phương pháp lấy trung bình chu kỳ đồ (periodogram averaging) để ước lượng phổ. Một phương pháp khác là mô hình hóa  $P_f(\omega_1, \omega_2)$  bằng một hàm đơn giản như

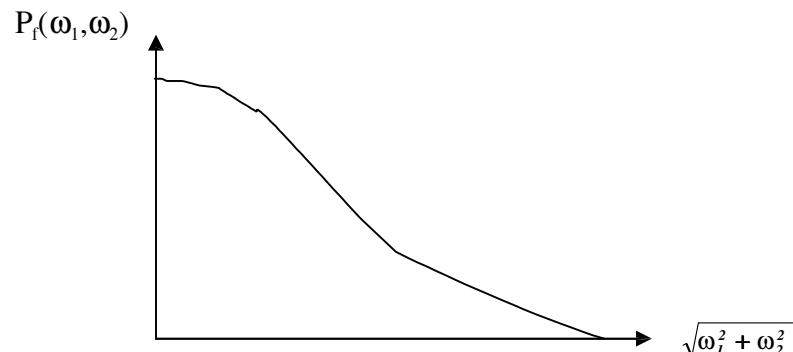
$$R_f(n_1, n_2) = \rho^{\sqrt{n_1^2 + n_2^2}} \quad (3.9a)$$

$$P_f(\omega_1, \omega_2) = F[R_f(n_1, n_2)] \quad (3.9b)$$

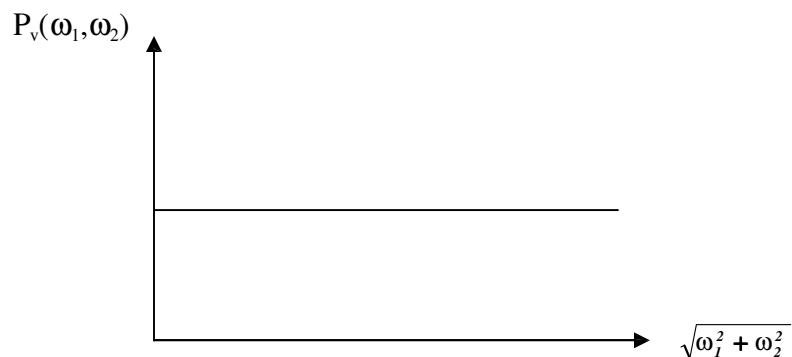
với hằng số  $0 < p < 1$ . Thông số  $p$  được ước lượng từ ảnh bị xuống cấp  $g(n_1, n_2)$ .



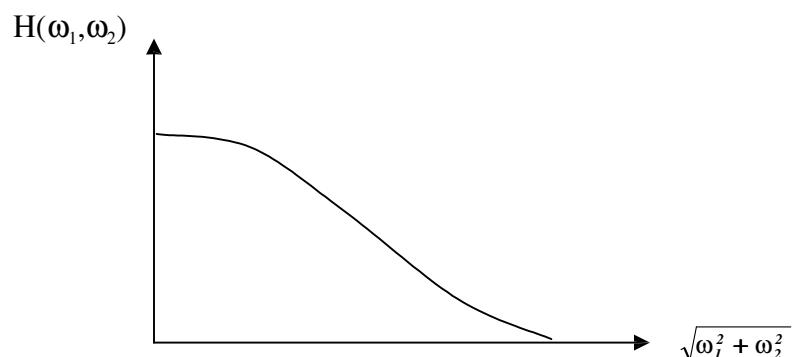
**Hình 9.3:** Bộ lọc Wiener không nhân quả cho việc ước lượng tuyến tính sai số quân phương tối thiểu của  $f(n_1, n_2)$  từ  $g(n_1, n_2) = f(n_1, n_2) + v(n_1, n_2)$ .



(a)



(b)



(c)

**Hình 9.4:** Minh họa rằng đáp ứng tần số của bộ lọc Wiener không nhân quả thường có đặc tính bộ lọc thông thấp.

Thông thường bộ lọc Wiener được thực thi trong miền tần số bởi

$$p(n_1, n_2) = \text{IDFT}[G(k_1, k_2) H(k_1, k_2)]. \quad (3.10)$$

Các dãy  $G(k_1, k_2)$  và  $H(k_1, k_2)$  biểu diễn hàm biến đổi Fourier rời rạc (DTF) của  $g(n_1, n_2)$  và  $h(n_1, n_2)$ . Trong công thức (3.10), kích thước của DFT và biến đổi DFT ngược có ít nhất cũng là  $(N + M - 1) \times (N + M - 1)$ , khi kích thước ảnh là  $N \times N$  và kích thước bộ lọc là  $M \times M$ . Nếu kích thước DFT nhỏ hơn  $(N + M - 1) \times (N + M - 1)$  thì hàm biến đổi Fourier ngược IDFT [ $G(k_1, k_2) H(k_1, k_2)$ ] sẽ không đồng nhất với  $g(n_1, n_2) * h(n_1, n_2)$  ở gần các đường biên của ảnh đã xử lý  $p(n_1, n_2)$ , vì hiệu ứng aliasing. Trong hầu hết các trường hợp, kích thước hiệu dụng của  $h(n_1, n_2)$  nhỏ, có thể nhận được kết quả vừa ý với biến đổi Fourier (DFT) và biến đổi ngược (IDFT) có kích thước  $N \times N$ . Một cách để nhận được  $H(k_1, k_2)$  là lấy mẫu đáp ứng tần số  $H(\omega_1, \omega_2)$  của bộ lọc Wiener bằng:

$$H(k_1, k_2) = H(\omega_1, \omega_2) \Big|_{\omega_1=2\pi k_1/L, \omega_2=2\pi k_2/L} \quad (3.11)$$

trong đó kích thước của DFT và IDFT là  $L \times L$ .

Bộ lọc Wiener thường là một bộ lọc thông thấp. Năng lượng của ảnh thường tập trung ở vùng tần số thấp. Vì nhiều nền nhiễu nhiên nhiên nói chung là băng rộng, nên đặc điểm bộ lọc Wiener là thông thấp. Hình 3.4 minh họa điều này. Hình 3.4(a) là một ví dụ của  $P_f(\omega_1, \omega_2)$ , nó giảm biên độ khi  $\omega_1$  và  $\omega_2$  tăng. Hình 3.4(b) là một ví dụ của  $P_v(\omega_1, \omega_2)$ , nó là hằng số, không phụ thuộc  $\omega_1$  và  $\omega_2$ . Hình 3.4(c) là bộ lọc Wiener nhận được,  $H(\omega_1, \omega_2)$  tính theo công thức (3.8) là có đặc tính lọc thông thấp.

Qua chương này, ta dựa vào sự so sánh chủ quan ảnh gốc, ảnh bị xuống cấp và ảnh đã xử lý của một quan sát viên minh họa hiệu năng của từng algorit phục hồi ảnh. Ngoài ra khi có sẵn thông tin, ta sẽ cung cấp sai số quân phương chuẩn hóa (NMSE) giữa ảnh gốc  $f(n_1, n_2)$  và ảnh bị xuống cấp  $g(n_1, n_2)$ , và giữa ảnh gốc  $f(n_1, n_2)$  và ảnh đã xử lý  $p(n_1, n_2)$ . NMSE giữa  $f(n_1, n_2)$  và  $p(n_1, n_2)$  được định nghĩa là:

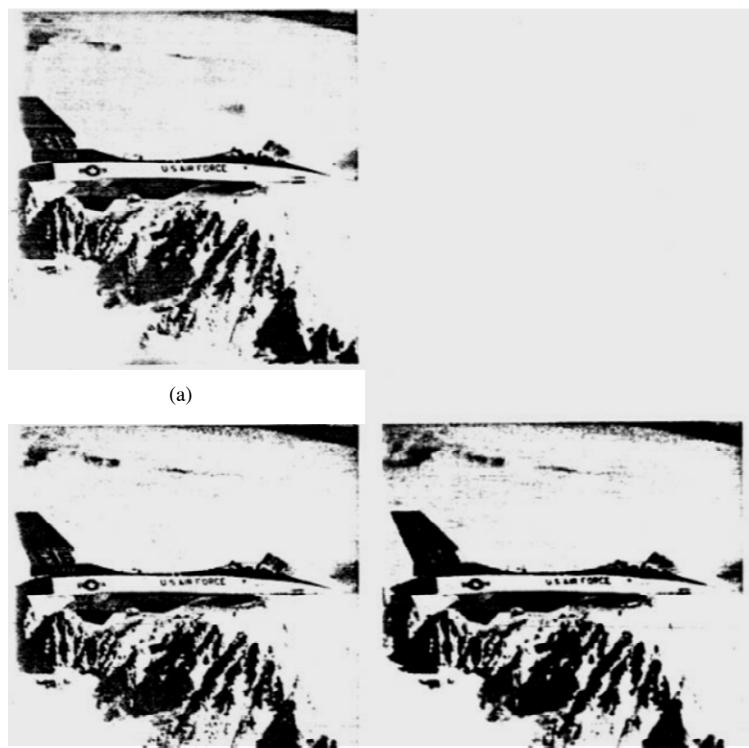
$$\text{NMSE}[f(n_1, n_2), p(n_1, n_2)] = 100 \times \frac{\text{Var}[f(n_1, n_2) - p(n_1, n_2)]}{\text{Var}[f(n_1, n_2)]}\% \quad (3.12)$$

Trong đó  $\text{Var}[\cdot]$  là phương sai. Sử dụng phương sai đảm bảo NMSE không bị ảnh hưởng khi cộng thêm độ thiên (bias) vào  $p(n_1, n_2)$ . Độ đo NMSE [ $f(n_1, n_2), p(n_1, n_2)$ ] được định nghĩa một cách tương tự. Mức cải thiện SNR do xử lý được định nghĩa là

$$\text{Mức cải thiện SNR} = 10\log_{10} \frac{\text{NMSE}[f(n_1, n_2), g(n_1, n_2)]}{\text{NMSE}[f(n_1, n_2), p(n_1, n_2)]} \text{dB.}$$

(9.13)

Một người quan sát hai ảnh bị xuống cấp với nguyên nhân như nhau, bao giờ cũng chọn cái có NMSE nhỏ hơn làm cái gần giống ảnh gốc hơn. NMSE rất bé thì có thể coi là ảnh gần như ảnh gốc. Tuy nhiên, cần lưu ý rằng NMSE chỉ là một trong nhiều độ đo khách quan có thể, và cũng có khi gây ra ngộ nhận. Chẳng hạn đem so sánh các ảnh bị xuống cấp bởi những nguyên nhân khác nhau, thì cái có NMSE nhỏ nhất không nhất thiết là cái gần ảnh gốc nhất. Như vậy, kết quả cải thiện NMSE và SNR chỉ mới có ý nghĩa tham khảo, chứ chưa thể dùng làm cơ sở để so sánh hiệu năng algorit này với algorit khác.



Hình 3.5:

- (a) Ảnh gốc 512x512 pixel;
- (b) Ảnh bị xuống cấp khi  $\text{SNR} = 7\text{dB}$  và  $\text{NMSE} = 19,7\%$ ;
- (c) Ảnh đã xử lý bởi bộ lọc Wiinter, với  $\text{NMSE} = 3,6\%$  và Mức cải thiện  $\text{SNR} = 7,4\text{dB}$ .

Hình 3.5 minh họa hiệu năng của một bộ lọc Wiener trong phục hồi ảnh. Hình 3.5(a) là ảnh gốc 512 x 512 pixels và hình 3.5(b) là ảnh bị xuống cấp bởi nhiễu Gauss trắng trung vị-không, SNR = 7dB. SNR theo định nghĩa trong chương 2 là

$$\text{SNR(dB)} = 10 \log_{10} \frac{\text{Var}[f(n_1, n_2)]}{\text{Var}[v(n_1, n_2)]} \quad (3.14)$$

Hình 3.5(c) là kết quả của việc áp dụng bộ lọc Wiener vào ảnh bị xuống cấp .Trong bộ lọc Wiener, giả thiết  $P_v(\omega_1, \omega_2)$  đã cho và  $P_f(\omega_1, \omega_2)$  ước lượng được bằng cách lấy giá trị trung bình của  $|F(\omega_1, \omega_2)|^2$  với 10 ảnh khác nhau. Khi bị xuống cấp bởi nhiễu trắng,  $P_v(\omega_1, \omega_2)$  là hằng số không phụ thuộc vào  $(\omega_1, \omega_2)$ . Sau khi xử lý, SNR của ảnh cải thiện được 7,4dB. Như ta thấy trên hình 3.5, bộ lọc Wiener làm giảm nhiễu nền rõ rệt. Điều đó cũng được chứng minh bởi sự cải thiện SNR. Tuy nhiên, nó cũng làm nhòe ảnh. Có nhiều phương án cải tiến bộ lọc Wiener để cải thiện hiệu năng. Tiếp sau sẽ thảo luận về vài phương án trong số đó.

## 2.2. CÁC BIẾN THỂ CỦA BỘ LỌC WIENER

Bộ lọc Wiener trình bày trong tiết 3.2.1 nhận được bằng cách tối thiểu hóa sai số quân phương giữa tín hiệu gốc và tín hiệu đã qua xử lý. Tuy nhiên, sai số quân bình phương không phải là tiêu chí mà người quan sát dùng trong việc đánh giá ảnh sau khi xử lý gần giống là ảnh gốc đến mức nào. Vì không nắm được tiêu chí mà con người sử dụng để đánh giá nên nhiều tác giả đã đề xuất những biến thể khác. Một biến thể là lọc phổ công suất. Trong phương pháp này, bộ lọc sử dụng có đáp ứng tần số  $H(\omega_1, \omega_2)$  như sau

$$H(\omega_1, \omega_2) = \left( \frac{P_f(\omega_1, \omega_2)}{P_f(\omega_1, \omega_2) + P_v(\omega_1, \omega_2)} \right)^{1/2} \quad (3.15)$$

Hàm  $H(\omega_1, \omega_2)$  trong (3.15) là căn bậc hai của đáp ứng tần số của bộ lọc Wiener. Nếu  $f(n_1, n_2)$  và  $v(n_1, n_2)$  là những mẫu của quá trình độc lập tuyến tính với nhau, thì ở đầu ra của bộ lọc sẽ có phổ công suất giống như phổ công suất tín hiệu gốc. Phương pháp này được gọi là *lọc phổ công suất*. Để chứng minh

$$P_p(\omega_1, \omega_2) = |H(\omega_1, \omega_2)|^2 P_g(\omega_1, \omega_2) \quad (3.16)$$

$$= | H(\omega_1, \omega_2) |^2 (P_f(\omega_1, \omega_2) + P_v(\omega_1, \omega_2)).$$

Từ (3.15) và (3.16),

$$P_p(\omega_1, \omega_2) = P_f(\omega_1, \omega_2). \quad (3.17)$$

Nhiều biến thể của bộ lọc Wiener dùng cho phục hồi ảnh có thể biểu diễn bằng  $H(\omega_1, \omega_2)$  sau đây:

$$H(\omega_1, \omega_2) = \left( \frac{P_f(\omega_1, \omega_2)}{P_f(\omega_1, \omega_2) + \alpha P_v(\omega_1, \omega_2)} \right)^\beta \quad (3.18)$$

Trong đó  $\alpha$  và  $\beta$  là các hằng số. Khi  $\alpha = 1$  và  $\beta = 1$ ,  $H(\omega_1, \omega_2)$  trở lại là bộ lọc Wiener. Khi  $\alpha = 1$  và  $\beta = \frac{1}{2}$ ,  $H(\omega_1, \omega_2)$  trở lại bộ lọc phổ công suất. Khi  $\alpha$  là thông số và  $\beta = 1$ , kết quả nhận được gọi là bộ lọc Wiener thông số. Vì  $H(\omega_1, \omega_2)$  trong (3.18) là dạng tổng quát hóa từ của bộ lọc Wiener, tất cả bình luận trong tiết 3.2.1 đều đúng cho lớp bộ lọc này. Chúng là những bộ lọc pha-không, có xu hướng giữ nguyên giá trị SNR của các phân hợp thành tần số cao. Phổ công suất  $P_f(\omega_1, \omega_2)$  và  $P_v(\omega_1, \omega_2)$  đều giả thiết đã biết và các bộ lọc thường được thực hiện bằng DFT và IDFT. Ngoài ra các bộ lọc này thường là bộ lọc thông thấp, chúng giảm nhiễu nhưng làm nhòe cho ảnh ở mức đáng kể. Hiệu năng của lọc phổ công suất biểu diễn trên hình 3.6. Ảnh gốc và ảnh bị xuống cấp như trên hình 3.5. Mức cải thiện SNR 6.6dB.



**Hình 3.6:** Ảnh trong hình 3.5(a) được xử lý bởi

bộ lọc phổ công suất, có NMSE = 4,3%

và SNR cải thiện = 6.6 dB.

### 2.3. XỬ LÝ ẢNH THÍCH NGHI

Lý do bộ lọc Wiener và các biến thể của nó làm nhòe ảnh là do sử dụng một bộ lọc duy nhất trên toàn bộ ảnh. Bộ lọc Wiener được triển khai với giả thiết là, qua các vùng khác nhau của ảnh đặc tính tín hiệu và nhiễu đều không thay đổi. Đó là bộ lọc bất biến trong không gian. Thông thường trong một bức ảnh, từ vùng này sang vùng khác các đặc tính ảnh rất khác nhau. Ví dụ, tường và bầu trời có cường độ nền xấp xỉ đồng đều, trái lại các tòa nhà và cây có cường độ thay đổi lớn, chi tiết. Sự xuống cấp cũng có thể thay đổi từ một vùng qua vùng khác. Như vậy thì nên thích nghi phép xử lý theo sự thay đổi của đặc tính của ảnh và sự xuống cấp. Ý tưởng xử lý thích nghi theo các đặc tính cục bộ của ảnh không những có ích cho phục hồi ảnh mà còn có ích trong nhiều ứng dụng xử lý ảnh khác, kể cả phép cải thiện ảnh đã thảo luận trong chương 2.

Có hai cách tiếp cận tới xử lý ảnh thích nghi đã được triển khai. Cách tiếp cận đầu tiên được gọi là *xử lý từng pixel* (pixel processing), quá trình xử lý được thích nghi ở mỗi pixel. Phương pháp xử lý thích nghi ở từng pixel dựa trên các đặc tính cục bộ của ảnh, sự xuống cấp và mọi thông tin hữu quan khác trong vùng lân cận từng pixel một. Vì mỗi pixel được xử lý khác nhau, cách tiếp cận này có tính thích nghi cao và không có những mất liên tục cường độ nhân tạo trong ảnh đã xử lý. Tuy nhiên, cách tiếp cận này chi phí tính toán cao và thường chỉ thực hiện trong miền không gian.

Cách tiếp cận thứ hai, được gọi là *xử lý từng ảnh con* (subimage processing) hoặc *xử lý từng khối* (block-by-block processing), ảnh được chia ra làm nhiều ảnh con và mỗi ảnh con được xử lý riêng rẽ và sau đó đem kết hợp lại với nhau. Kích thước ảnh con thường trong khoảng 8 x 8 và 32 x 32 pixels. Với từng ảnh con, dựa trên cơ sở của các đặc tính cục bộ của ảnh, sự xuống cấp và mọi thông tin hữu quan khác trong vùng, thực hiện phép lọc không gian bất biến thích hợp cho ảnh con được chọn. Vì phép xử lý áp dụng tới từng ảnh con là lọc không gian bất biến, nên thực hiện mềm dẻo hơn xử lý từng pixel. Chẳng hạn, một bộ lọc thông thấp có thể thực hiện trong cả miền không gian hoặc miền tần số. Ngoài ra, nói chung *xử lý từng ảnh con* chi phí tính toán ít hơn *xử lý từng pixel*, vì phép xử lý đem sử dụng chỉ phải xác định một lần cho toàn bộ ảnh con. Vì phép xử lý thay đổi đột ngột khi ta chuyển từ một ảnh con tới ảnh tiếp theo, nên có thể xuất hiện những mất liên tục cường độ theo dọc đường biên của các ảnh con lân cận, điều này được gọi là *hiệu ứng khối*. Trong một vài ứng dụng,

như phục hồi ảnh trong môi trường SNR cao thì hiệu ứng khói có thể không xuất hiện và không cần phải xét đến. Trong các ứng dụng khác, như mã hoá biến đổi với tốc độ bít thấp, hiệu ứng khói có thể rất rõ và là đặc tính đáng chê trách nhất của ảnh đã xử lý.

Trong một số trường hợp có thể làm giảm hiệu ứng khói bằng cách cho các vùng đường bao ảnh con của ảnh đã xử lý qua bộ lọc thông thấp. Một phương pháp khác làm giảm hiệu ứng khói là cho các ảnh con gối mép nhau. Trong phương pháp này, để nhận được một ảnh con, ta đem một cửa sổ  $w_{ij}(n_1, n_2)$  áp dụng vào ảnh đã xử lý  $g(n_1, n_2)$ . Cửa sổ  $w_{ij}(n_1, n_2)$  phải thoả mãn hai điều kiện. Điều kiện thứ nhất có thể biểu diễn là:

$$\sum_i \sum_j w_{ij}(n_1, n_2) = 1 \text{ cho mọi giá trị } (n_1, n_2) \text{ hữu quan} \quad (3.19)$$

điều kiện này đảm bảo rằng khi đem cộng đơn giản các ảnh con chưa xử lý sẽ nhận lại được ảnh gốc. Điều kiện thứ hai yêu cầu  $w_{ij}(n_1, n_2)$  là một hàm trơn mà giá trị sụt xuống gần bằng không khi đến gần đường bao của sổ. Điều này xu hướng làm giảm những chỗ không liên tục hoặc xuống cấp có thể xuất hiện ở vùng đường biên ảnh con trong ảnh đã xử lý.

Một cách để tìm hàm cửa sổ 2-D nhẵn thoả mãn cả hai điều kiện trên là hình thành một cửa sổ 2-D tách được từ hai cửa sổ 1-D thoả mãn được những điều kiện tương tự.

$$w_{ij}(n_1, n_2) = w_i(n_1) w_j(n_2) \quad (3.20)$$

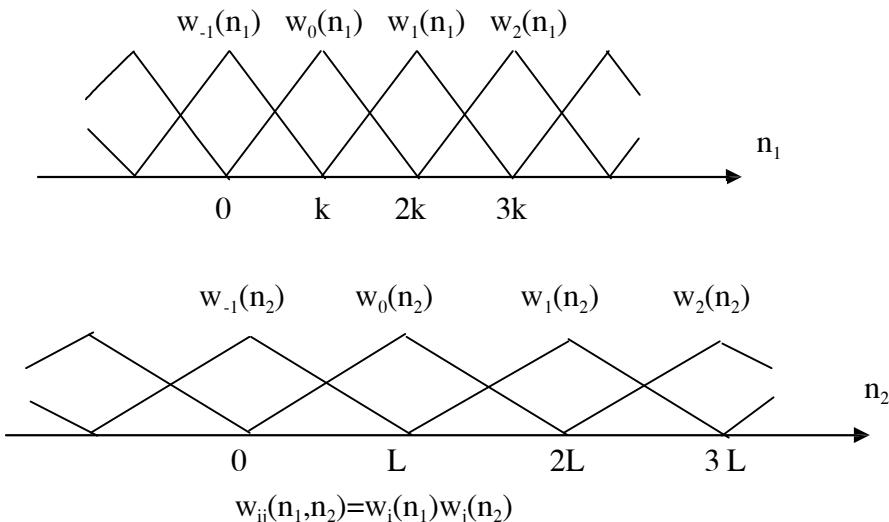
Hai hàm cửa sổ như vậy là cửa sổ 2-D tách được hình tam giác và cửa sổ Ham-ming gối mép lên các cửa sổ lân cận trong nửa thời gian cửa sổ trên mỗi chiều. Cửa sổ tam giác 2-D tách được biểu diễn trên hình 3.7. Trong xử lý ảnh con, phải xét đến cửa sổ sử dụng để hình thành ảnh con.

Có nhiều biến thể của các phép xử lý từng pixel và xử lý từng ảnh con. Chẳng hạn thiết kế một bộ lọc cho mỗi khối  $8 \times 8$  hoặc  $32 \times 32$  pixel, nhưng lại đem áp dụng cho kiểu xử lý từng pixel.

Một hệ xử lý thích nghi tổng quát được biểu diễn trên hình 3.8. Phép xử lý phải thực hiện ở mỗi pixel hoặc mỗi ảnh con, thích nghi theo các đặc tính cục bộ của ảnh, sự xuống cấp và mọi thông tin hữu quan khác trong vùng. Kiến thức về các đặc tính này có thể nhận được từ hai nguồn. Một là một vài thông tin sẵn có mà ta có thể biết. Chẳng hạn, loại ảnh mong đợi đối với một ứng dụng đã cho, hoặc các đặc điểm xuống cấp từ một nguyên nhân gây xuống cấp đã biết. Một nguồn thông tin khác là ảnh được xử lý.

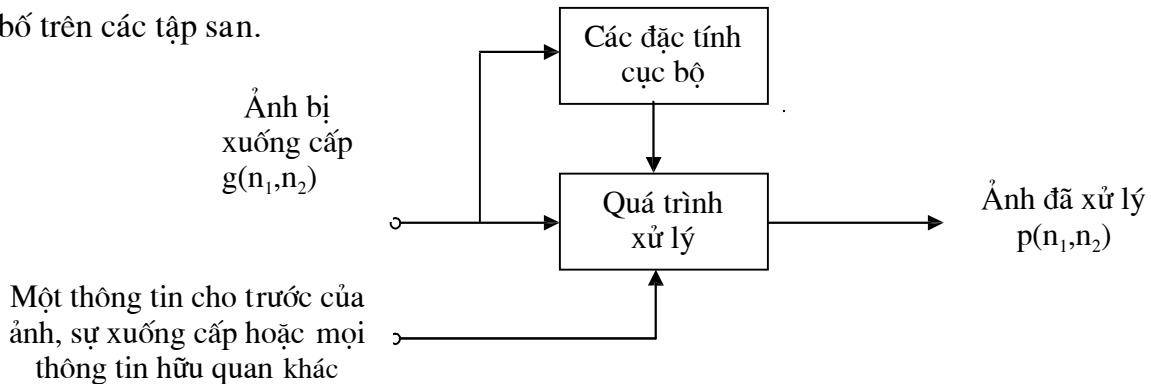
Bằng các phép đo của các đặc điểm như phương sai cục bộ, có thể xác định sự tồn tại của những chi tiết tân số quan trọng.

Việc xác định sử dụng loại xử lý gì phụ thuộc vào nhiều yếu tố, bao gồm loại kiến thức mà ta biết về ảnh và cách khai thác kiến thức này để ước lượng các thông số của phương pháp xử lý, ví dụ tân số cắt của bộ lọc thông thấp. Không có bối cảnh cụ thể của ứng dụng, thường chỉ có thể đưa ra những định hướng chung nhất mà thôi. Những hiểu biết sẵn có càng nhiều thì chất lượng xử lý càng cao. Nếu thông tin sẵn có không chính xác thì hiệu năng của hệ xử lý sẽ kém cỏi. Nói chung, xử lý từng ảnh con thì quy tắc thích nghi phải tinh tế hơn, còn xử lý từng pixel thì quy tắc thích nghi đơn giản hơn.



**Hình 3.7:** Ví dụ về Cửa sổ tam giác 2-D tách.

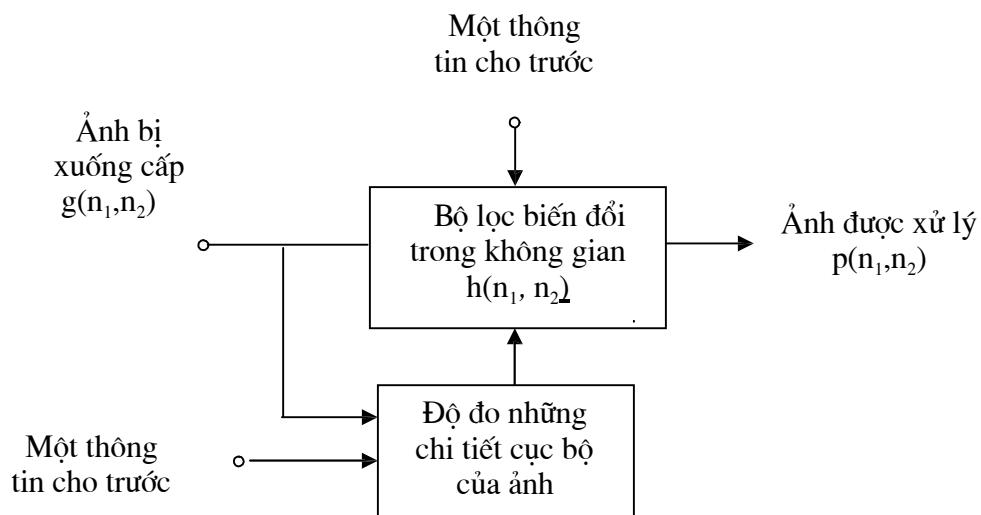
Khi áp dụng xử lý ảnh thích nghi để phục hồi ảnh bị xuống cấp bởi nhiễu công ngẫu nhiên, có thể làm giảm nhiễu nền mà không gây ra nhoè ảnh đáng kể. Trong bốn tiết tiếp theo ta thảo luận về một vài hệ phục hồi ảnh thích n ghi chọn trong số đã công bố trên các tập san.



**Hình 3.8:** Hệ xử lý ảnh thích nghi tổng quát.

#### 2.4. BỘ LỌC WIENER THÍCH NGHI.

Hầu hết các algorit phục hồi thích nghi dùng để giảm nhiễu cộng trong ảnh đều có thể biểu diễn bằng hệ ở trên hình 3.9. Từ ảnh bị xuống cấp và những thông tin cho trước, có thể xác định ra phép đo những chi tiết cục bộ của ảnh không nhiễu. Một trong những phép đo là phương sai cục bộ. Từ đó xác định được bộ lọc biến đổi trong không gian  $h(n_1, n_2)$ , - một hàm của các chi tiết cục bộ của ảnh và những thông tin cho trước.



**Hình 3.9:** Hệ phục hồi ảnh thích nghi điển hình cho việc giảm nhiễu cộng.

Bộ lọc biến đổi trong không gian ấy được áp dụng vào ảnh xuống cấp tại vùng cục bộ mà người ta đã lấy thông tin để thiết kế nó. Khi nhiễu là băng rộng, bộ lọc biến đổi trong không gian  $h(n_1, n_2)$  có đặc tính bộ lọc thông thấp. Trong vùng ảnh ít chi tiết như các vùng cường độ đồng đều, ở đó nhiễu hiển thị rõ hơn ở vùng nhiều chi tiết, dùng lọc thông thấp sâu (tần số cắt thấp) để làm giảm nhiễu càng nhiều càng tốt. Vì trong vùng ít chi tiết biến thiên của tín hiệu nhỏ, lọc thông thấp sâu không làm ảnh hưởng đến phân hợp thành tín hiệu. Trong vùng ảnh nhiều chi tiết như ở vùng biên, có một phân hợp thành lớn của tín hiệu, chỉ nên lọc thông thấp ít để không làm méo (nhoè) phân hợp thành tín hiệu. Như vậy không làm giảm nhiễu nhiều, nhưng với cùng mức nhiễu thì ở vùng ảnh có nhiều chi tiết không thấy rõ nhiều như trong vùng ít chi tiết.

Có thể triển khai một số algorit khác nhau, tuỳ theo độ đo cụ thể được dùng để biểu thị chi tiết cục bộ của ảnh. Bộ lọc thay đổi trong không gian  $h(n_1, n_2)$  được xác định như thế nào là tuỳ theo chi tiết cục bộ của ảnh và những thông tin có sẵn. Một trong nhiều cách là thiết kế thích nghi và thực hiện bộ lọc Wiener đã thảo luận trong tiết 3.2.1. Như biểu diễn trên hình 3.3, bộ lọc Wiener yêu cầu phải biết giá trị trung vị của tín hiệu  $m_f$ , giá trị trung vị của nhiều  $m_v$ , phổ công suất tín hiệu  $P_f(\omega_1, \omega_2)$  và phổ công suất nhiễu  $P_v(\omega_1, \omega_2)$ . Thay vì giả thiết  $m_f$ ,  $m_v$ ,  $P_f(\omega_1, \omega_2)$  và  $P_v(\omega_1, \omega_2)$  là cố định trên toàn bộ ảnh, ta ước lượng chúng trong từng vùng. Cách tiếp cận này dẫn đến bộ lọc Wiener biến đổi trong không gian. Tuy cùng một cách tiếp cận nhưng có thể có nhiều biến thể, tuỳ theo cách ước lượng cục bộ  $m_f$ ,  $m_v$ ,  $p_f(\omega_1, \omega_2)$  và  $p_v(\omega_1, \omega_2)$  và cách thực hiện bộ lọc Wiener biến đổi trong không gian. Ta sẽ khai triển một algorit để minh họa cách tiếp cận này.

Trước tiên ta giả thiết rằng nhiều cộng  $v(n_1, n_2)$  có trung vị bằng không và nhiễu trắng có phương sai là  $\sigma_v^2$ . Phổ công suất  $P_v(\omega_1, \omega_2)$  khi ấy là

$$P_v(\omega_1, \omega_2) = \sigma_v^2 \quad (3.21)$$

Xét một vùng nhỏ ở đó tín hiệu  $f(n_1, n_2)$  có thể coi là dừng. Trong vùng đó tín hiệu  $f(n_1, n_2)$  có mô hình là

$$f(n_1, n_2) = m_f + \sigma_f w(n_1, n_2) \quad (3.22)$$

trong đó  $m_f$  và  $\sigma_f$  là trung vị cục bộ và độ lệch chuẩn của  $f(n_1, n_2)$ ; còn  $w(n_1, n_2)$  là nhiễu trắng có trung vị bằng không và phương sai đơn vị. Theo kinh nghiệm (3.22) là một mô hình hợp lý đối với các loại ảnh thường gặp.

Trong (3.22), mô hình tín hiệu  $f(n_1, n_2)$  là tổng của trung vị cục bộ  $m_f$  (của biến đổi trong không gian) và phương sai cục bộ  $\sigma_f^2$  (của nhiễu trắng biến đổi trong không gian). Khi ấy bộ lọc Wiener  $H(\omega_1, \omega_2)$  là:

$$\begin{aligned} H(\omega_1, \omega_2) &= \frac{P_f(\omega_1, \omega_2)}{P_f(\omega_1, \omega_2) + P_v(\omega_1, \omega_2)} \\ &= \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2}. \end{aligned} \quad (3.23)$$

Từ (3.23), suy ra đáp ứng xung

$$h(n_1, n_2) = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2} \delta(n_1, n_2) \quad (3.24)$$

Từ (3.24) và hình 3.3, suy ra ảnh được xử lý trong vùng cục bộ là:

$$\begin{aligned} p(n_1, n_2) &= m_f + (g(n_1, n_2) - m_f) * \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2} \delta(n_1, n_2) \\ &= m_f + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2} (g(n_1, n_2) - m_f). \end{aligned} \quad (3.25)$$

Nếu ta giả thiết rằng  $m_f$  và  $\sigma_f^2$  được cập nhật ở mỗi pixel.

$$p(n_1, n_2) = m_f(n_1, n_2) + \frac{\sigma_f^2(n_1, n_2)}{\sigma_f^2(n_1, n_2) + \sigma_v^2} (g(n_1, n_2) - m_f(n_1, n_2)). \quad (3.26)$$

Phương trình (3.26) là cốt lõi của algorit do Lee phát triển năm 1980.

Algorit dựa trên cơ sở (3.26) có thể được xem như trường hợp đặc biệt của xử lý hai kênh. Trong xử lý hai kênh xử lý ảnh được xử lý chia làm hai phần, trung vị cục bộ  $m_f(n_1, n_2)$  và độ tương phản cục bộ  $g(n_1, n_2) - m_f(n_1, n_2)$ . Trung vị cục bộ và độ tương phản cục bộ được xử lý riêng rẽ và rồi đem kết quả được tổ hợp lại. Trong trường hợp (3.26) trung vị cục bộ được giữ không đổi trong khi độ tương phản thay đổi theo các biến độ tương đối của  $\sigma_f^2$  và  $\sigma_v^2$ . Nếu  $\sigma_f^2 \gg \sigma_v^2$ , độ tương phản tại chỗ của  $g(n_1, n_2)$  coi như chủ yếu là do  $f(n_1, n_2)$  và độ tương phản của  $g(n_1, n_2)$  không giảm. Trong trường hợp đó  $p(n_1, n_2)$  xấp xỉ bằng  $g(n_1, n_2)$ , trong vùng như vậy không cần xử lý gì nhiều.

Nếu  $\sigma_f^2 \ll \sigma_v^2$ , độ tương phản tại chỗ của  $g(n_1, n_2)$  coi như chủ yếu là do  $v(n_1, n_2)$  và độ tương phản của  $g(n_1, n_2)$  suy giảm nhiều. Trong trường hợp này  $p(n_1, n_2)$  xấp xỉ bằng  $m_f$ ,  $g(n_1, n_2)$  bị làm nhãm một cách đáng kể. Một ví dụ khác của xử lý hai kênh là algorit thích nghi được khai triển trong tiết 2.1.4 để làm giảm ảnh hưởng của lớp mây che phủ ảnh chụp từ máy bay.

Chú ý rằng  $m_f$  đồng nhất bằng  $m_g$  khi  $m_v = 0$ , ta có thể ước lượng  $m_f(n_1, n_2)$  trong (3.26) từ  $g(n_1, n_2)$  bằng công thức

$$\hat{m}_f(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} g(k_1, k_2) \quad (3.27)$$

trong đó  $(2M+1)^2$  là số lượng pixels trong vùng cục bộ được sử dụng khi ước lượng. Bên trong vùng cục bộ ở đó  $\sigma_f^2(n_1, n_2)$  có thể coi là bất biến trong thời gian, thế  $\hat{m}_f(n_1, n_2)$  trong (3.27) vào  $m_f(n_1, n_2)$  trong (3.26) nhận được

$$p(n_1, n_2) = g(n_1, n_2) * h(n_1, n_2) \quad (3.28a)$$

trong đó

$$h(n_1, n_2) = \begin{cases} \frac{\sigma_f^2 + \frac{\sigma_v^2}{(2M+1)^2}}{\sigma_f^2 + \sigma_v^2}, & n_1 = n_2 = 0 \\ \frac{\sigma_v^2}{(2M+1)^2}, & -M \leq n_1 \leq M, -M \leq n_2 \leq M \\ 0, & \text{Ngoài trừ } n_1 = n_2 = 0. \\ & \text{Các trường hợp khác.} \end{cases} \quad (3.28b)$$

Hình 3.10 là bộ lọc  $h(n_1, n_2)$  khi  $\sigma_f^2 \gg \sigma_v^2$ ,  $\sigma_f^2 = \sigma_v^2$  và  $\sigma_f^2 \ll \sigma_v^2$ , với  $M = 1$ . Từ hình 3.10 thấy rằng, khi  $\sigma_f^2$  giảm so với  $\sigma_v^2$ , nhiễu được làm nhãm nhiều hơn. Để đo chi tiết tín hiệu cục bộ trong hệ ở hình 3.9, algorit được khai triển đã sử dụng phương sai tín hiệu  $\sigma^2$ . Phương pháp cụ thể được sử dụng để thiết kế bộ lọc biến đổi theo không gian  $h(n_1, n_2)$  dựa vào (3.28b). Việc thiết kế bộ lọc biến đổi trong không gian  $h(n_1, n_2)$  là đơn giản và bộ lọc  $h(n_1, n_2)$  nhận được thường là một bộ lọc FIR nhỏ (kích thước  $3 \times 3, 5 \times 5$  hoặc  $7 \times 7$ ), và thường áp dụng xử lý từng pixel,

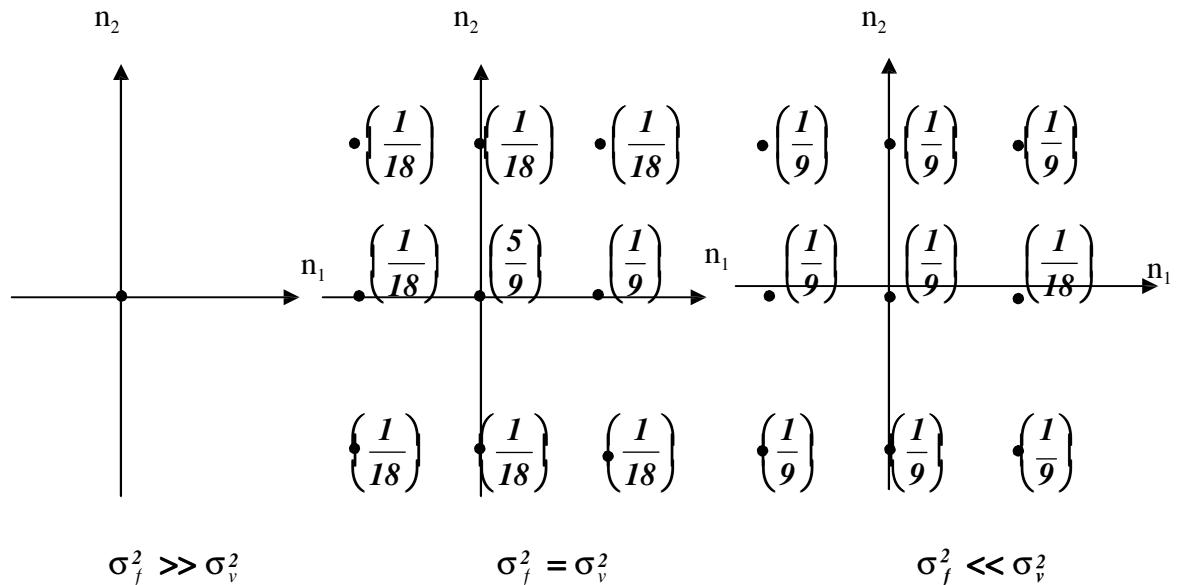
Vì  $\sigma_g^2 = \sigma_f^2 + \sigma_v^2$ ,  $\sigma_f^2$  có thể được ước lượng từ  $g(n_1, n_2)$  bằng

$$\hat{\sigma}_g^2(n_1, n_2) = \begin{cases} \sigma_g^2(n_1, n_2) - \sigma_v^2, & \text{nếu } \sigma_g^2(n_1, n_2) > \sigma_v^2 \\ 0, & \text{các trường hợp khác.} \end{cases} \quad (3.29a)$$

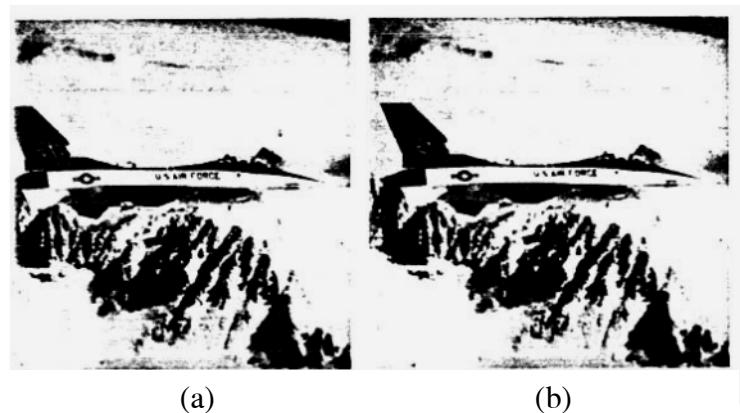
trong đó

$$\hat{\sigma}_g^2(n_1, n_2) = \frac{1}{(2M+1)^2} \sum_{k_1=n_1-M}^{n_1+M} \sum_{k_2=n_2-M}^{n_2+M} (g(k_1, k_2) - \hat{m}_f(n_1, n_2))^2$$

(3.29b) ước lượng trung vị cục bộ  $\hat{m}_f(n_1, n_2)$  có thể nhận được từ (3.27), và  $\sigma_v^2$  giả thiết là đã biết.



**Hình** (a) (b) (c) ảnh như là một hàm của  $\sigma_f^2$  và  $\sigma_v^2$ . Khi (phương sai tín hiệu của)  $\sigma_f^2 \gg \sigma_v^2$  (phương sai của nhiễu), thì bộ lọc gần như  $\delta(n_1, n_2)$ . Khi  $\sigma_f^2$  giảm so với  $\sigma_v^2$ ,  $h(n_1, n_2)$  gần như của số hình chữ nhật.



**Hình 3.11:** Minh họa hiệu năng của một phương pháp lọc Wiener thích nghi. Sử dụng ảnh bị xuống cấp trong hình 3.5(b).

- (a) ảnh được xử lý bởi lọc thích nghi, với NMSE = 3,8% và mức cải thiện SNR = 7,1dB.  
 (b) ảnh được xử lý bởi bộ lọc Wiener không gian bất biến ,với NMSE = 3,6% và mức cải thiện SNR =7,4dB.

Hình 3.11 minh họa hiệu năng algorit này. Hình 3.11(a) là ảnh được xử lý. Ảnh gốc và ảnh bị xuống cấp biểu diễn trên các hình 3.5(a) và (b). Sự xuống cấp tạo nên ảnh ở hình 3.5(b) là nhiễu công trắng Gauss. Mức cải thiện SRN là 7,4 dB. Ảnh sau xử lý nhận được bằng cách sử dụng các công thức (3.27), (3.28), (3.29) với  $M = 2$ . Từ ảnh được xử lý, thấy rằng nhiễu đã được làm giảm nhiều mà không gây nhòe ảnh. Nếu sử dụng bộ lọc không thích nghi thì với mức giảm nhiễu này sẽ kèm theo nhòe ảnh ở mức có thể nhận thấy. Hình 3.11(b) là kết quả sử dụng bộ lọc Wiener không thích nghi. Hình 3.11(b) giống như hình 3.5(c).

## 2.5. PHỤC HỒI ẢNH THÍCH NGHI DỰA VÀO HÀM RỖ NHIỄU.

Khi triển khai algorit thích nghi phục hồi ảnh trong tiết 3.2.4 không sử dụng một độ đo nào để định lượng mức nhiễu mà thị giác người xem cảm nhận được. Nếu có được độ đo này thì có thể sử dụng để triển khai một hệ phục hồi ảnh. Hàm biểu diễn độ đo đó sẽ được gọi là *hàm rõ nhiễu* (noise visibility function), nó phụ vào loại nhiễu và cũng phụ thuộc vào loại tín hiệu mà nó được cộng thêm vào. Nhiễu trắng và nhiễu mâu cùng mức nói chung có ảnh hưởng khác nhau tới người quan sát. Vùng ảnh nhiều chi tiết sẽ che lấp nhiễu tốt hơn vùng ảnh ít chi tiết.

Có nhiều cách để định nghĩa và đo hàm độ rõ nhiễu. Ta sẽ thảo luận cách mà Anderson và Netrvalali sử dụng trong việc triển khai một hệ phục hồi ảnh. Giả thiết nhiễu nên gây ra sự xuống cấp là nhiễu trắng, mặc dù cách tiếp cận này cũng áp dụng được với các loại nhiễu khác. Gọi  $M(n_1, n_2)$  là một độ đo nào đó về chi tiết ảnh cục bộ của một ảnh gốc  $f(n_1, n_2)$ . Hàm  $M(n_1, n_2)$  được gọi là *hàm che lấp* (masking function), vì vùng nhiều chi tiết ( $M$  cao) che lấp nhiễu tốt hơn vùng ít chi tiết ( $M$  thấp). *Hàm rõ nhiễu*  $V(M)$  được định nghĩa để biểu diễn độ rõ tương đối của một mức nhiễu đã cho ở mức che lấp  $M$ . Cụ thể hơn, ta giả sử nhiễu với phương sai  $\sigma_1^2$  ở  $M = M_1$  được người xem nhận thấy cũng rõ như nhiễu với phương sai  $\sigma_2^2$  ở  $M = M_2$ , thì hàm  $V(M)$  được định nghĩa bởi:

$$\sigma_1^2 V(M_1) = \sigma_2^2 V(M_2) \quad (3.30)$$

Độ rõ nhiễu  $V(M)$  ở  $M = M_1$  càng cao thì mức nhiễu  $\sigma_1^2$  cần thiết để đạt độ rõ bằng mức nhiễu cố định  $\sigma_2^2$  ở mức che lấp cố định  $M_2$  càng thấp. Cùng mức nhiễu nhưng ở

vùng ít chi tiết ( $M$  nhỏ) thì nhìn thấy nhiều rõ hơn, nên có thể ước đoán là hàm  $V(M)$  giảm khi  $M$  tăng.

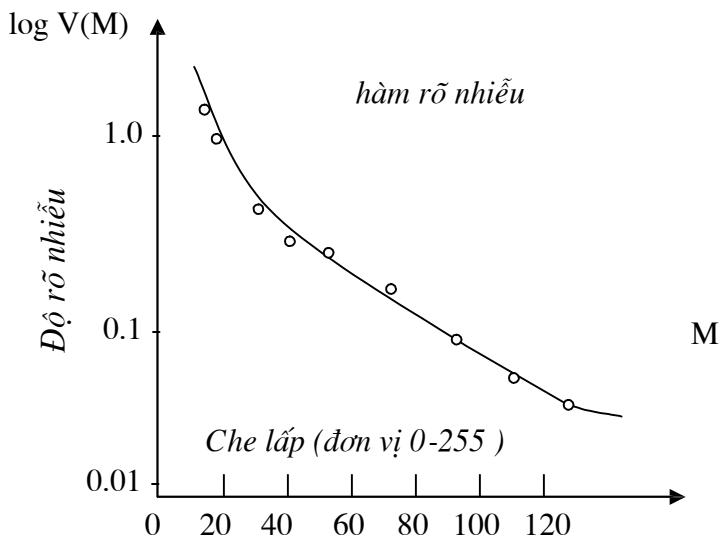
Ít ra là trên lý thuyết, có thể sử dụng (3.30) để đo *hàm rõ nhiều*  $V(M)$ , với kết quả chỉ chênh lệch một hệ số tỉ lệ. Giả sử đem cộng thêm nhiều với phư ơng sai  $\sigma_1^2$  vào một vùng ảnh cục bộ có mức che lấp là  $M_1$ . Ta có thể yêu cầu người quan sát so sánh độ rõ nhiều trong vùng cục bộ này với một vùng ảnh khác ở đó  $M$  là  $M_2$  và mức nhiều sử dụng là  $\sigma^2$ . Cho phép người xem thay đổi  $\sigma^2$  sao cho nhiều trong cả hai vùng có độ rõ như nhau và ký hiệu giá trị  $\sigma^2$  chọn được là  $\sigma_2^2$ . Ta gọi thí nghiệm tâm vật lý(psycho-physical experiment) này là thí nghiệm phối hợp độ rõ nhiều (visibility matching experiment). Từ  $\sigma_1^2$  sử dụng trong thí nghiệm và  $\sigma_2^2$  được người quan sát chọn, ta có thể xác định  $V(M_2)/V(M_1)$  theo  $\sigma_1^2/\sigma_2^2$ .

Phương trình (3.30) có thể căn cứ vào những giả thiết khác nhau. Chẳng hạn giả thiết  $V(M)$  chỉ phụ thuộc vào  $M$ . Như vậy hàm che lấp  $M(n_1, n_2)$  phải được chọn sao cho khi mức nhiều như nhau thì trong tất cả các vùng ảnh có cùng giá trị  $M$  độ rõ nhiều phải như nhau. Cách chọn  $M$  theo đề xuất của Anderson và Netrvalali là:

$$M(n_1, n_2) =$$

$$\sum_{k_1=n_1-L}^{n_1+L} \sum_{k_2=n_2-L}^{n_2+L} 0,35^{\sqrt{(k_1-n_1)^2 + (k_2-n_2)^2}} (|f(k_1+1, k_2) - f(k_1-1, k_2)| + |f(k_1, k_2+1) - f(k_1, k_2-1)|) \quad (3.31)$$

trong đó  $f(n_1, n_2)$  là ảnh không nhiều (hay ảnh gốc) và  $(2L+1) \times (2L+1)$  là kích thước của vùng cục bộ sử dụng trong việc đo *mức che lấp*  $M$  ở điểm  $(n_1, n_2)$ . Trong (3.31),  $M(n_1, n_2)$  tăng khi độ dốc theo phương ngang và phương dọc của  $f(n_1, n_2)$  tăng. Tác dụng của độ dốc theo phương ngang và phương dọc đến  $M(n_1, n_2)$  giảm theo hàm mũ khi khoảng cách Oclid giữa  $(n_1, n_2)$  và điểm tiến hành đo độ dốc tăng. Trong (3.30) giả định hàm rõ nhiều được giữ nguyên khi hệ số tỉ lệ của  $\sigma_1$  và  $\sigma_2$  như nhau. Giả định này chỉ đúng trong một vùng nhỏ của hệ số tỉ lệ.



**Hình 3.12:** Hàm rõ nhiễu  $V(M)$ .

Ngoài các giả định đã đặt ra cho (3.30), vốn cũng chỉ là xấp xỉ gần đúng, có nhiều khó khăn thực tế khi dùng (3.30) để đo  $V(M)$ . Trong một bức ảnh điển hình, số lượng pixels ứng với một mức  $M$  đã cho có thể ít, đặc biệt là khi  $M$  lớn. Trong trường hợp như vậy, dùng thí nghiệm phối hợp độ rõ nhiễu để đo  $V(M)$  sẽ khó khăn. Tuy vậy dựa vào (3.30) và (3.31) và thí nghiệm phối hợp độ rõ nhiễu cũng đã đo được  $V(M)$  một cách xấp xỉ. Kết quả biểu diễn trên hình 3.12. Như dự đoán,  $V(M)$  giảm khi  $M$  tăng trong một dải rộng của  $M$ .

Có nhiều cách sử dụng hàm rõ nhiễu để khai triển algorit phục hồi ảnh. Ta sẽ khai triển một algorit phục hồi, có thể xem như trường hợp đặc biệt của hệ phục hồi thích nghi biểu diễn trên hình 3.9. Trong algorit này, bộ lọc biến đổi trong không gian  $h(n_1, n_2)$  có dạng Gauss, tính theo công thức:

$$h(n_1, n_2) = k \cdot \exp(- (n_1^2 + n_2^2)/2\sigma^2) w(n_1, n_2) \quad (3.32)$$

trong đó  $k$  và  $\sigma^2$  được xác định một cách thích nghi và  $w(n_1, n_2)$  là một cửa sổ hình chữ nhật, nó giới hạn vùng kích thước của  $h(n_1, n_2)$ . Để xác định  $k$  và  $\sigma^2$ , có một điều kiện ràng buộc là,

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} h(n_1, n_2) = I \quad (3.33)$$

Một điều ràng buộc khác là nhiễu trong ảnh được xử lý phải có độ rõ như nhau trên toàn ảnh. Để thoả mãn điều kiện ràng buộc này, lưu ý rằng theo lý thuyết cơ bản về quá trình ngẫu nhiên, khi nhiễu gây ra xuống cấp  $V(n_1, n_2)$  là nhiễu trắng với phương sai  $\sigma_v^2$ , nhiễu trong ảnh được nhuộm màu với phương sai  $\sigma_p^2$ , trong đó:

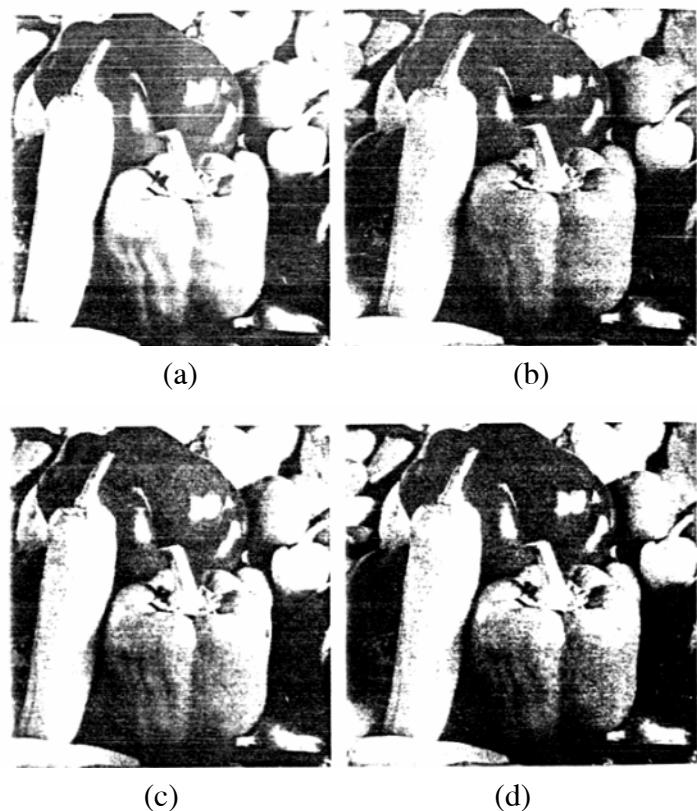
$$\sigma_p^2 = \sigma_v^2 \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)|^2 \quad (3.34)$$

Nếu ta chọn  $h(n_1, n_2)$  trong mỗi vùng sao cho  $\sigma_p^2$  thoả mãn.

$$\sigma_p^2 V(M) = \text{hằng số(constant)} \quad (3.35)$$

mức nhiễu còn lại trên toàn bộ bức ảnh đã xử lý sẽ bằng nhau khi nào  $V(M)$  còn phản ánh chính xác định nghĩa trong (3.30) và  $V(M)$  cho nhiễu trắng và nhiễu màu xấp xỉ như nhau. Hằng số trong công thức (3.35) được chọn sao cho đạt được sự dung hoà giữa giảm nhiễu và gây nhòe. Nếu chọn hằng số quá lớn thì nhiễu nền giảm rất ít. Nếu chọn hằng số quá nhỏ thì giảm được nhiều nhưng gây ra méo tín hiệu (nhòe) nhiều. Ở mỗi pixel, bộ lọc biến đổi trong không gian  $h(n_1, n_2)$  có thể được định nghĩa từ (3.32), (3.33), (3.34) và (3.35). Vì trong algorit này các thông số  $k$  và  $\sigma^2$  của bộ lọc chỉ phụ thuộc vào  $M$ , nên có thể tính sẵn  $k$  và  $\sigma^2$  và lưu trữ trong một bảng như một hàm của  $M$ . Để phục hồi một ảnh, ta ước lượng  $M(n_1, n_2)$  của ảnh không nhiễu  $f(n_1, n_2)$  từ ảnh bị xuống cấp, và lấy  $k(n_1, n_2)$ ,  $\sigma^2(n_1, n_2)$  từ bảng tính sẵn. Ở mỗi pixel  $(n_1, n_2)$ , bộ lọc biến đổi trong không gian  $h(n_1, n_2)$  có thể được xác định từ (3.32) bằng cách sử dụng các giá trị  $k$  và  $\sigma^2$  mà ta vừa xác định.

Algorit trên đây được khai triển theo quan niệm là trên toàn bức ảnh được xử lý độ rõ nhiều như nhau, không phụ thuộc vào ảnh chi tiết cục bộ. Tuy vậy, đã không khống chế được một cách rõ ràng mức độ nhòe gây ra. May mắn là trong những vùng nhiễu chi tiết mà ta mong muốn tín hiệu càng ít nhòe càng tốt, thì  $M$  lại lớn. Như vậy  $V(M)$  nhỏ, mức nhiễu  $\sigma_p^2$  còn lại trong ảnh bị xử lý tương đối lớn và sẽ ít nhòe.



**Hình 3.13:** Minh họa hiệu năng về algorit phục hồi ảnh thích nghi dựa vào hàm độ rõ nhiễu.

- (a) Ảnh gốc 512x512 pixel ;
- (b) Ảnh bị xuống cấp bởi nhiễu trắng với  $SNR = 7dB$  Và  $NMSE = 19,8\%$ ;
- (c) Ảnh được xử lý sử dụng hàm rõ nhiễu đạt được từ ảnh gốc với  $NMSE = 3,4\%$  và mức cải thiện  $SNR = 7,7 dB$ ;
- (d) Ảnh được xử lý sử dụng hàm rõ nhiễu đạt được từ ảnh gốc với  $NMSE = 7,0\%$  và mức cải thiện  $SNR = 4,5 dB$ .

Hình 3.13 minh họa hiệu năng của algorit này. Hình 3.13(a) là ảnh gốc 512 x 512 pixels. Hình 3.13(b) là ảnh bị xuống cấp bởi nhiễu Gauss với SNR bằng 7dB. Hình 3.13(c) là ảnh được xử lý, mức cải thiện SNR là 7,7dB. Ảnh đã xử lý nhận được bằng cách cho bộ lọc thích nghi ở từng pixel và xác định hàm che lấp  $M(n_1, n_2)$  từ ảnh gốc (không nhiễu).

Mặc dù đã đặt ra nhiều giả định và lấy xấp xỉ khi khai triển algorit này, nhưng cũng đã làm giảm nhiều đáng kể mà tín hiệu ít bị nhoè. Trong thực tế không có ảnh gốc không nhiều để ước lượng  $M(n_1, n_2)$ . Nếu nhận được  $M(n_1, n_2)$  từ ảnh bị nhiễu thì hiệu năng của algorit này kém đi. Hình 3.13(c) là ảnh đã xử lý bởi algorit ứng với  $M(n_1, n_2)$  nhận được từ ảnh bị xuống cấp, mức cải thiện SNR là 4,5dB. Algorit này là một ví dụ về khai thác hàm rõ nhiễu  $V(M)$ . Còn có nhiều định nghĩa khác của  $V(M)$  và nhiều cách khai thác  $V(M)$  khác được dùng để khai triển algorit phục hồi ảnh.

## 2.6. TRỪ PHỐ TRONG KHÔNG GIAN HỆ

Phương pháp được thảo luận trong tiết này là sự mở rộng trực tiếp của phương pháp đã phát triển để làm giảm nhiễu cộng ngẫu nhiên trong lời nói [Lim]. Vì việc thiết kế và thực hiện bộ lọc biến đổi trong không gian dùng trong phương pháp này chi phí tính toán rất tốn kém, cho nên phải dùng phép xử lý từng ảnh con.

Áp dụng cửa sổ  $w(n_1, n_2)$  cho ảnh bị xuống cấp  $g(n_1, n_2)$ , ta có:

$$g(n_1, n_2)w(n_1, n_2) = f(n_1, n_2)w(n_1, n_2) + v(n_1, n_2)w(n_1, n_2) \quad (3.36)$$

Viết lại (3.36), ta có:

$$g_w(n_1, n_2) = f_w(n_1, n_2) + v_w(n_1, n_2). \quad (3.37)$$

Cửa sổ được chọn sao cho ảnh con  $g_w(n_1, n_2)$  có thể coi là dừng. Với  $G_w(\omega_1, \omega_2)$ ,  $F_w(\omega_1, \omega_2)$  và  $V_w(\omega_1, \omega_2)$  theo thứ tự là biến đổi Fourier của  $g_w(n_1, n_2)$ ,  $f_w(n_1, n_2)$  và  $v_w(n_1, n_2)$ , từ (3.37).

$$\begin{aligned} |G_w(\omega_1, \omega_2)|^2 &= |F_w(\omega_1, \omega_2)|^2 + |V_w(\omega_1, \omega_2)|^2 + F_w(\omega_1, \omega_2)V_w^*(\omega_1, \omega_2) \\ &\quad + F_w^*(\omega_1, \omega_2)V_w(\omega_1, \omega_2) \end{aligned} \quad (3.38)$$

Các hàm  $V_w^*(\omega_1, \omega_2)$  và  $F_w^*(\omega_1, \omega_2)$  là liên hợp phức của  $V_w(\omega_1, \omega_2)$  và  $F_w(\omega_1, \omega_2)$ . Viết lại (3.38), ta nhận được:

$$\begin{aligned} |F_w(\omega_1, \omega_2)|^2 &= |G_w(\omega_1, \omega_2)|^2 - |V_w(\omega_1, \omega_2)|^2 - F_w(\omega_1, \omega_2)V_w^*(\omega_1, \omega_2) \\ &\quad - F_w^*(\omega_1, \omega_2)V_w(\omega_1, \omega_2) \end{aligned} \quad (3.39)$$

trong phương pháp trừ phế, dựa vào (3.39) ước lượng  $|F_w(\omega_1, \omega_2)|$ . Từ ảnh bị xuống cấp  $g_w(n_1, n_2)$ , trực tiếp nhận được  $|G_w(\omega_1, \omega_2)|^2$ . Các số hạng

$|V_w(\omega_1, \omega_2)|^2$ ,  $F_w(\omega_1, \omega_2) V_w^*(\omega_1, \omega_2)$  và  $F_w^*(\omega_1, \omega_2) V_w(\omega_1, \omega_2)$  không thể nhận được chính xác và được lấy xấp xỉ bằng  $E[|V_w(\omega_1, \omega_2)|^2]$ ,  $E[F_w(\omega_1, \omega_2) V_w^*(\omega_1, \omega_2)]$  và  $E[F_w^*(\omega_1, \omega_2) V_w(\omega_1, \omega_2)]$ . Với  $v_w(n_1, n_2)$ , có trung vị bằng 0 và không tương quan với  $f(n_1, n_2)$ ,  $E[F_w(\omega_1, \omega_2) V_w^*(\omega_1, \omega_2)]$  và  $E[F_w^*(\omega_1, \omega_2) V_w(\omega_1, \omega_2)]$  là bằng 0 và (3.39) gợi ý một ước lượng của  $|F_w(\omega_1, \omega_2)|^2$  là:

$$|\hat{F}_w(\omega_1, \omega_2)|^2 = |G_w(\omega_1, \omega_2)|^2 - E[|V_w(\omega_1, \omega_2)|^2] \quad (3.40)$$

trong đó  $E[|V_w(\omega_1, \omega_2)|^2]$  nhận được từ các thuộc tính đã biết hoặc đo được của  $v(n_1, n_2)$ . Giá trị ước lượng  $|\hat{F}_w(\omega_1, \omega_2)|$  trong (3.40) không đảm bảo là không âm. Để đảm bảo  $|\hat{F}_w(\omega_1, \omega_2)|$  không âm, giả thiết  $|\hat{F}_w(\omega_1, \omega_2)| = 0$  nếu  $|G_w(\omega_1, \omega_2)|^2$  nhỏ hơn  $E[|V_w(\omega_1, \omega_2)|^2]$ .

Cho một ước lượng của  $|F_w(\omega_1, \omega_2)|$ , có nhiều cách khác nhau để ước lượng  $f_w(n_1, n_2)$ . Một phương pháp thường dùng và kiên định ý tưởng bộ lọc sai số quân phương tối thiểu, như bộ lọc Wiener, có pha bằng không, và lấy xấp xỉ  $\theta_{fw}(\omega_1, \omega_2)$  bằng  $\theta_{gw}(\omega_1, \omega_2)$ , sao cho:

$$\hat{F}_w(\omega_1, \omega_2) = |\hat{F}_w(\omega_1, \omega_2)| \exp(j\theta_{gw}(\omega_1, \omega_2)) \quad (3.41a)$$

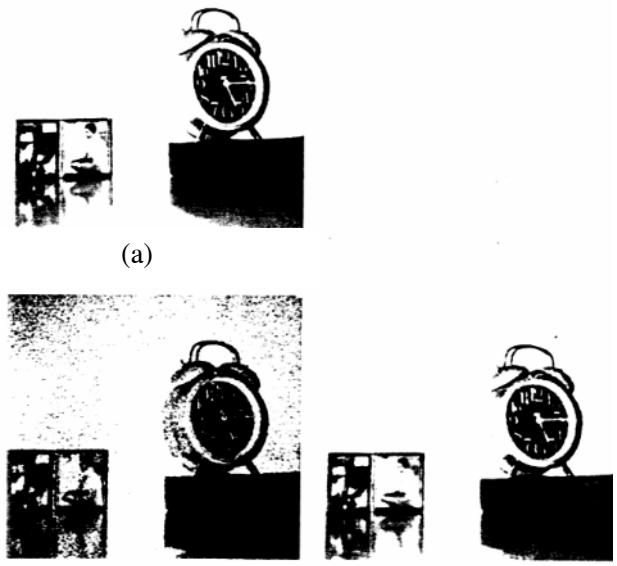
$$\hat{f}_w(\omega_1, \omega_2) = F^{-1}[\hat{F}_w(\omega_1, \omega_2)] \quad (3.41b)$$

Từ (3.40), thấy rõ là có thể ước lượng  $|F_w(\omega_1, \omega_2)|$  bằng cách đem  $|G_w(\omega_1, \omega_2)|^2$  trừ đi số hạng độ lệch  $E[|V_w(\omega_1, \omega_2)|^2]$ . Điều này dẫn đến kỹ thuật trừ phỏ. Vì  $|V_w(\omega_1, \omega_2)|^2$  có thể xem như một periodogram, và vì phương sai của periodogram rất lớn, nên trừ đi  $E[|V_w(\omega_1, \omega_2)|^2]$  cũng không làm giảm nhiễu nền đủ mức. Để làm giảm thêm nhiễu nền và trả giá bằng méo thêm tín hiệu, thường đem trừ đi  $\alpha E[|V_w(\omega_1, \omega_2)|^2]$ , với  $\alpha > 1$ . Trong trường hợp này, ước lượng  $\hat{F}_w(\omega_1, \omega_2)$  nhận được từ:

$$\hat{F}_w(\omega_1, \omega_2) = \begin{cases} \left[ |G_w(\omega_1, \omega_2)|^2 - \alpha E[|V_w(\omega_1, \omega_2)|^2] \right]^{1/2} e^{j\theta_{gw}(\omega_1, \omega_2)}, \\ \quad |G_w(\omega_1, \omega_2)|^2 > \alpha E[|V_w(\omega_1, \omega_2)|^2] \\ 0, \quad \text{trường hợp khác.} \end{cases} \quad (3.42)$$

trong đó  $\alpha$ , là thông số khống chế giảm mức nhiễu. Vì sử dụng (3.42) cho từng ảnh con và các ảnh con đã xử lý được tổ hợp lại thành ảnh toàn bộ, nên phương pháp này được gọi là *trừ phổi không gian hẹp* (short-space spectral subtraction). Khi tín hiệu lời nói được xử lý theo kiểu xử lý từng đoạn, phương pháp này được gọi là xử lý tiếng nói thời gian ngắn (short-time speech processing).

Phương pháp trừ phổi có thể được xem như là cải thiện SNR. Vì từ  $|G_w(\omega_1, \omega_2)|^2$  trừ hàm giống nhau được trong cả vùng chi tiết cao và vùng chi tiết thấp, phép trừ có ảnh hưởng nhỏ trong vùng chi tiết cao ở đó  $|G_w(\omega_1, \omega_2)|^2$  lớn, trong vùng chi tiết thấp, ở đó  $|G_w(\omega_1, \omega_2)|^2$  nhỏ và bao gồm chủ yếu thành phần nhiễu, phép trừ loại bỏ phần lớn của  $|G_w(\omega_1, \omega_2)|^2$ .



**Hình 3.14:** Minh họa hiệu năng của phép trừ phổi không gian hẹp.

- (a) Ảnh gốc 256 x 256 pixels;
- (b) Ảnh bị xuống cấp bởi nhiễu trắng, với SNR = 10 dB;
- (c) Ảnh được xử lý bởi phép trừ phổi không gian hẹp.

Hình 3.14 minh họa hiệu năng của phép trừ phông không gian hẹp. Hình 3.14(a) là ảnh gốc 256 x 256 pixels. Hình 3.14(b) là ảnh bị xuống cấp bởi nhiễu trắng Gausian, khi SNR = 10dB. Hình 3.14(c) là ảnh được xử lý với  $\alpha = 2$  trong (3.42). Sử dụng kích thước ảnh con là 32 x 32 pixels và cửa sổ hình tam giác được chồng lên riêng biệt được sử dụng phương pháp xử lý từng ảnh con.

## 2.7. PHỤC HỒI ẢNH THÍCH NGHI NHẬY BIÊN

Trong ba tiết trước, ta đã thảo luận algorit phục hồi thích nghi, nó thích nghi với đặc tính cục bộ của ảnh. Trong vùng cục bộ thường giả định ảnh là một mẫu của quá trình ngẫu nhiên dừng. Ở vùng biên giả định này có vấn đề, vì ở đó mô hình tín hiệu không còn là một mẫu của quá trình ngẫu nhiên dừng, đâu chỉ là mô hình cục bộ. Bộ lọc dựa vào giả định này sẽ duy trì đường biên nhưng cũng để lại một lượng nhiễu lớn ở lân cận đường biên. Mặc dù trong vùng gần biên nhiễu rõ như ở vùng cường độ đều, nhưng loại bỏ được nhiễu ở gần biên thì vẫn có lợi.

Một cách tiếp cận để giảm nhiễu gần đường biên mà không làm nhoè thêm là dùng một mô hình ảnh chính xác hơn (thí dụ coi biên như một thành phần xác định) và dựa vào đó để khai triển algorit phục hồi ảnh. Tuy nhiên, tìm mô hình chính xác cho ảnh là một việc khó khăn và algorit phục hồi dựa vào mô hình ảnh chi tiết và chính xác thường rất phức tạp. Một cách tiếp cận khác là thoát tiên dùng algorit tách biên đã thảo luận ở tiết 2.3 và sau đó sử dụng đường biên tách được để thiết kế và thực hiện một bộ lọc thích nghi. Chẳng hạn, có thể lấy biên làm đường bao của vùng ảnh cục bộ mà trong đó ảnh được coi là dừng. Khi thiết kế bộ lọc biến đổi không gian  $h(n_1, n_2)$ , vùng chứa đựng  $h(n_1, n_2)$  có thể được chọn sao cho  $h(n_1, n_2)$  không bao phủ những pixels nằm trên nhiều hơn một vùng. Tuy nhiên, cách tiếp cận này yêu cầu phải xác định được đường biên rõ ràng mà tách biên khi có nhiễu tồn tại lại là điều không đơn giản.

Một cách tiếp cận khác là sử dụng một dãy bộ lọc thích nghi 1-D, mà không thay đổi mô hình ảnh hoặc nguyên lý cơ sở dùng trong việc khai triển hệ phục hồi ảnh 2-D. Với bộ lọc 1-D định hướng theo hướng đường biên, đường biên được chừa ra còn ảnh được lọc dọc theo biên. Với các bộ lọc 1-D khác định hướng xuyên qua đường biên thì thực hiện ít việc xử lý và biên được duy trì.

Cho  $T_i[.]$ , ( $1 \leq i \leq N$ ) đại diện cho bộ lọc 1-D, cách thiết kế giống như bộ lọc thích nghi 2-D nhưng được xác định từ một vùng 1-D cục bộ và định hướng theo hướng thứ  $i$ . Trên thực tế,  $N$  thường chọn là 4 và bốn hướng được chọn có góc là  $0^\circ, 45^\circ, 90^\circ$ ,

135°. Ảnh bị xuống cấp được lọc qua một dãy liên tiếp bốn bộ lọc 1 -D, đối với từng bộ lọc này, ảnh giống như là tín hiệu 1 -D. Ảnh đã xử lý  $p(n_1, n_2)$  là:

$$p(n_1, n_2) = T_4[T_3[T_2[T_1[g(n_1, n_2)]]]] \quad (3.43)$$

Toán tử  $T_i[\cdot]$  là biến đổi trong không gian. Vì ảnh lần lượt đi qua một dãy bốn bộ lọc, nên sau mỗi bộ lọc đặc tính tín hiệu và nhiễu thay đổi, phải được cập nhật trước khi đi vào bộ lọc tiếp theo.

Để minh họa phương pháp xử lý 1 -D cho phục hồi ảnh thích nghi này, ta hãy xét ứng dụng của nó vào bộ lọc Wiener thích nghi đã nói đến ở tiết 3.2.4. Các phương trình (3.26), (3.27) và (3.29) định rõ algorit phục hồi. Xét một bộ lọc biến đổi trong không gian 1-D định hướng theo hướng ngang và được thiết kế, thực hiện theo nguyên lý algorit phục hồi ảnh 2-D. Các phương trình về đầu ra của bộ lọc 1-D  $p_1(n_1, n_2)$  là:

$$p_1(n_1, n_2) = m_f(n_1, n_2) + \frac{\sigma_f^2(n_1, n_2)}{\sigma_f^2(n_1, n_2) + \sigma_v^2(n_1, n_2)}(g(n_1, n_2) - m_f(n_1, n_2)). \quad (3.44)$$

$$\hat{m}_f(n_1, n_2) = \frac{I}{(2M+1)} \sum_{k_1=n_1-M}^{n_1+M} g(k_1, n_2). \quad (3.45)$$

và  $\hat{\sigma}_f^2(n_1, n_2) = \begin{cases} \hat{\sigma}_g^2(n_1, n_2) - \sigma_v^2(n_1, n_2), & \text{nếu } \hat{\sigma}_g^2(n_1, n_2) > \sigma_v^2(n_1, n_2) \\ 0, & \text{trường hợp khác.} \end{cases}$  (3.46a)

$$\hat{\sigma}_g^2(n_1, n_2) = \frac{I}{(2M+1)} \sum_{k_1=n_1-M}^{n_1+M} (g(k_1, n_2) - \hat{m}_f(k_1, n_2))^2. \quad (3.46b)$$

Các phương trình (3.44), (3.45) và (3.46) theo thứ tự ứng với các phương trình (3.26), (3.27) và (3.29). Bộ lọc 1-D thứ hai định hướng theo phương dọc và áp dụng cho  $p_1(n_1, n_2)$ , số hạng nhiễu  $\sigma_v^2(n_1, n_1)$  phải được cập nhật, vì bộ lọc đã làm giảm công suất nhiễu. Công suất nhiễu đã giảm có thể tính được từ bộ lọc biến đổi theo không gian 1 -D và  $\sigma_v^2(n_1, n_1)$ . Còn lại hai bộ lọc 1-D định hướng theo hai đường chéo, cách áp dụng cũng tương tự.

Hệ các bộ lọc 1-D nối thành dãy có khả năng thích nghi với định hướng của đường biên ảnh. Đường biên sắc nét và làm thành một góc lớn với hướng bộ lọc thì sẽ được giữ nguyên không đổi. Nếu hướng bộ lọc gần giống hướng đường biên thì nhiễu ở gần biên sẽ được lọc bỏ. Cách tiếp cận này yêu cầu tính toán ít hơn so với algorit phục

hồi 2-D tương ứng. Ngoài ra, cách tiếp cận này có vẻ cải thiện được hiệu năng của một vài algorit phục hồi thích nghi 2 -D.



(a)



(b)

**Hình 3.15:** Minh họa hiệu năng của hệ phục hồi ảnh nhạy biên.



(c)



(d)

(a) Ảnh gốc 256 x 256 pixels;

(b) Ảnh bị xuống cấp bởi nhiễu trắng, với SNR = 6dB và NMSE = 25,1%;

(c) Ảnh được xử lý bởi bộ lọc thích nghi 2 -D, với NMSE = 5,2% và mức cải thiện SNR = 6,8 dB;

(d) Ảnh được xử lý bởi bộ lọc thích nghi 1 -D, với NMSE = 4,7% và mức cải thiện SNR = 7,3dB.

**Hình 3.16:** Những đoạn mở rộng của các ảnh trên hình 3.15.

(a) Ảnh gốc;

(b) Ảnh bị xuống cấp;

(c) Ảnh được xử lý bởi bộ lọc thích nghi 2 -D;

(d) Ảnh được xử lý bởi bộ lọc thích nghi 1 -D;



(a)

(b)



(c)

(d)

Hình 3.15 minh họa hiệu năng của cách tiếp cận 1-D này. Hình 3.15(a) là ảnh gốc 256 x 256 pixels. Hình 3.15(b) là ảnh bị xuống cấp bởi nhiễu trắng Gauss, khi SNR = 6dB. Hình 3.15(c) là ảnh được xử lý bởi bộ lọc Wiener thích nghi 2 -D trong tiết 3.2.4, mức cải thiện SNR của ảnh được xử lý là 6,79 dB. Hình 3.15(d) là ảnh được xử lý bởi bốn tầng bộ lọc biến đổi trong không gian 1 -D được thiết kế trên cơ sở (3.44), (3.45) và (3.46). Mức cải thiện SNR của ảnh được xử lý là 7,28 dB . Hình 3.16 biểu diễn những đoạn mở rộng của các ảnh trên hình 3.15. So sánh các hình 3.16(c) và (d) thấy rằng ảnh ở hình sau nhiễu ít hơn.

Trong tiết này và ba tiết trước ta đã thảo luận về một vài algorit phục hồi ảnh thích nghi. Từ thảo luận trên thấy rằng còn có thể phát triển nhiều thêm algorit khác. Các algorit phục hồi ảnh thích nghi yêu cầu tính toán nhiều hơn algorit không thích nghi, nhưng hiệu quả tốt hơn.

### 3. GIẢM NHOÈ ẢNH

Một ảnh bị xuống cấp vì nhoè có thể mô hình hoá như sau.

$$g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2) \quad (3.47)$$

trong (3.47), ảnh bị xuống cấp  $g(n_1, n_2)$  là kết quả nhân chập ảnh gốc  $f(n_1, n_2)$  với một đáp ứng xung  $b(n_1, n_2)$ . Chuỗi  $b(n_1, n_2)$  được gọi là *hàm phân tán điểm* (point spread function) hoặc hàm nhoè. Sự xuống cấp này có thể được mô hình hoá bằng nhân chập với hàm nhoè do các nguyên nhân như thấu kính lệch tiêu cự, máy bị rung và nhiễu loạn (turbulence) của khí quyển.

Bài toán làm giảm nhoè có thể chia thành hai loại. Loại thứ nhất là giải tích chập (deconvolution), trong đó giả thiết  $b(n_1, n_2)$  đã biết, loại thứ hai là giải tích chập mù (blind deconvolution), trong đó  $b(n_1, n_2)$  là không biết và phải ước lượng từ những thông tin sẵn có. Trong tiết 3.3.1, thảo luận về bộ lọc ngược là một cách tiếp cận chuẩn để giải bài toán giải tích chập. Trong tiết 3.3.2 thảo luận về các algorit để giải bài toán giải tích chập mù.

#### 3.1. BỘ LỌC NGƯỢC

Khi hàm nhoè  $b(n_1, n_2)$  đã biết, một cách tiếp cận để khử nhoè là bộ lọc ngược. Từ (3.47):

$$G(\omega_1, \omega_2) = F(\omega_1, \omega_2) B(\omega_1, \omega_2) \quad (3.48)$$

Từ (3.48)

$$F(\omega_1, \omega_2) = \frac{G(\omega_1, \omega_2)}{B(\omega_1, \omega_2)}. \quad (3.49)$$

Theo (3.49), một hệ khôi phục được  $f(n_1, n_2)$  từ  $g(n_1, n_2)$  là một bộ lọc ngược, biểu diễn trên hình 3.17.

Bộ lọc ngược trong hình 3.17 có khuynh hướng rất nhạy cảm với nhiễu. Khi  $B(\omega_1, \omega_2)$  rất nhỏ,  $1/B(\omega_1, \omega_2)$  rất lớn, và trong vùng tần số mà  $1/B(\omega_1, \omega_2)$  rất lớn nhiều nhỏ cũng nổi lên. Một phương pháp làm giảm vấn đề nhạy cảm với nhiễu là giới hạn đáp ứng tần số  $1/B(\omega_1, \omega_2)$  ở một ngưỡng  $\gamma$  như sau.

$$H(\omega_1, \omega_2) = \begin{cases} \frac{1}{B(\omega_1, \omega_2)}, & \text{nếu } \left| \frac{1}{B(\omega_1, \omega_2)} \right| < \gamma \\ \gamma \frac{|B(\omega_1, \omega_2)|}{B(\omega_1, \omega_2)}, & \text{trường hợp khác.} \end{cases} \quad (3.50)$$

Bộ lọc ngược  $1/B(\omega_1, \omega_2)$  và biến thể của nó trong (3.50) có thể được thực hiện bằng nhiều cách. Ta có thể thiết kế một bộ lọc mà đáp ứng tần số gần như mong đợi bằng kỹ thuật thiết kế bộ lọc đã thảo luận trước đây, và sau đó nhân chập ảnh nhoè với bộ lọc được thiết kế. Nói cách khác, ta có thể thực hiện hệ bằng cách sử dụng biến đổi DFT và biến đổi ngược IDFT theo cách tương tự như với bộ lọc Wiener đã thảo luận trong tiết 3.2.1

Một phương pháp khác thực hiện bộ lọc ngược là sử dụng quá trình lặp ở đó ước lượng tín hiệu  $f(n_1, n_2)$  được cập nhật sau mỗi lần lặp. Gọi  $\hat{f}_k(n_1, n_2)$  là ước lượng của tín hiệu sau  $k$  lần lặp. Nếu  $\hat{f}_k(n_1, n_2)$  là một ước lượng tốt của  $f(n_1, n_2)$ , thì  $\hat{f}_k(n_1, n_2) * b(n_1, n_2)$  sẽ rất gần  $g(n_1, n_2)$ . Ước lượng tín hiệu sau  $k + 1$  lần lặp  $\hat{f}_{k+1}(n_1, n_2)$  nhận được bằng cách cộng thêm vào  $\hat{f}_k(n_1, n_2)$  số hạng hiệu chỉnh gồm hằng số tỉ lệ  $\lambda$  nhân với hiệu giữa  $g(n_1, n_2)$  và  $\hat{f}_k(n_1, n_2) * b(n_1, n_2)$ . Sử dụng  $\lambda g(n_1, n_2)$  như là ước lượng ban đầu của  $\hat{f}_0(n_1, n_2)$ , quy trình lặp là:

$$\hat{f}_0(n_1, n_2) = \lambda g(n_1, n_2) \quad (3.51a)$$

$$\hat{f}_{k+1}(n_1, n_2) = \hat{f}_k(n_1, n_2) + \lambda(g(n_1, n_2) - \hat{f}_k(n_1, n_2) * b(n_1, n_2)) \quad (3.51b)$$

Trong đó  $\lambda$  là một thông số dương và có thể dùng để điều khiển sự hội tụ của quy trình lặp.

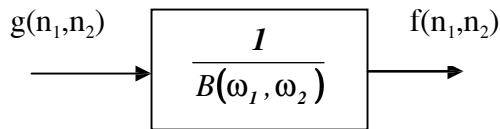
Để xem (3.51) liên quan tới bộ lọc ngược như thế nào, ta biểu diễn (3.51) trong miền tần số.

$$\hat{F}_\theta(n_1, n_2) = \lambda G(\omega_1, \omega_2) \quad (3.52a)$$

$$\hat{F}_{k+1}(\omega_1, \omega_2) = \hat{F}_k(\omega_1, \omega_2) + \lambda(G(\omega_1, \omega_2) - \hat{F}_k(\omega_1, \omega_2)B(\omega_1, \omega_2)) \quad (3.52b)$$

Giải phương trình (3.52) bằng phương pháp đệ quy, ta đạt được.

$$\begin{aligned} \hat{F}_k(\omega_1, \omega_2) &= \lambda G(\omega_1, \omega_2)[1 + (1 - \lambda B(\omega_1, \omega_2)) + \dots + (1 - \lambda B(\omega_1, \omega_2))^k] \\ &= \frac{G(\omega_1, \omega_2)}{B(\omega_1, \omega_2)} [I - (I - \lambda B(\omega_1, \omega_2))^{k+1}] \end{aligned} \quad (3.53)$$



**Hình 9.17:** Bộ lọc ngược cho phục hồi ảnh.

Từ (3.53), khi k tiến tới  $\infty$ ,  $\hat{F}_k(\omega_1, \omega_2)$  tiến tới  $G(\omega_1, \omega_2)/B(\omega_1, \omega_2)$ , nó là kết quả của phép lọc ngược, với điều kiện là:

$$|(1 - \lambda B(\omega_1, \omega_2))| < 1 \quad (3.54)$$

Trong phạm vi mà thông số  $\lambda$  thoả mãn (3.54), phương trình (3.51) có thể dùng để thực hiện bộ lọc ngược. Một ưu điểm của quy trình lặp là nó có thể dừng lại sau một số hữu hạn bước lặp. Kết quả nhận được sau một số hữu hạn bước lặp không hoàn toàn giống như bộ lọc ngược, nhưng trong một số trường hợp, nó kém nhậy cảm hơn với nhiễu.

Hình 3.18 minh họa hiệu năng của bộ lọc ngược. Hình 3.18(a) là ảnh gốc 512 x 512 pixels. Hình 3.18(b) là ảnh gốc bị nhòe bởi hàm nhòe dạng -Gauss. kích thước của ảnh kết quả lớn hơn 512 x 512 pixels, nhưng được lồng trong cửa sổ hình chữ nhật 512 x 512 điểm. Mô hình ảnh bị xuống cấp trong trường hợp này là:

$$g(n_1, n_2) = [f(n_1, n_2) * b(n_1, n_2)]w(n_1, n_2). \quad (3.55)$$

Hình 3.18(c) là ảnh đã xử lý bằng bộ lọc ngược. Ảnh đã xử lý  $p(n_1, n_2)$  được tính theo:

$$p(n_1, n_2) = \text{IDFT}[G(k_1, k_2)H(k_1, k_2)] \quad (3.56)$$

trong đó  $G(k_1, k_2)$  là biến đổi Fourier của  $g(n_1, n_2)$  và  $H(k_1, k_2)$  nhận được từ:

$$H(k_1, k_2) = \frac{1}{B(\omega_1, \omega_2)} \Big|_{\omega_1=2\pi k_1/N, \omega_2=2\pi k_2/N} \quad (3.57)$$

Kích thước DFT và IDFT sử dụng là 512 x 512. Trong trường hợp không có nhiễu và  $B(\omega_1, \omega_2)$  rất nhỏ, thì bộ lọc ngược làm việc rất tốt mặc dù  $g(n_1, n_2)$  trong (3.55) bị cửa sổ ảnh hưởng.



(a)

**Hình 3.18:**

- (a) Ảnh gốc 512x512 pixel ;
- (b) Ảnh bị nhòe bởi hàm nhòe dạng -Gauss;
- (c) Kết quả của bộ lọc ngược.



(b)



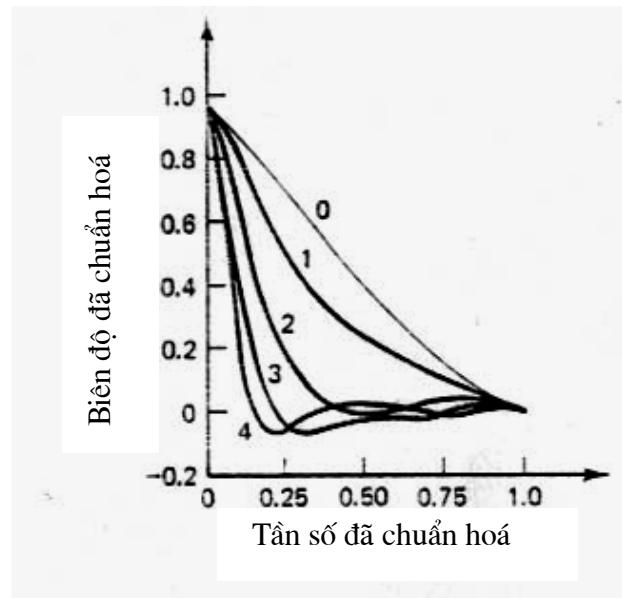
(c)

### 3.2. ALGORIT CHIA CHẬP MÙ

Nếu hàm nhòe  $b(n_1, n_2)$  không biết chính xác, phải ước lượng  $b(n_1, n_2)$  trước khi đưa tới bộ lọc ngược. Vì ta muốn chia chập  $g(n_1, n_2)$  khi không có hiểu biết chi tiết về  $b(n_1, n_2)$ , nên phép xử lý này được gọi là bài toán *chia chập mù*.

Nếu ta chẳng biết về  $f(n_1, n_2)$  hoặc  $b(n_1, n_2)$ , thì không thể giải bài toán chia chập mù. Bài toán này cũng giống như phải tìm hai số từ tổng của chúng khi không biết gì về một trong hai số đó. Để giải quyết vấn đề chia chập mù, phải biết một vài thông

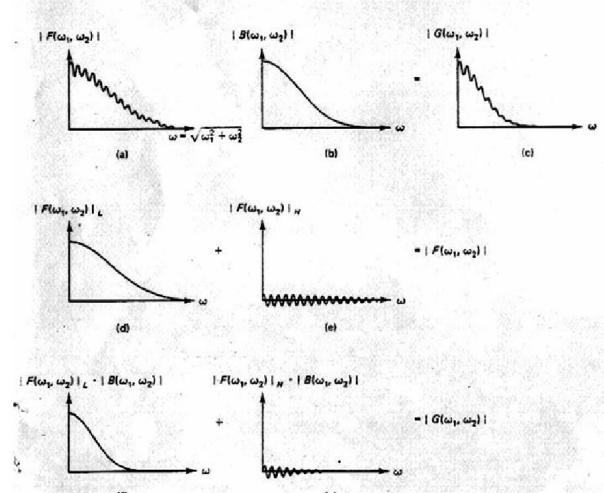
tin nào đó về  $f(n_1, n_2)$ ,  $b(n_1, n_2)$  hoặc cả hai. Các algorit chia chập mù khác nhau về giả thiết đã biết cái gì và sự hiểu biết đó được khai thác như thế nào.



**Hình 3.19:** Hàm truyền điều chế cho một thấu kính tròn mỏng, là một hàm của mức độ lệch tiêu cự. Số lớn tương ứng với mức độ lệch tiêu cự lớn.

Giả sử  $f(n_1, n_2)$  và  $b(n_1, n_2)$  là các dãy mở rộng hữu hạn với các hàm biến đổi  $z$  không nhân tử hóa (nonfactorable)  $F(z_1, z_2)$  và  $B(z_1, z_2)$ . Ta có thể từ  $g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$  phục hồi  $f(n_1, n_2)$  bằng một algorit nhân tử hóa (factorization) đa thức, kết quả nhận được chỉ chênh lệch một phép tịnh tiến và một hệ số tỉ lệ.  $G(z_1, z_2)$  là biến đổi  $z$  của  $g(n_1, n_2)$ , nhận được từ  $G(z_1, z_2) = F(z_1, z_2) B(z_1, z_2)$ . Vì ta giả thiết rằng  $f(n_1, n_2)$  và  $b(n_1, n_2)$  là những dãy mở rộng hữu hạn,  $G(z_1, z_2)$  là một đa thức 2-D có bậc hữu hạn trong  $z_1^{-1}$  và  $z_2^{-1}$ . Ngoài ra ta giả thiết rằng  $F(z_1, z_2)$  và  $B(z_1, z_2)$  là không nhân tử hóa (không phân tích thành thừa số nhân được, - nonfactorable) và do đó những nhân tử không tầm thường (nontrivial) của  $G(z_1, z_2)$  là  $F(z_1, z_2)$  và  $B(z_1, z_2)$ . Các algorit nhân tử hóa (factorization) đa thức định ra các nhân tử không tầm thường (nontrivial) của  $G(z_1, z_2)$  tồn tại [Iralevitz và Lim] và có thể được sử dụng trong việc xác định  $F(z_1, z_2)$  hoặc  $f(n_1, n_2)$ , kết quả nhận được chỉ chênh lệch một phép tịnh tiến và một hệ số tỉ lệ. Đáng tiếc là, cách tiếp cận để giải quyết vấn đề chia chập mù đó trong thực tế có nhiều khó

khăn. Cho đến ngày nay, các algorit đ được phát triển để nhân tử hoá (factorization) đa thức đều đòi hỏi chi phí tính toán rất cao. Ngoài ra các algorit rất nhạy cảm với bất kỳ sự vi phạm nào đến giả thiết  $G(z_1, z_2) = F(z_1, z_2) B(z_1, z_2)$ , hoặc  $g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$ . Trong thực tiễn mô hình tích chập  $g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$  không hoàn toàn chính xác do sự tồn tại nhiều nền hoặc là do những phép xấp xỉ khi xây dựng mô hình.



**Hình 3.20:** Khai triển về một phương pháp chia chập mù.

Một algorit chia chập mù có tính thực tiễn dựa vào giả thiết  $|B(\omega_1, \omega_2)|$  là một hàm tròn. Trong một số ứng dụng giả thiết xấp xỉ này có ích. Khi ảnh bị nhoè bởi một thấu kính tròn mỏng, hàm truyền đạt điều chế  $|H(\Omega_x, \Omega_y)|$  là bộ lọc thông thấp đối xứng tròn (circularly symmetric) biểu diễn trên hình 3.19. Khi ảnh bị mờ bởi nhiễu loạn khí quyển, hàm nhoè  $b(x,y)$  và biến đổi Fourier  $B(\Omega_x, \Omega_y)$  của nó gần có dạng Gauss. Khi ảnh bị nhoè bởi chuyển động phương ngang,  $B(\Omega_x, \Omega_y)$  là hàm sinc [phương trình (3.5)] và  $|B(\Omega_x, \Omega_y)|$  là hàm tròn, ngoại trừ tại những vùng ở đó  $B(\Omega_x, \Omega_y)$  đi qua điểm không.

Để ước lượng  $|B(\omega_1, \omega_2)|$  với giả thiết  $|B(\omega_1, \omega_2)|$  là hàm tròn, trước tiên ta chú ý rằng.

$$|G(\omega_1, \omega_2)| = |F(\omega_1, \omega_2)| \cdot |B(\omega_1, \omega_2)| \quad (3.58)$$

Các ví dụ về  $|G(\omega_1, \omega_2)|$ ,  $|F(\omega_1, \omega_2)|$  và  $|B(\omega_1, \omega_2)|$  được biểu diễn trên hình 3.20(a), (b) và (c). Hàm  $|F(\omega_1, \omega_2)|$  có thể coi như tổng của hai phần hợp thành, một hàm trơn ký hiệu là  $|F(\omega_1, \omega_2)|_L$ , và một hàm biến thiên nhanh ký hiệu là  $|F(\omega_1, \omega_2)|_H$ .

$$|F(\omega_1, \omega_2)| = |F(\omega_1, \omega_2)|_L + |F(\omega_1, \omega_2)|_H \quad (3.59)$$

Các hàm  $|F(\omega_1, \omega_2)|_L$ ,  $|F(\omega_1, \omega_2)|_H$  cho hàm  $|F(\omega_1, \omega_2)|$  trong hình 3.20(a) là hình 3.20(d) và (e), từ (3.58) và (3.59) ta có.

$$|F(\omega_1, \omega_2)|_L \cdot |B(\omega_1, \omega_2)| + |F(\omega_1, \omega_2)|_H \cdot |B(\omega_1, \omega_2)| = |G(\omega_1, \omega_2)| \quad (3.60)$$

Hàm  $|F(\omega_1, \omega_2)|_L \cdot |B(\omega_1, \omega_2)|$  và  $|F(\omega_1, \omega_2)|_H \cdot |B(\omega_1, \omega_2)|$  là hình 3.20(f) và (g), giả sử ta áp dụng toán tử tròn S tới (3.60), giả sử toán tử tròn là tuyến tính, ta có .

$$S[|F(\omega_1, \omega_2)|_L \cdot |B(\omega_1, \omega_2)|] + S[|F(\omega_1, \omega_2)|_H \cdot |B(\omega_1, \omega_2)|] = S[|G(\omega_1, \omega_2)|] \quad (3.61)$$

Vì cả  $|F(\omega_1, \omega_2)|_L$  và  $|B(\omega_1, \omega_2)|$  là các hàm trơn, nên việc làm tròn sẽ không ảnh hưởng đáng kể đến  $|F(\omega_1, \omega_2)|_L \cdot |B(\omega_1, \omega_2)|$ . Tuy nhiên, từ hình 3.20(g), làm tròn sẽ làm giảm đáng kể  $|F(\omega_1, \omega_2)|_H \cdot |B(\omega_1, \omega_2)|$ . Dựa vào nhận xét này có thể rút gọn (3.61) thành:

$$|F(\omega_1, \omega_2)|_L \cdot |B(\omega_1, \omega_2)| \approx S[|G(\omega_1, \omega_2)|] \quad (3.62)$$

Từ (3.62),

$$|B(\omega_1, \omega_2)| = \frac{S[|G(\omega_1, \omega_2)|]}{|F(\omega_1, \omega_2)|_L} \quad (3.63)$$

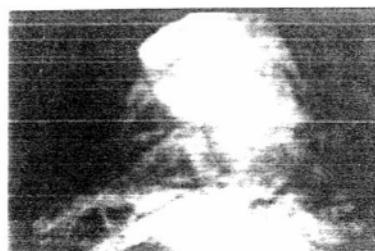
phương trình (3.63) là cơ sở để ước lượng  $|B(\omega_1, \omega_2)|$ . Tử số  $S[|G(\omega_1, \omega_2)|]$  có thể xác định từ ảnh nhoè  $g(n_1, n_2)$ . Mẫu số  $|F(\omega_1, \omega_2)|_L$  được ước lượng từ nhận xét theo kinh nghiệm rằng, đối với những lớp ảnh tương tự  $|F(\omega_1, \omega_2)|_L$  xấp xỉ như nhau. Sự khác nhau về chi tiết ảnh chỉ ảnh hưởng tới  $|F(\omega_1, \omega_2)|_H$ , nhưng không ảnh hưởng đáng kể đến  $|F(\omega_1, \omega_2)|_L$ . Dựa vào điều này ta có:

$$|F(\omega_1, \omega_2)|_L = |F'(\omega_1, \omega_2)|_L \quad (3.64)$$

trong đó  $|F'(\omega_1, \omega_2)|_L$  nhận được từ một ảnh gốc không bị xuống cấp, nội dung tương tự  $f(n_1, n_2)$ . Từ (3.63) và (3.64),

$$|\hat{B}(\omega_1, \omega_2)| = \frac{S[G(\omega_1, \omega_2)]}{|F'(\omega_1, \omega_2)|_L} \quad (3.65)$$

Mặc dù khi suy diễn (3.65) có đặt ra những giả định khác nhau và có dùng phương pháp kinh nghiệm (heuristic) nhưng từ (3.65) vẫn có thể tính ra một ước lượng hợp lý của  $|B(\omega_1, \omega_2)|$ . Cũng có thể nhận được từ (3.65) biểu thức về pha  $\theta_b(\omega_1, \omega_2)$  theo kiểu tương tự như đã tính  $|\hat{B}(\omega_1, \omega_2)|$ . Tuy nhiên, cách tiếp cận này chưa thành công, một phần bởi vì chi tiết ảnh có ảnh hưởng đến hàm pha, và pha của ảnh này chưa đựng rất ít thông tin về pha của ảnh khác, ngay cả khi hai ảnh có nội dung như nhau. Trong khi không có phương pháp tốt để ước lượng  $\theta_b(\omega_1, \omega_2)$ , ta giả định  $\theta_b(\omega_1, \omega_2) = 0$ . Khi đã ước lượng  $B(\omega_1, \omega_2)$  được, có thể dùng bộ lọc ngược và các biến thể của nó đã thảo luận trong tiết 3.3.1, để làm giảm nhoè.



(a)



(b)

**Hình 3.21:** Minh họa hiệu năng của phương pháp chia chập mù.

- (a) Ảnh gốc 512x340 pixel bị nhoè;
- (b) Ảnh đã xử lý.

Hình 3.21 minh họa hiệu năng của phương pháp chia chập mù. Hình 3.21(a) là ảnh bị nhoè của “Titanic” nhận được bằng một máy quay phim dưới nước. Hình 3.21(b) là ảnh đã xử lý. Vì nhoè trong ví dụ này là thực (không phải là do tổng hợp), nên không có sẵn ảnh gốc không nhoè. Trong xử lý ảnh, hàm nhoè  $B(\omega_1, \omega_2)$  đã được ước lượng bằng một phương pháp biến thể của chia chập mù đã thảo luận ở trên. Thuật toán lọc ngược được thực hiện bằng quy trình lặp của (3.51). So sánh ảnh không được xử lý và ảnh đã xử lý thấy rằng đã giảm nhoè một cách đáng kể, nhưng nhiều nền lại tôn lên. Tiết 3.4 sẽ thảo luận về cách làm giảm cả nhoè ảnh và nhiễu nền.

Có thể khai triển một phương pháp chia chập mù khác với giả thiết randle g kích thước hiệu dụng của  $b(n_1, n_2)$  là nhỏ hơn  $f(n_1, n_2)$ . Trong phương pháp này, ảnh bị xuống cấp  $g(n_1, n_2)$  được chia đoạn thành nhiều ảnh con  $g_{ij}(n_1, n_2)$  bằng cách sử dụng những cửa sổ hình chữ nhật gối mép nhau. Kích thước cửa sổ được chọn lớn hơn kích thước hiệu dụng của  $b(n_1, n_2)$  nhưng nhỏ hơn kích thước của  $f(n_1, n_2)$  nhiều. Khi đó ảnh con  $g_{ij}(n_1, n_2)$  được coi như là:

$$g_{ij}(n_1, n_2) \approx f_{ij}(n_1, n_2) * b(n_1, n_2) \quad (3.66)$$

trong đó  $f_{ij}(n_1, n_2)$  là bộ phận ảnh gốc trong cửa sổ đã dùng để nhận được  $g_{ij}(n_1, n_2)$ . Ở xa đường biên phương trình (3.66) là một cách biểu diễn tốt của ảnh con, nhưng tại khu vực gần đường biên nó không còn chính xác. Từ (3.66),

$$|G_{ij}(\omega_1, \omega_2)| \approx |F_{ij}(\omega_1, \omega_2)| |B(\omega_1, \omega_2)| \quad (3.67)$$

Lấy tổng cả hai vế của (3.67) trên toàn bộ ảnh con và viết lại biểu thức là:

$$|B(\omega_1, \omega_2)| = \frac{\sum_i \sum_j |G_{ij}(\omega_1, \omega_2)|}{\sum_i \sum_j |F_{ij}(\omega_1, \omega_2)|}. \quad (3.68)$$

Phương trình (3.68) là cơ sở để ước lượng  $|B(\omega_1, \omega_2)|$ . Số hạng tử số  $\sum_i \sum_j |G_{ij}(\omega_1, \omega_2)|$  đạt được từ  $g(n_1, n_2)$ . Số hạng mẫu số có thể được ước lượng theo nhận xét dựa trên kinh nghiệm là .

$$\sum_i \sum_j |F_{ij}(\omega_1, \omega_2)| = \sum_i \sum_j |F'_{ij}(\omega_1, \omega_2)| \quad (3.69)$$

trong đó  $F'_{ij}(\omega_1, \omega_2)$  nhận được từ một ảnh không bị xuống cấp, nội dung tương tự  $f(n_1, n_2)$ . Từ (3.68) và (3.69), tính ra ước lượng của  $|B(\omega_1, \omega_2)|$  là:

$$|\hat{B}(\omega_1, \omega_2)| = \frac{\sum_i \sum_j |G_{ij}(\omega_1, \omega_2)|}{\sum_i \sum_j |F'_{ij}(\omega_1, \omega_2)|}. \quad (3.70)$$

Hiệu năng của phương pháp chia chập mù dựa vào (3.70) cũng giống như phương pháp dựa vào (3.65).

#### 4. LÀM GIẢM NHOÈ VÀ NHIỄU CỘNG NGẦU NHIÊN

Trong hai tiết trước, ta khai triển riêng biệt các algorit giảm nhoè và giảm nhiễu cộng ngẫu nhiên. Trong thực tế, một ảnh có thể bị xuống cấp bởi cả nhoè và nhiễu cộng ngẫu nhiên:

$$g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2) + v(n_1, n_2) \quad (3.71)$$

điều này biểu diễn trên trong hình 3.22.

Từ (3.71) một cách tiếp cận hợp lý để phục hồi ảnh  $f(n_1, n_2)$  là áp dụng một hệ làm giảm nhiễu để từ  $g(n_1, n_2)$  ước lượng  $r(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$  và sau đó áp dụng một hệ khử nhoè để từ  $r(n_1, n_2)$  ước lượng  $f(n_1, n_2)$ , như hình 3.23.

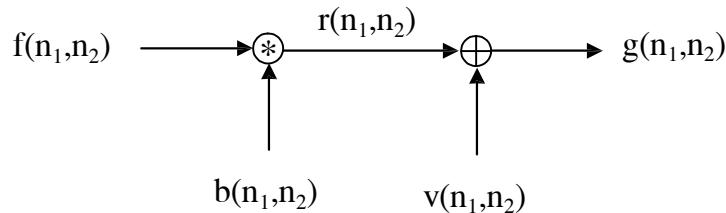
Cách tiếp cận lân lượt khử các loại xuống cấp từng cái một cho phép chúng ta khai triển những algorit phục hồi riêng cho mỗi loại, sau đó cứ kết hợp chúng lại một cách đơn giản nếu ảnh bị xuống cấp vì nhiều loại nguyên nhân khác nhau. Ngoài ra, trong một vài trường hợp đó cũng là cách tiếp cận tối ưu. Chẳng hạn, giả thiết  $f(n_1, n_2)$  và  $v(n_1, n_2)$  là những mẫu độc lập tuyến tính của quá trình n gẫu nhiên dừng có trung vị bằng không. Ngoài ra, giả thiết  $b(n_1, n_2)$  đã biết. Vậy thì bộ ước lượng tuyến tính tối ưu (optimal linear estimator) có thể tối thiểu hoá  $E \left[ (f(n_1, n_2) - \hat{f}(n_1, n_2))^2 \right]$  là một hệ LSI mà đáp ứng tần số  $H(\omega_1, \omega_2)$  là:

$$H(\omega_1, \omega_2) = \frac{P_f(\omega_1, \omega_2) B^*(\omega_1, \omega_2)}{P_f(\omega_1, \omega_2) B(\omega_1, \omega_2)^2 + P_v(\omega_1, \omega_2)} \quad (3.72)$$

Biểu thức (3.72) có thể suy diễn theo cách tương tự như đã suy diễn bộ lọc Wiener ở tiết 6.1.4 . Phương trình (3.72) có thể biểu diễn là:

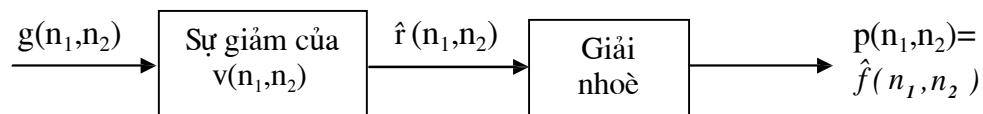
$$\begin{aligned}
 H(\omega_1, \omega_2) &= \frac{P_f(\omega_1, \omega_2) |B(\omega_1, \omega_2)|^2}{P_r(\omega_1, \omega_2) |B(\omega_1, \omega_2)|^2 + P_v(\omega_1, \omega_2)} \frac{I}{B(\omega_1, \omega_2)} \\
 &= \frac{P_r(\omega_1, \omega_2)}{P_r(\omega_1, \omega_2) |B(\omega_1, \omega_2)|^2 + P_v(\omega_1, \omega_2)} \frac{I}{B(\omega_1, \omega_2)}
 \end{aligned} \quad (3.73)$$

biểu thức  $P_r(\omega_1, \omega_2) / (P_r(\omega_1, \omega_2) + P_v(\omega_1, \omega_2))$  là một hệ làm giảm nhiễu bằng phép lọc Wiener. Hệ này ước lượng  $r(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2)$  từ  $g(n_1, n_2)$ .

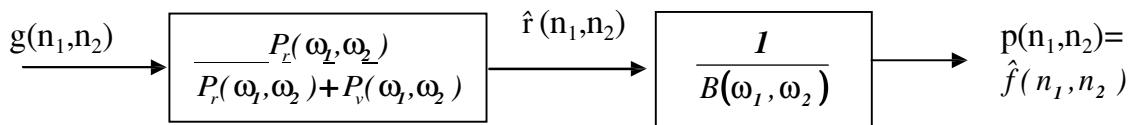


**Hình 3.22:** Mô hình của ảnh bị xuống cấp bởi nhoè và nhiễu cộng ngẫu nhiên.

Biểu thức 1/  $B(\omega_1, \omega_2)$  là bộ lọc ngược, nó ước lượng  $f(n_1, n_2)$  từ giá trị ước lượng  $\hat{r}(n_1, n_2)$ . Như vậy, có thể kết luận rằng hệ toàn thể gồm một hệ làm nhiễu và một hệ khử nhoè nối tiếp nhau. Điều này biểu diễn trên hình 3.24.



**Hình 9.23:** Sự làm giảm nhoè và nhiễu cộng ngẫu nhiên bởi một hệ làm nhiễu và một hệ giải nhoè nối tiếp nhau.

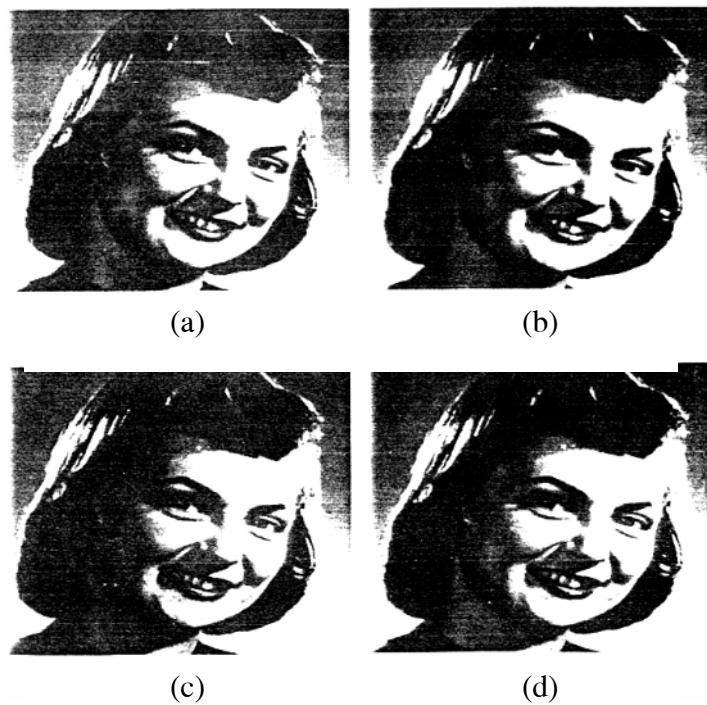


**Hình 3.24:** Một dãy nối tiếp nhau gồm bộ lọc Wiener làm giảm nhiễu cộng ngẫu nhiên và bộ lọc ngược làm giảm nhoè.

Hình 3.25 minh họa hiệu năng của hệ phục hồi ảnh khi ảnh bị xuống cấp bởi nhoè và nhiễu cộng. Hình 3.25(a) là một ảnh gốc 512 x 512 pixels. Hình 3.25(b) là ảnh bị làm nhoè bởi một hàm nhoè dạng Gauss và sau đó lại bị xuống cấp bởi nhiễu Gauss trắng ở mức SNR là 25 dB. Hình 3.25(c) là ảnh được xử lý bởi hệ làm giảm nhiễu được thảo luận trong tiết 2.4 sau đó đem lọc ngược. Hình 3.25(d) cho kết quả khi chỉ lọc ngược mà không làm giảm nhiễu. Rõ ràng là bộ lọc ngược rất nhạy cảm với nhiễu, như đã thảo luận trong tiết 3.1.

**Hình 3.25 :**

- (a) Ảnh gốc 512x512 pixel;
- (b) Ảnh bị xử lý bởi nhoè và nhiễu cộng ngẫu nhiên với SNR= 25dB;
- (c) Ảnh được xử lý bởi một bộ lọc Wiener thích nghi cho việc làm giảm nhiễu cộng ngẫu nhiên và bộ lọc ngược cho việc làm giảm nhoè;
- (d) Ảnh được xử lý bởi một bộ lọc ngược đơn.



Trong xử lý ảnh, quan trọng là giữ kết quả trung gian với độ chính xác cao để tránh ảnh hưởng có hại của lượng tử hoá. Bản thân một lượng nhỏ nhiễu lượng tử có thể không hiện rõ, nhưng đến phép xử lý sau nó sẽ được khuếch đại lên. Chẳng hạn, đem xử lý ảnh xuống cấp bởi nhoè và nhiễu cộng ngẫu nhiên bằng một hệ giảm nhiễu và một hệ khử nhoè đặt nối tiếp. Nếu đâu ra của hệ giảm nhiễu được lượng tử hoá 8 bit /pixel, nhiễu lượng tử sẽ không hiển thị ở tầng này, nhưng đến hệ giải nhoè kề theo sau đó, - thường là một bộ lọc thông cao, có thể khuếch đại nhiễu lưỡng tử và trong kết quả

cuối cùng nhiều có thể hiện lên rất rõ. Vì ảnh hưởng của lượng tử hoá trung gian đến kết quả cuối cùng thường không phân tích minh bạch ra được, cho nên cần lưu trữ kết quả trung gian với độ chính xác cao.

Trong tiết này, ta đã thảo luận vấn đề phục hồi ảnh khi ảnh bị xuống cấp bởi hai loại nguyên nhân. Ý tưởng lần lượt khử các loại xuống cấp từng cái một có thể áp dụng với nhiều loại xuống cấp khác. Cụ thể là, khi một ảnh bị xuống cấp bởi nguyên nhân 1, tiếp theo là nguyên nhân 2, sau đó lại nguyên nhân 3. Có một cách tiếp cận để xét là làm giảm xuống cấp 3 trước, tiếp đến xuống cấp 2 và sau cùng là xuống cấp 1. Một khi hệ toàn thể bao gồm nhiều hệ con được khai triển, có thể làm cho hiệu suất tính toán cao hơn bằng cách sự sắp xếp lại các hệ con. Như cách tiếp cận trên, tuy không phải bao giờ cũng tối ưu, nhưng thường làm cho bài toán phục hồi đơn giản hơn và trong một vài trường hợp đó là cách tiếp cận tối ưu dẫn đến những kết quả giống như xử lý đồng thời các sự xuống cấp.

## 5. LÀM GIẢM NHIỀU PHỤ THUỘC TÍN HIỆU

Một ảnh bị xuống cấp  $g(n_1, n_2)$  bất kỳ có thể được biểu diễn là:

$$g(n_1, n_2) = D[f(n_1, n_2)] = f(n_1, n_2) + d(n_1, n_2) \quad (3.74a)$$

$$\text{Trong đó} \quad d(n_1, n_2) = g(n_1, n_2) - f(n_1, n_2) \quad (3.74b)$$

và  $D[.]$  là một toán tử xuống cấp áp dụng vào  $f(n_1, n_2)$ . Nếu  $d(n_1, n_2)$  không là hàm của tín hiệu  $f(n_1, n_2)$  thì  $d(n_1, n_2)$  được gọi là *nhiều cộng không phụ thuộc tín hiệu*. Nếu  $d(n_1, n_2)$  là hàm của tín hiệu  $f(n_1, n_2)$  thì  $d(n_1, n_2)$  được gọi là *nhiều cộng phụ thuộc tín hiệu*. Những ví dụ về nhiều phụ thuộc tín hiệu là nhiều đốm, nhiều hạt trên phim (film noise grain) và nhiều lượng tử. Một cách tiếp cận để làm giảm nhiều phụ thuộc tín hiệu là biến đổi  $g(n_1, n_2)$  vào một miền, ở đó nhiều trở thành nhiều cộng không phụ thuộc tín hiệu và sau đó làm giảm nhiều không phụ thuộc tín hiệu. Một cách tiếp cận khác là làm giảm nhiều trực tiếp trong miền tín hiệu. Các cách tiếp cận này được thảo luận trong hai tiết sau.

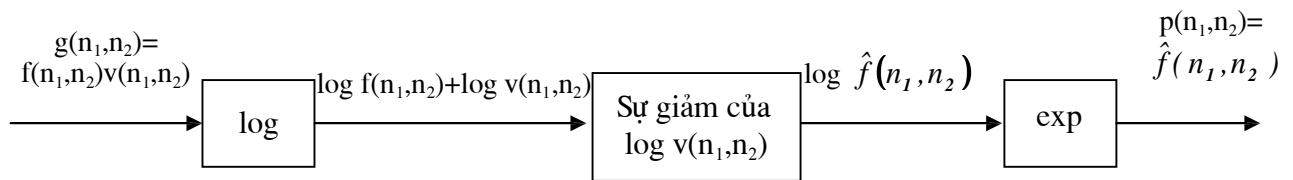
### 5.1. BIẾN ĐỔI THÀNH NHIỀU CỘNG KHÔNG PHỤ THUỘC TÍN HIỆU

Giả sử ta có thể tìm được một toán tử  $T$  sao cho khi áp dụng vào  $g(n_1, n_2)$  trong (3.74a),  $T[g(n_1, n_2)]$  được biểu diễn là:

$$T[g(n_1, n_2)] = T[f(n_1, n_2) + d(n_1, n_2)]$$

$$= T_1[f(n_1, n_2)] + v(n_1, n_2) \quad (3.75)$$

trong đó  $T_1[\cdot]$  là một toán tử, có thể khác với  $T[\cdot]$  và  $v(n_1, n_2)$  là một nhiễu cộng không phụ thuộc tín hiệu. Một cách tiếp cận để phục hồi ảnh  $f(n_1, n_2)$  từ  $g(n_1, n_2)$  là thoát tiên ước lượng  $T_1[f(n_1, n_2)]$  bằng cách làm giảm nhiễu cộng không phụ thuộc tín hiệu  $v(n_1, n_2)$  và sau đó ước lượng  $f(n_1, n_2)$  từ  $T_1[f(n_1, n_2)]$ . Cách tiếp cận này dựa trên thực tế là *làm giảm nhiễu cộng không phụ thuộc tín hiệu nói chung đơn giản hơn làm giảm nhiễu phụ thuộc tín hiệu*, và một số algorit để làm giảm nhiễu cộng không phụ thuộc tín hiệu đã được khai triển sẵn.



**Hình 9.26:** Hệ làm giảm nhiễu nhân.

Để minh họa cách tiếp cận này, ta xét vấn đề làm giảm nhiễu nhân (multiplicative noise). Một ví dụ về nhiễu nhân là hiệu ứng nhiễu đốm, thường nhận thấy ở những ảnh được tạo ra từ tia laser có tính kết hợp (coherent) cao như là ảnh radar hông ngoại. Ảnh bị xuống cấp do nhiễu nhân  $g(n_1, n_2)$ , được biểu diễn là

$$g(n_1, n_2) = f(n_1, n_2) v(n_1, n_2) \quad (3.76)$$

trong đó  $v(n_1, n_2)$  là nhiễu ngẫu nhiên, không phải là hàm của  $f(n_1, n_2)$ . Vì  $g(n_1, n_2)$  và  $f(n_1, n_2)$  biểu thị những cường độ ảnh và do đó là không âm,  $v(n_1, n_2)$  cũng là số không âm. Bằng cách áp dụng thuật toán logarit cho (3.76), ta có.

$$T[g(n_1, n_2)] = \log g(n_1, n_2) = \log f(n_1, n_2) + \log v(n_1, n_2) \quad (3.77)$$

Nếu ta ký hiệu  $\log g(n_1, n_2)$  bằng  $g'(n_1, n_2)$  và cũng ký hiệu  $\log f(n_1, n_2)$  và  $\log v(n_1, n_2)$  tương tự, thì (3.77) trở thành:

$$g'(n_1, n_2) = f'(n_1, n_2) + v'(n_1, n_2) \quad (3.78)$$

Nhiễu nhân  $v(n_1, n_2)$  bây giờ đổi thành nhiễu cộng  $v'(n_1, n_2)$  và các algorit phục hồi ảnh được khai triển để làm giảm nhiễu cộng không phụ thuộc tín hiệu có thể được áp

dụng để làm giảm  $v'(n_1, n_2)$ . Ảnh kết quả được mũ hoá để bù lại phép toán logarithm, toàn bộ hệ biểu diễn trên hình 3.26.

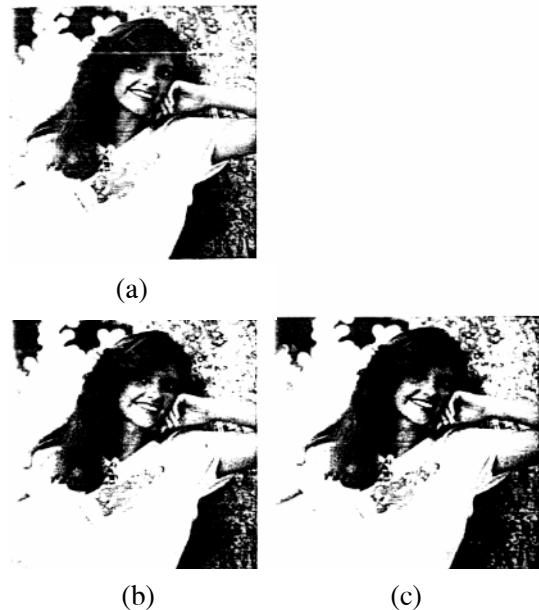
Hình 3.27 minh họa hiệu năng của algorit phục hồi ảnh h này trong việc làm giảm nhiễu nhôm. Nhiêu  $v(n_1, n_2)$  là nhiễu trắng được tạo ra bằng cách sử dụng hàm mật độ xác suất.

$$p_{v(n_1, n_2)}(v_0) = ke^{-(v_0 - \sigma_1)^2 / 2\sigma_2^2 u(v_0)} \quad (3.79)$$

Trong đó  $u(v_0)$  là hàm bậc thang đơn vị,  $\sigma_1$  và  $\sigma_2$  là những hằng số và  $k$  là hệ số tỉ lệ để đảm bảo tích phân của hàm mật độ xác suất bằng 1. Hình 3.27(a) là ảnh gốc 512 x 512 pixels. Hình 3.27(b) là ảnh bị xuống cấp bởi nhiễu nhôm  $v(n_1, n_2)$  nhận được từ (3.79) với  $\sigma_1 = 1$  và  $\sigma_2 = 1$ . Hình 3.27(c) là ảnh đã xử lý bằng hệ trong hình 3.26. Mức cải thiện SNR ở ảnh được xử lý là 5,4 dB. Hệ phục hồi để làm giảm nhiễu cộng là bộ lọc Wiener thích nghi được thảo luận trong tiết 3.2.4.

**Hình 3.27:** Minh họa hiệu năng của hệ làm giảm nhiễu nhôm trên hình 3.26.

- (a) Ảnh gốc 512x512 pixel ;
- (b) Ảnh bị xuống cấp bởi nhiễu nhôm với  $NMSE = 5,1\%$ ;
- (d) Ảnh được xử lý với  $NMSE = 1,5\%$  và mức cải thiện SNR = 5,4 dB.



Ảnh nhoè được thảo luận trong tiết 3.3 cũng là nhiễu phụ thuộc tín hiệu. Bộ lọc ngược để khử nhoè có thể coi như là bộ biến đổi để làm cho ảnh nhoè thành ra ảnh bị xuống cấp vì nhiễu cộng không phụ thuộc tín hiệu, lọc bỏ thành phần nhiễu cộng và sau đó biến đổi ngược trở lại miền tín hiệu. Cụ thể, từ mô hình xuống cấp do nhoè.

$$G(\omega_1, \omega_2) = F(\omega_1 \omega_2).B(\omega_1, \omega_2) \quad (3.80)$$

Áp dụng thuật toán logarith phức cho (3.80), ta có.

$$\log G(\omega_1, \omega_2) = \log F(\omega_1, \omega_2) + \log B(\omega_1, \omega_2) \quad (3.81)$$

Đem  $\log G(\omega_1, \omega_2)$  trừ đi thành phần cộng  $\log B(\omega_1, \omega_2)$  và mũ hoá kết quả cũng tương đương với lọc ngược. Một ví dụ khác biến đổi nhiễu phụ thuộc tín hiệu thành nhiễu cộng không phụ thuộc tín hiệu để lọc là trường hợp giải tương quan (decorrelation) nhiễu lượng tử trong mã hoá ảnh, sẽ thảo luận trong chương 4.

## 5.2. GIẢM NHIỄU PHỤ THUỘC TÍN HIỆU TRONG MIỀN TÍN HIỆU

Ưu điểm chính của cách tiếp cận được thảo luận trong tiết trước là đơn giản. Tuy nhiên cách tiếp cận này dựa vào giả thiết có thể tìm được một miền, ở đó nhiễu phụ thuộc tín hiệu trở thành nhiễu cộng không phụ thuộc tín hiệu. Đối với một số loại nhiễu phụ thuộc tín hiệu có thể không tồn tại một miền như vậy. Ngay cả khi tìm được một miền như vậy, bài toán khôi phục ảnh phải giải quyết trong miền mới và có thể gây ra một ít xuống cấp trong việc thực hiện. Để xem xét điều này, ta giả thiết đã khai triển được algorit làm giảm nhiễu không phụ thuộc tín hiệu  $v(n_1, n_2)$  trong:

$$g(n_1, n_2) = f(n_1, n_2) + v(n_1, n_2) \quad (3.82)$$

bằng cách cố gắng tối thiểu hoá

$$Error = E \left[ \left( f(n_1, n_2) - \hat{f}(n_1, n_2) \right)^2 \right] \quad (3.83)$$

Nếu cũng sử dụng algorit này làm giảm nhiễu không phụ thuộc tín hiệu  $v(n_1, n_2)$  trong:

$$T[g(n_1, n_2)] = T_1[f(n_1, n_2)] + v(n_1, n_2) \quad (3.84)$$

Nó sẽ có xu thế làm giảm

$$Error = E \left[ \left( T_1[f(n_1, n_2)] - \hat{T}_1[\hat{f}(n_1, n_2)] \right)^2 \right] \quad (3.85)$$

Các biểu thức sai số trong (3.83) và (3.85) không giống nhau.

Có một cách tiếp cận để làm giảm nhiễu phụ thuộc tín hiệu bằng cách sử dụng trực tiếp tiêu chuẩn sai số trong (3.83). Lời giải tối ưu để tối thiểu hóa sai số trong (3.83) là:

$$\hat{f}(n_1, n_2) = E[f(n_1, n_2) | g(n_1, n_2)] \quad (3.86)$$

Đó là kỳ vọng (expectation) có điều kiện của  $f(n_1, n_2)$  khi biết  $g(n_1, n_2)$ . Tuy phương trình (3.86) trông có vẻ đơn giản, nhưng việc giải nó thường rất khó. Để giải đơn giản hơn, ta giả thiết bộ ước lượng (estimator) là tuyến tính:

$$\hat{f}(n_1, n_2) = \sum_{k_1} \sum_{k_2} g(k_1, k_2) h(n_1, n_2; k_1, k_2) + c(n_1, n_2) \quad (3.87)$$

trong đó  $h(n_1, n_2; k_1, k_2)$  và  $c(n_1, n_2)$  được chọn để  $E[\hat{f}(n_1, n_2) - f(n_1, n_2)]^2$  cực tiểu.

Lời giải của bài toán tối ưu hoá tuyến tính có thể viết gọn lại bằng cách sử dụng ký hiệu ma trận. Như đã thảo luận trong tiết 3.7, chi phí tính toán để tìm lời giải chung vẫn còn lớn và yêu cầu những hàm thống kê bậc nhất và bậc hai như hàm hiệp biến chéo (cross covariance) của tín hiệu  $f(n_1, n_2)$  và nhiễu  $d(n_1, n_2)$  phụ thuộc tín hiệu, trong thực tế vẫn còn rất khó giải. Tuy nhiên, nếu ta đưa ra một vài giả định, thì cách giải sẽ được đơn giản đi một cách đáng kể.

Ví dụ như, ta giả sử rằng tín hiệu  $f(n_1, n_2)$  và nhiễu phụ thuộc tín hiệu  $d(n_1, n_2)$  là những mẫu của quá trình ngẫu nhiên dừng. Khi đó lời giải cho ước lượng sai số quân phương tối thiểu tuyến tính của  $f(n_1, n_2)$  là:

$$\hat{f}(n_1, n_2) = m_f + (g(n_1, n_2) - m_g) * h(n_1, n_2), \quad (3.88)$$

$$H(\omega_1, \omega_2) = F[h(n_1, n_2)] = \frac{P_{fg}(\omega_1, \omega_2)}{P_g(\omega_1, \omega_2)} \quad (3.89)$$

trong đó  $m_f = E[f(n_1, n_2)]$ ,  $m_g = E[g(n_1, n_2)]$ ,  $P_{fg}(\omega_1, \omega_2)$  là phổ công suất chéo của  $f(n_1, n_2)$  và tín hiệu bị xuống cấp  $g(n_1, n_2)$ , và  $P_g(\omega_1, \omega_2)$  là phổ công suất của  $g(n_1, n_2)$ . Đây là lời giải của bộ lọc Wiener được thảo luận trong phụ lục. Khi  $d(n_1, n_2)$  không phụ thuộc tín hiệu, phương trình (3.89) trở về dạng đơn giản của bộ lọc Wiener đã thảo luận trong tiết 3.2. Vì tín hiệu  $f(n_1, n_2)$  và nhiễu phụ thuộc tín hiệu  $d(n_1, n_2)$  không thể giả thiết là dừng trong bài toán phục hồi ảnh, bộ lọc (3.88) và (3.89) có thể được thực hiện cục bộ theo kiểu thích nghi chừng nào  $P_{fg}(\omega_1, \omega_2)$  và  $P_g(\omega_1, \omega_2)$  có thể được ước lượng cục bộ.

Trong ví dụ khác, mà lời giải cho ước lượng sai số quân phương tối thiểu tuyến tính của  $f(n_1, n_2)$  đơn giản đi nhiều thì tín hiệu  $f(n_1, n_2)$  có thể mô hình hoá là:

$$f(n_1, n_2) = m_f(n_1, n_2) + \sigma_f(n_1, n_2) w(n_1, n_2) \quad (3.90)$$

trong đó  $m_f(n_1, n_2)$  là  $E[f(n_1, n_2)]$ ,  $\sigma_f^2(n_1, n_2)$  là phương sai của  $f(n_1, n_2)$  và  $w(n_1, n_2)$  là nhiễu trắng trung vị bằng không và phương sai bằng 1. Đó cũng là mô hình đã dùng trong algorit phục hồi ảnh đã khai thiển trong tiết 3.2.4. Đối với một số lớp nhiễu phụ thuộc tín hiệu bao gồm cả nhiễu nhân và nhiễu Poisson, mô hình tín hiệu (3.90) dẫn đến algorit rất đơn giản. Để minh họa điều này, ta xem lại vấn đề làm giảm tập âm nhân.

Xét một ảnh bị xuống cấp  $g(n_1, n_2)$  là:

$$g(n_1, n_2) = f(n_1, n_2)v(n_1, n_2) \quad (3.91)$$

trong đó  $v(n_1, n_2)$  là nhiễu trắng dừng với trung vị bằng  $m_v$  và phương sai là  $\sigma_v^2$ . Từ (3.90) và (3.91) ta có

$$g(n_1, n_2) = (m_f(n_1, n_2) + \sigma_f(n_1, n_2)w(n_1, n_2))v(n_1, n_2) \quad (3.92)$$

Vì cả  $w(n_1, n_2)$  và  $v(n_1, n_2)$  đều là trắng, mọi thông tin hữu quan về  $f(n_1, n_2)$  ở một điểm  $(n_1, n_2)$  đều chứa đựng trong  $g(n_1, n_2)$ . Như thế, ta có thể coi vấn đề ước lượng  $f(n_1, n_2)$  ở từng pixel như ước lượng một biến ngẫu nhiên  $f$  từ một quan sát  $g$  tính theo:

$$g = fv = (m_f + \sigma_f w)v. \quad (3.93)$$

Giả định một bộ ước lượng (estimator) tuyến tính, ta nhận được:

$$\hat{f} = ag + b \quad (3.94)$$

trong đó  $a$  và  $b$  có thể được xác định đơn giản. Theo lý thuyết ước lượng,  $E[\hat{f}] = E[f]$

được tối thiển hóa bằng cách đưa ra nguyên lý trực giao, đặt ra hai điều kiện bắt buộc sau đây khi áp dụng vào bài toán ước lượng trên:

$$E[f - \hat{f}] = 0 \quad (3.95a)$$

$$E[(f - \hat{f})g] = 0 \quad (3.95b)$$

Từ (3.94) và (3.95)

$$E[f - ag - b] = 0 \quad (3.96a)$$

$$E[(f - ag - b)g] = 0 \quad (3.96b)$$

Giải hai phương trình tuyến tính trong (3.96) cho  $a$  và  $b$ , ta nhận được:

$$a = \frac{E[fg] - E[f]E[g]}{\sigma_g^2} \quad (3.97a)$$

và

$$b = E[f] - E[g]a \quad (3.97b)$$

trong đó  $\sigma_g^2$  là phương sai của g. Từ (3.93) và (3.94) và (3.97),

$$\hat{f} = m_f + (g - m_g)a \quad (3.98a)$$

trong đó

$$a = \frac{m_v \sigma_f^2}{\sigma_g^2} \quad (3.98b)$$

Từ (3.93) và (3.98) và sau một vài phép tính đại số.

$$\hat{f} = \frac{m_g}{m_v} + (g - m_g) \frac{\sigma_g^2 m_v^2 - m_g^2 \sigma_v^2}{m_v \sigma_g^2 (\sigma_v^2 + m_v^2)} \quad (3.99)$$

Vì (3.99) có thể sử dụng để ước lượng  $f(n_1, n_2)$  ở mỗi pixel

$$\hat{f}(n_1, n_2) = \frac{m_g(n_1, n_2)}{m_v} + (g(n_1, n_2) - m_g(n_1, n_2)) \frac{\sigma_g^2(n_1, n_2)m_v^2 - m_g^2(n_1, n_2)\sigma_v^2}{m_v \sigma_g^2(n_1, n_2)(\sigma_v^2 + m_v^2)} \quad (3.100)$$

Các đại lượng thống kê về nhiễu  $\sigma_v^2$  và  $m_v$  giả sử đã biết. Chừng nào mô hình tín hiệu còn phù hợp và còn có thể ước lượng  $m_g(n_1, n_2)$  và  $\sigma_g^2(n_1, n_2)$  từ  $g(n_1, n_2)$  ở vùng lân cận cục bộ mỗi pixel, thì còn có thể được ước lượng  $f(n_1, n_2)$  từ (9100). Cách tiếp cận này có thể sử dụng để khai triển các algorit làm giảm các loại nhiễu phụ thuộc tín hiệu, kể cả nhiễu Poisson.



**Hình 3.28:** Ảnh trong hình 3.27(b) được xử lý

bởi một hệ làm giảm nhiễu nhân dựa vào  
phương trình (3.100). Ảnh được xử lý có  
 $NMSE = 2,8\%$  và mức cải thiện  $SNR = 2,5$  dB.

Hình 3.28 là kết quả áp dụng phương trình (3.100) để làm giảm nhiễu nhôm. Ảnh bị xuống cấp được sử dụng trong hình 3.27(b). Ảnh được xử lý chỉ ra trên hình có Mức cải thiện SNR = 2,5dB.

## 6. PHÉP LỌC THỜI GIAN CHO PHỤC HỒI ẢNH

Trong các tiết trước, ta đã thảo luận về các algorit lọc không gian cho phục hồi ảnh. Trong các ứng dụng như ảnh động, tồn tại một dãy ảnh tương quan với nhau trong thứ nguyên thời gian và có thể được khai thác sự tương quan thời gian này qua sự lọc thời gian. Một ưu điểm lớn của sự lọc thời gian so với sự lọc không gian là nó có khả năng làm giảm xuống cấp mà không gây méo(distortion) tín hiệu. Trong tiết này ta thảo luận về các algorit lọc thời gian.

### 6.1. LẤY TRUNG BÌNH KHUNG

Phương pháp đơn giản nhất của bộ lọc thời gian là lấy trung bình khung, rất hiệu quả trong việc xử lý một dãy ảnh, ở đó ảnh không thay đổi từ khung này tới khung khác nhưng sự xuống cấp lại thay đổi. Có nhiều kiểu trung bình khung khác nhau. Cách đơn giản nhất và thông dụng nhất là ước lượng ảnh  $f(n_1, n_2)$  từ một dãy  $N$  khung ảnh bị xuống cấp  $g_i(n_1, n_2)$ , khi  $1 \leq i \leq N$ :

$$\hat{f}(n_1, n_2) = \frac{1}{N} \sum_{i=1}^N g_i(n_1, n_2) \quad (3.101)$$

Một ví dụ khác của trung bình khung là:

$$\hat{f}(n_1, n_2) = \left( \prod_{i=1}^N g_i(n_1, n_2) \right)^{1/N} \quad (3.102)$$

Loại hình trung bình thời gian nào là tốt nhất trong một ứng dụng phụ thuộc vào nhiều yếu tố, bao gồm tiêu chí sai số được sử dụng và những giả định ban đầu về sự xuống cấp của ảnh.

Giả sử có một dãy ảnh bị xuống cấp  $g_i(n_1, n_2)$  được đại biểu bởi:

$$g_i(n_1, n_2) = f(n_1, n_2) + v_i(n_1, n_2), \quad 1 \leq i \leq N \quad (3.103)$$

trong đó  $v_i(n_1, n_2)$  là nhiễu Gauss trắng dừng, trung vị bằng không, phương sai là  $\sigma_v^2$ , và  $v_i(n_1, n_2)$  độc lập với  $v_j(n_1, n_2)$  khi  $i \neq j$ . Nếu ta giả sử rằng  $f(n_1, n_2)$  là không ngẫu

nhiên, thì đánh giá gần đúng nhất (ML) của  $f(n_1, n_2)$  có thể cực đại hoá  $P_{g_i(n_1, n_2) \mid f(n_1, n_2)}(g'_i(n_1, n_2) \mid f'(n_1, n_2))$  là:

$$\hat{f}(n_1, n_2) = \frac{1}{N} \sum_{i=1}^N g_i(n_1, n_2) \quad (3.104)$$

Từ (3.103) và (3.104)

$$\hat{f}(n_1, n_2) = f(n_1, n_2) + \frac{1}{N} \sum_{i=1}^N v_i(n_1, n_2) \quad (3.105)$$

Từ (3.105), sự xuống cấp trong ảnh trung bình hoá khung vẫn là nhiễu trắng Gauss dùng có trung vị bằng 0, phương sai bằng  $\sigma^2/N$ , biểu diễn sự giảm phương sai nhiễu N lần so với  $v_i(n_1, n_2)$ . Vì đặc tính nhiễu trong ví dụ này không đổi trong khi phương sai bị giảm, ảnh được xử lý sẽ không nhòe chút nào mặc dù nhiễu giảm. Phép trung bình hoá khung trong (3.104) thường sử dụng để giảm nhiễu có thể xuất hiện khi số hoá ảnh tĩnh bằng một thiết bị chụp hình như camera vidic on.

Lấy một ví dụ khác, ta giả sử  $g_i(n_1, n_2)$  có thể được biểu thị là:

$$g_i(n_1, n_2) = f(n_1, n_2) v_i(n_1, n_2) \quad (3.106)$$

trong đó  $v_i(n_1, n_2)$  cho mỗi bộ giá trị  $i$  và  $(n_1, n_2)$  là một mẫu độc lập dẫn xuất từ hàm mật độ xác suất sau:

$$p_v(v_0) = \lambda(\exp(-\lambda v_0) u(v_0)) \quad (3.107)$$

trong đó  $u(\cdot)$  là hàm bậc thang đơn vị. Trong một số trường hợp có thể mô hình hoá nhiễu đốm được bằng nhiễu nhân trong (3.106) và (3.107) với  $\lambda = 1$ . Nếu ta ký hiệu  $g$  là  $g_i(n_1, n_2)$  ở một bộ giá trị  $i$  và  $(n_1, n_2)$  nào đó, tương ứng cũng ký hiệu  $f$  và  $v$  theo thứ tự là  $f(n_1, n_2)$  và  $v_i(n_1, n_2)$  lấy từ (3.106) và (3.107).

$$P_{g \mid f}(g_\theta \mid f_\theta) = \frac{\lambda}{f_\theta} \exp(-\lambda g_\theta \mid f_\theta) u(g_\theta) \quad (3.108)$$

Từ (3.106) và (3.108), nếu giả định rằng  $f(n_1, n_2)$  là không ngẫu nhiên, đánh giá gần đúng nhất (ML, - Maximum Likelihood) của  $f(n_1, n_2)$  là:

$$\hat{f}(n_1, n_2) = \frac{\lambda}{N} \sum_{i=1}^N g_i(n_1, n_2) \quad (3.109)$$

Từ (3.106) và (3.109),

$$\hat{f}(n_1, n_2) = f(n_1, n_2) \frac{\lambda}{N} \sum_{i=1}^N v_i(n_1, n_2) \quad (3.110)$$

Từ (3.110) thấy rằng kiểu xuống cấp trong ảnh đã xử lý vẫn là nhiễu nhôm trong ví dụ này.

Phương pháp đánh giá gần đúng nhất (ML) sử dụng trong hai ví dụ trên thường đơn giản hơn phương pháp ước lượng sai số quan phương tối thiểu(MMSE) hoặc phương pháp ước lượng cực đại hậu nghiệm (MAP). Trong phương pháp ước lượng ML (maximum likelihood) ta giả thiết là không có sẵn thông tin trước về các thông số (hoặc tín hiệu) cần ước lượng, vì vậy chỉ nhận được lời giải có ích, khi số lượng các thông số cần ước lượng nhỏ hơn số lần quan sát. Chẳng hạn, trong (3.104) và (3.110) nếu  $N = 1$ , làm cho số thông số cần ước lượng bằng số lần quan sát, thì  $\hat{f}(n_1, n_2)$  chính là tín hiệu bị xuống cấp hoặc chỉ khác tín hiệu bị xuống cấp một hệ số tỉ lệ. Đó là một lý do tại sao trong việc khai triển algorit bộ lọc không gian ở các tiết trước, ta đã dựa vào tiêu chí MMSE chứ không phải tiêu chí ML. Tiêu chí MMSE kết hợp quan sát với một vài thông tin biết trước về các thông số cần ước lượng. Chừng nào thông tin có sẵn còn chính xác và có ý nghĩa, thì còn có thể khai triển một quy trình ước lượng có ích, ngay cả khi mà số thông số cần ước lượng có thể so sánh được với số lần quan sát .

**Hình 3.29:** Minh họa hiệu năng của trung bình hoá khung.

- (a) Ảnh gốc 512 x 512 pixels;
- (b) Ảnh bị xuống cấp bởi nhiễu trắng, với SNR = 2dB và NMSE = 65,2%;
- (c) Ảnh được xử lý bằng cách lấy trung bình hoá khung (8 khung), có NMSE = 8,5% và Mức cải thiện SNR = 8,8 dB.



(a)



(b)

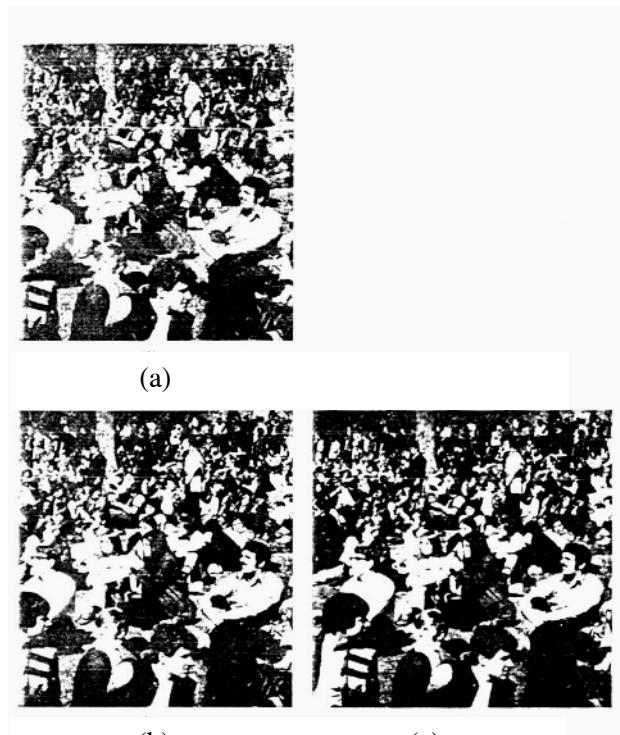


(c)

Hình 3.29 và 3.30 minh họa hiệu năng của trung bình hoá khung. Hình 3.29(a) là ảnh gốc  $512 \times 512$  pixels. Hình 3.29(b) là ảnh bị xuống cấp bởi nhiễu trắng Gauss dừng, có trung vị bằng 0, mức SNR là 2 dB. Hình 3.29(c) là kết quả của việc lấy trung bình 8 khung bị xuống cấp sử dụng (3.104). Hình 3.30(a) là ảnh gốc trong hình 3.29(a) bị xuống cấp bởi nhiễu nhân của (3.106) và (3.107) với  $\lambda = 1$ . Hình 3.30(c) là kết quả lấy trung bình 8 khung ảnh bị xuống cấp, sử dụng (3.110).

**Hình 3.30:** Minh họa hiệu năng của trung bình hoá khung.

- (a) Ảnh trong hình 3.29(a) bị xuống cấp bởi nhiễu nhân, với NMSE = 28,8%;
- (b) Ảnh được xử lý bằng cách lấy trung bình khung (8 khung), với NMSE = 4,7% và Mức cải thiện SNR = 7,9dB;
- (c) Ảnh được xử lý bằng cách áp dụng một bộ lọc không gian tới ảnh trong (b), có NMSE = 4,3% và Mức cải thiện SNR = 8,3dB.



Ngoài ra lọc thời gian cũng có thể thực hiện lọc không gian. Chẳng hạn, ảnh đã xử lý trong hình 3.30(b) có thể được mô hình hóa như một ảnh gốc bị xuống cấp bởi nhiễu nhân mà đặc tính có thể nhận được từ đặc tính nhiễu trong khung bị xuống cấp đơn. Nếu ta áp dụng phương pháp trong tiết 3.5.2 để làm giảm nhiễu nhân cho ảnh ở hình 3.30(b), ta sẽ nhận được ảnh biểu diễn trên hình 3.30(c). Trong ví dụ này xử lý riêng rẽ, các phép lọc thời gian và không gian được thực hiện riêng. Cũng có thể thiết kế và thực hiện chung một bộ lọc không gian -thời gian 3-D, nhưng với cách tiếp cận

này chi phí tính toán thường cao hơn xử lý riêng biệt nhiều, mà không có sự cải tiến hiệu năng đáng kể.

Mặc dù trung bình khung trong một số hình thức thường rất đơn giản và có hiệu quả, nhưng đăng ký-ghi lại (registration) tín hiệu chính xác là yếu tố quan trọng cho sự thành công. Trong nhiều ứng dụng như ảnh động và truyền hình, ảnh có thể thay đổi từ khung này sang khung kề theo sau do đối tượng chuyển động, camera chuyển động, v.v.. Để áp dụng một hình thức trung bình khung nào đó cho những khung ảnh này, ta cần ước lượng được sự chuyển động của ảnh từ một khung tới khung tiếp theo. Việc bù chuyển động phục hồi ảnh sẽ được thảo luận trong tiết tiếp theo.

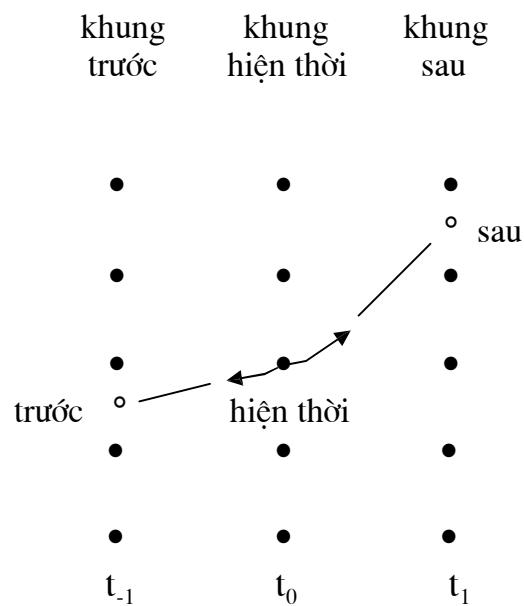
## 6.2. PHỤC HỒI ẢNH BẰNG BÙ CHUYỂN ĐỘNG

Trong phục hồi ảnh bằng bù chuyển động thường ước lượng trước các thông số chuyển động, sau đó các khung ảnh được lọc dọc theo quỹ đạo chuyển động. Các algorit ước lượng chuyển động đã thảo luận trong tiết 2.4.2 có thể sử dụng để ước lượng quỹ đạo chuyển động. Vì các algorit trong tiết 2.4.2 được khai triển với giả thiết đã có các khung ảnh không nhiễu (noise-free), cho nên cần phải xét đến sự xuống cấp trước khi ước lượng chuyển động. Một vài algorit ước lượng chuyển động, như algorit ước lượng chuyển động có ràng buộc không gian-thời gian dùng nội suy đa thức tín hiệu đã thảo luận trong tiết 2.4.2 có khuynh hướng ít nhạy cảm với nhiễu. Với các algorit khác, thực hiện vài phép giảm xuống cấp đơn giản trước khi ước lượng chuyển động có thể nâng cao hiệu năng.

Loại hình phép lọc thực hiện dọc theo quỹ đạo chuyển động phụ thuộc vào kiểu xuống cấp. Kỹ thuật phục hồi ảnh bằng bù chuyển động thường nói đến trên các tạp chí chủ yếu được dùng làm giảm nhiễu cộng ngẫu nhiên. Để làm giảm nhiễu cộng ngẫu nhiên, cường độ ảnh trong các khung khác nhau dọc theo quỹ đạo chuyển động được lọc thông thấp. Cả bộ lọc FIR và bộ lọc IIR đều được xét. Vì phép lọc thời gian bao hàm cả lưu trữ các khung, nên thông thường chỉ sử dụng các bộ lọc FIR và IIR bậc thấp.

Hình 3.31 minh họa bộ lọc FIR ba điểm áp dụng dọc theo quỹ đạo chuyển động. Khung giữa  $g(n_1, n_2, t_0)$  trong hình là khung hiện thời (current) đang được xử lý. Ảnh được xử lý  $\hat{f}(n_1, n_2, t_0)$  nhận được từ  $g(n_1, n_2, t_0)$ , khung trước  $g(n_1, n_2, t_1)$  và khung sau  $g(n_1, n_2, t_1)$ . Tại mỗi pixel trong khung hiện thời tính hai tốc độ (hoặc hai đoạn)

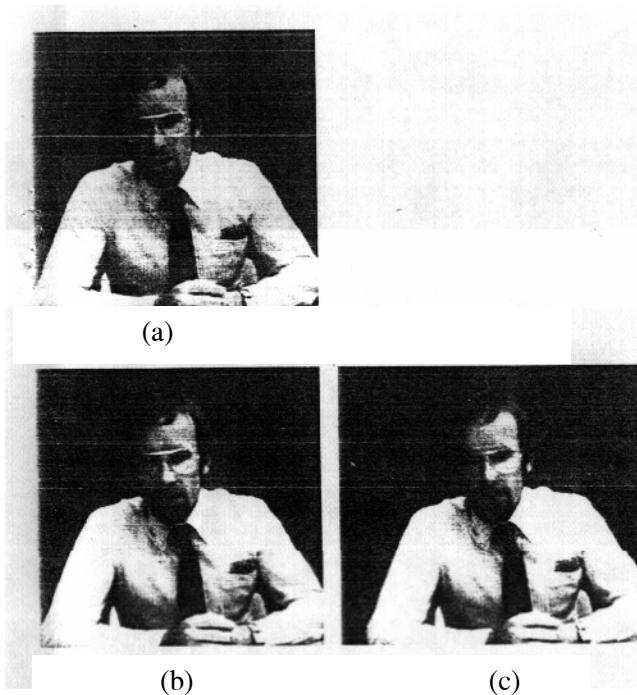
dịch chuyển, một cái từ khung  $g(n_1, n_2, t_{-1})$  và  $g(n_1, n_2, t_0)$ , cái kia từ khung  $g(n_1, n_2, t_0)$  và  $g(n_1, n_2, t_1)$ . Hai đoạn dịch chuyển dùng để xác định các vị trí không gian ứng với các thời điểm  $t_{-1}$  và  $t_1$ , như trên hình 3.31. Lấy trung bình ba cường độ ảnh ứng với ba pixels để nhận được  $\hat{f}(n_1, n_2, t_0)$  ở pixel. Vì các vị trí không gian ứng với các thời điểm  $t_{-1}$  và  $t_1$  thường không nằm đúng trên lưới lấy mẫu (sampling grid), cho nên cần nội suy không gian để xác định  $g(x, y, t_{-1})$  và  $g(x, y, t_1)$  ở các vị trí không gian mong muốn.



**Hình 3.31:** Phục hồi ảnh bằng bù chuyển động.

Hình 3.32 minh họa hiệu năng của bộ lọc thời gian FIR ba -điểm được áp dụng dọc theo quỹ đạo chuyển động. Hình 3.32(a) là khung hiện thời bị xuống cấp bởi nhiều Gauss trắng có trung vị bằng 0. Hình 3.32(b) là kết quả của trung bình khung bù chuyển động. Một dãy ba khung bị xuống cấp(trước, hiện thời và sau) được dùng trong phép xử lý. Algorit ước lượng chuyển động được sử dụng là phương pháp ràng buộc không gian-thời gian với phép nội suy đa thức tín hiệu đã thảo luận trong tiết 2.4.4. Sau khi đã ước lượng được các thông số chuyển động, dùng bộ lọc nội suy cắt ngắn lý tưởng để nội suy không gian theo yêu cầu của phép lọc thời gian. Hình 3.32(c) là kết quả của

việc lấy trung bình khung không bù chuyển động. Sự nhòe ảnh ở đây thể hiện lượng chuyển động tồn tại trong dãy ba khung ảnh được sử dụng.



**Hình 3.32:** Minh họa hiệu năng của phục hồi ảnh bằng bù chuyển động.

- (a) Khung hiện thời bị xuống cấp;
- (b) Ảnh được xử lý bằng cách lấy trung bình khung bù chuyển động;
- (c) Ảnh được xử lý bằng cách lấy trung bình khung không bù chuyển động. Sự nhòe ảnh ở đây thể hiện lượng chuyển động tồn tại trong dãy ba khung ảnh được sử dụng.

## 7. BÌNH LUẬN

Trong chương này, ta sử dụng ảnh đơn sắc để minh họa hiệu năng của các hệ phục hồi ảnh khác nhau. Hầu hết những điều đã thảo luận đều áp dụng được cho phục hồi ảnh màu. Như đã nói trong chương 1, một ảnh màu được biểu diễn bằng ba ảnh đơn sắc. Để phục hồi ảnh màu, ta có thể lần lượt xử lý ba ảnh đơn sắc riêng rẽ và đem kết quả tổ hợp lại. Hoặc là ta có thể phục hồi một vector gồm cả ba màu ảnh đơn sắc,

khai thác những hiểu biết về tín hiệu và về sự xuống cấp trong mỗi ảnh đơn sắc và tương quan giữa chúng. Nguyên lý chung và cách tiếp cận cơ bản đã thảo luận trong chương này để khai triển các hệ phục hồi ảnh khác nhau cho ảnh đơn sắc cũng áp dụng được cho một vector gồm cả 3 ảnh đơn sắc.

Trong chương này sử dụng cách kí hiệu thường dùng trong lý thuyết xử lý tín hiệu số để khai triển các algorit phục hồi ảnh. Cũng có thể sử dụng kí hiệu ma trận để khai triển các algorit phục hồi ảnh. Ưu điểm lớn của kí hiệu ma trận là có thể biểu diễn algorit phục hồi ảnh dưới dạng rất gọn (compact) và có thể tận dụng những thành tựu của đại số tuyến tính, và giải tích số.

Xét một ảnh  $f(n_1, n_2)$  có  $N \times N$  pixels.  $N^2$  phần tử trong  $f(n_1, n_2)$  có thể được biểu diễn bởi  $N^2 \times 1$  vector cột  $f$ . Có nhiều cách để sắp xếp  $N^2$  phần tử lại thành  $f$ . Một phương pháp là sắp xếp các phần tử từ dưới lên trên, từ trái qua phải:

$$f = [f(0, 0), f(0, 1), \dots, f(0, N-1), f(1, 0), f(1, 1), \dots, f(1, N-1), \dots, f(N-1, 0), f(N-1, 1), \dots, f(N-1, N-1)]^T \quad (3.111)$$

Xét một mô hình ảnh xuống cấp là:

$$g(n_1, n_2) = f(n_1, n_2) * b(n_1, n_2) + v(n_1, n_2) \quad (3.112)$$

Phương trình (3.112) có thể biểu thị như là:

$$g = Bf + v \quad (3.113)$$

với cách chọn  $g$ ,  $B$ ,  $f$  và  $v$  thích hợp. Chẳng hạn, giả sử rằng  $f(n_1, n_2)$  và  $b(n_1, n_2)$  là những dãy  $2 \times 2$  điểm, điểm không nằm ngoài  $0 \leq n_1 \leq 1, 0 \leq n_2 \leq 1$  và  $g(n_1, n_2)$  và  $v(n_1, n_2)$  là những dãy  $3 \times 3$  điểm, điểm không nằm ngoài  $0 \leq n_1 \leq 2, 0 \leq n_2 \leq 2$ , thì một tập  $f$ ,  $B$ ,  $v$  và  $g$  là:

$$f = [f(0, 0), f(0, 1), f(1, 0), f(1, 1)]^T \quad (3.114a)$$

$$v = [v(0, 0), v(0, 1), v(0, 2), v(1, 0), v(1, 1), v(1, 2), v(2, 0), v(2, 1), v(2, 2)]^T \quad (3.114b)$$

$$g = [g(0, 0), g(0, 1), g(0, 2), g(1, 0), g(1, 1), g(1, 2), g(2, 0), g(2, 1), g(2, 2)]^T \quad (3.114c)$$

$$B = \begin{bmatrix} b(0,0) & 0 & 0 & 0 \\ b(0,1) & b(0,0) & 0 & 0 \\ 0 & b(0,1) & 0 & 0 \\ b(1,0) & 0 & b(0,0) & 0 \\ b(1,1) & b(1,0) & b(0,1) & b(0,0) \\ 0 & b(1,1) & 0 & b(0,1) \\ 0 & 0 & b(1,0) & 0 \\ 0 & 0 & b(1,1) & b(1,0) \\ 0 & 0 & 0 & b(1,1) \end{bmatrix} \quad (3.114d)$$

Biểu thức (3.113) rất tổng quát. Chẳng hạn, ta có thể biểu diễn ảnh nhoè biến đổi trong không gian bằng cách chọn các phần tử thích hợp trong ma trận B,

Nếu ta ước lượng  $f$  bằng một bộ ước lượng tuyến tính có thể cực tiểu hoá:

$$\text{Error} = E[(f - \hat{f})^T (f - \hat{f})] \quad (3.115)$$

Trong đó  $\hat{f}$  là ước lượng của  $f$ , lời giải Sage và Melsa đưa ra là:

$$\hat{f} = E[f] + \Phi_{fg} \Phi_g^{-1} [g - E[g]] \quad (3.116a)$$

$$\text{trong đó } \Phi_{fg} = E[(f - E[f])(g - E[g])^T] \quad (3.116b)$$

$$\text{và } \Phi_g = E[(g - E[g])(g - E[g])^T] \quad (3.116c)$$

Lời giải ở (3.116) được viết dưới một dạng rất gọn.

Ưu điểm của kí hiệu ma trận cũng phải trả giá. Trong bài toán phục hồi ảnh thường gấp,  $f$  có thể gồm một phần tư triệu phần tử và các ma trận như là ma trận B,  $\Phi_{fg}$  hoặc  $\Phi_g$  có thể bao gồm một phần tư triệu  $\times$  một phần tư triệu phần tử. Vì thế, phải có một số giả định để làm cho đơn giản bớt trước khi lời giải của bài toán có thể sử dụng trong thực tế.

Nếu sử dụng cùng một tiêu chuẩn sai số, và các giả thiết ban đầu đưa ra để giải bài toán phục hồi ảnh như nhau thì kết quả nhận được sẽ như nhau, bất kể là dùng cách kí hiệu nào. Tuy vậy các cách kí hiệu khác nhau cũng có thể cung cấp những cách nhìn khác nhau vào bên trong lời giải và góp phần giúp ta hiểu biết sâu sắc hơn bài toán phục hồi ảnh.

## CHƯƠNG 4

### MÃ HOÁ ẢNH

#### v MỎ ĐẦU.

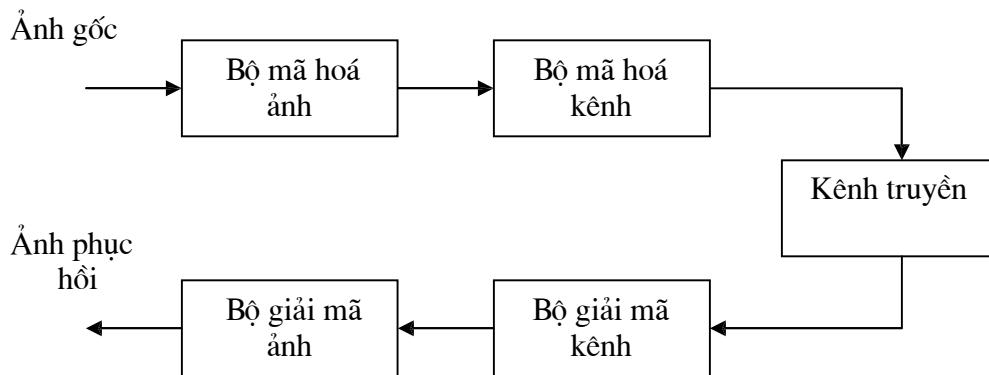
Mục tiêu chính của mã hoá ảnh là làm sao trùng bầy ảnh với số bít càng nhỏ càng tốt trong khi vẫn giữ được mức chất lượng và độ dẽ hiểu ở mức chất lượng vừa đủ với một ứng dụng đã cho. Có hai lĩnh vực ứng dụng:

Một là giảm bê rộng bằng tân cần thiết cho hệ truyền ảnh. Ví dụ truyền hình số, hội nghị video, fax – ứng dụng thứ hai là giảm bớt yêu cầu về lưu trữ. Ví dụ giảm lưu trữ số liệu ảnh trong các chương trình vũ trụ và số liệu video trong máy ghi hình số. Tuỳ theo tính chất của ứng dụng, mức độ chất lượng ảnh và độ dẽ hiểu có thể biến đổi trong một phạm vi rộng. Trong lưu trữ ảnh của chương trình vũ trụ hay lưu trữ ảnh lịch sử (không thể có lại được) phải lưu trữ lại toàn bộ tư liệu số của nguyên bản để sử dụng về sau.

Những kỹ thuật không làm mất tí thông tin nào và cho phép phục hồi chính xác tư liệu số ban đầu, gọi là kỹ thuật có tính bảo tồn thông tin. Trong truyền hình số thì bộ mã hoá không cần phải là loại bảo tồn thông tin như vậy. Ở đây chất lượng cao là quan trọng, nhưng có thể bỏ qua một số thông tin từ tư liệu gốc, trong phạm vi mà tín hiệu giải mã ra và hiện lên màn hình vẫn vừa mắt người xem. Trong ứng dụng về điều khiển con tàu từ xa, độ dẽ hiểu của ảnh là quan trọng nhất, nhưng có thể hi sinh một phần chất lượng. Càng giảm yêu cầu về chất lượng và độ dẽ hiểu, thì tốc độ bit càng hạ.

Mã hoá ảnh liên quan đến cải thiện ảnh và phục chế ảnh. nếu ta có thể cải thiện cảm quan thị giác của ảnh được lập lại hay nếu ta có thể giảm sự xuống cấp do algorit mã hoá hình gây ra (ví dụ như tạp âm lượng tử hoá ) thì ta có thể giảm bớt số lượng bit cần thiết để biểu diễn một ảnh ở mức độ chất lượng và độ dẽ hiểu đã cho, hay có thể giữ nguyên số bit mà cải thiện chất lượng và độ dẽ hiểu .

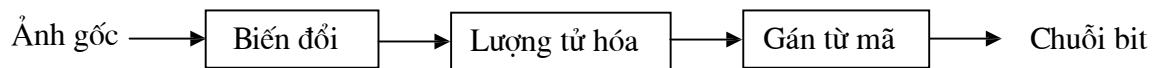
Môi trường điển hình về mã hoá ảnh như trên hình 4.1. Ảnh digital được mã hoá ảnh mã hoá. Bộ mã hoá này gọi là bộ mã hoá nguồn. Đầu ra bộ mã hoá này là một chuỗi bit gọi là ảnh gốc.



**Hình 4.1.** Môi trường điển hình về mã hóa ảnh.

Bộ mã hóa kênh biến chuỗi bit này ra một dạng thích hợp cho việc truyền qua một kênh thông tin, thông qua một dạng điều chế nào đó. Tín hiệu đã điều chế được truyền qua kênh thông tin. Kênh thông tin sẽ đưa vào một ít nhiễu và trong bộ mã hóa kênh phải trữ liệu một biện pháp sửa lỗi để khắc phục tạp âm kênh này. Ở đầu thu, tín hiệu nhận được qua giải điều chế và hoàn nguyên thành chuỗi bit nhờ bộ giải mã kênh. Bộ giải mã ảnh đem chuỗi bít hoàn nguyên thành ảnh cho hiện lên màn hình và in ra. Khác với môi trường truyền tin ở hình 4.1, trong những ứng dụng mã hóa ảnh để giảm lưu trữ, không có kênh thông tin. Ở đây chuỗi bit ở đầu ra bộ mã hóa ảnh được lưu trữ vào môi trường lưu trữ chờ sau lấy ra dùng.

Bộ mã hóa ảnh ở hình 4.1 có ba phần tử cơ bản (Hình 4.2).



**Hình 4.2.** Ba thành phần chính trong mã hóa ảnh.

Phần tử đầu tiên và quan trọng nhất làm biến đổi ảnh vào một không gian (miền) thích hợp nhất cho việc lượng tử hóa và gán từ mã. Về thực chất phần tử này quyết định xem cái gì phải đem mã hóa. Algorit mã hóa ảnh chia làm ba loại chính, tùy theo đặc trưng nào của ảnh được mã hóa. Loại thứ nhất gọi là bộ mã hóa dạng sóng, cường độ ảnh

hay một biến thiên của cường độ ảnh, ví dụ cường độ của hai pixel kề nhau, được mã hoá. Loại thứ hai, gọi là bộ mã hoá hệ số biến đổi (hay hàm biến đổi) , ảnh được biến đổi sang không gian khác, chẳng hạn biến đổi Fourier hoặc biến đổi Cosin, như vậy là sang một miền (domain) khác với miền cường độ, và các hệ số biến đổi được mã hoá. Loại thứ ba gọi là bộ mã hoá mô hình (model) tín hiệu, người ta mô hình hoá ảnh hoặc một mảnh nào đó của ảnh và các thông số của mô hình được mã hoá. Sau đó ảnh được tổng hợp từ các thông số mô hình đã mã hoá. Phần tử thứ hai là để lượng tử hoá. Để biểu diễn một ảnh với một số bít hữu hạn, thì cường độ ảnh, hệ số biến đổi hay thông số mô hình phải được lượng tử hoá. Việc lượng tử hoá bao gồm việc gán mức lượng tử và các biên quyết định. Phần tử thứ ba để gán từ mã tức là chuỗi bít biểu diễn các mức lượng tử.

Mỗi phần tử đều nhằm để khai thác sự dư thừa trong ảnh gốc và những giới hạn của thiết bị hiện hình cũng như của hệ thị giác con người .

Vì vậy ba phần tử liên quan chặt chẽ với nhau. Chẳng hạn nếu phần tử biến đổi trong bộ mã hoá làm cho các số liệu giảm sự tương quan đủ mức thì ưu thế của lượng tử hóa vectơ so với lượng tử hóa vô hướng giảm đi. Nếu các mức lượng tử trong bộ lượng tử hóa được chọn sao cho mỗi mức được sử dụng với xác suất như nhau thì ưu thế của từ mã có độ dài biến đổi so với từ mã có độ dài cố định giảm đi.

## 1. LUỢNG TỬ HOÁ.

### 1.1. Lượng tử hoá vô hướng.

Gọi  $f$  là một lượng vô hướng liên tục, có thể đại biểu cường độ một pixel hoặc một hệ số biến đổi hay một thông số của mô hình ảnh. Để biểu diễn  $f$  bằng một số lượng bit hữu hạn, ta chỉ dùng một số lượng hữu hạn mức lượng tử. Giả sử có  $L$  mức được dùng để biểu  $f$ . Quá trình gán một giá trị  $f$  cho một trong  $L$  mức gọi là lượng tử hoá biên độ hay gọi tắt là lượng tử hoá. Nếu mỗi đại lượng vô hướng được lượng tử hoá một cách độc lập thì quá trình gọi là lượng tử hoá vô hướng. Nếu hai hoặc trên hai đại lượng vô hướng kết hợp cùng lượng tử hoá thì quá trình gọi là lượng tử hoá vectơ hay lượng tử hoá khối.

Gọi  $\hat{f}$  là  $f$  đã được lượng tử hoá.

$$\hat{f} = Q(f) = r_i \quad ; \quad d_{i-1} < f < d_i \quad (4.1)$$

$Q$ =thuật toán lượng tử hoá.

$r_i$  = với  $1 \leq i \leq L$  là L mức lượng tử.

$d_i$  = với  $0 \leq i \leq L$  là L mức quyết định hay L bờ quyết định.

Theo (4.1) thì nếu  $f$  rơi vào giữa  $d_{i-1}$  và  $d_i$  thì nó được ánh xạ vào mức lượng tử  $r_i$ .

Nếu ta đã xác định các mức quyết định và mức lượng tử thì quá trình lượng tử hoá  $f$  là một quá trình xác định. Cũng có thể biểu diễn :

$$\hat{f} = Q(f) = f + e_Q \quad (4.2)$$

Trong đó  $e_Q$  là sai số lượng tử tính theo :

$$e_Q = \hat{f} - f \quad (4.3)$$

Sai số lượng tử hoá  $e_Q$  còn gọi là tạp âm lượng tử. Đại lượng  $e_Q^2$  coi như trường hợp đặc biệt của độ đo độ méo  $d(f, \hat{f})$  là một độ đo khoảng cách giữa  $f$  và  $\hat{f}$ . Những ví dụ khác của  $d(f, \hat{f})$  bao gồm  $|\hat{f} - f|$  và  $\|\hat{f}^p - |f|^p\|$ .

Các mức lượng tử và mức quyết định thường được xác định bằng cách tối thiểu hoá một tiêu chuẩn sai số nào đó dựa trên  $d(f, \hat{f})$  chẳng hạn như độ méo trung bình D :

$$D = E[d(f, \hat{f})] = \int_{f_0 = -\infty}^{\infty} d(f_0, \hat{f}) p_f(f_0) df_0 \quad (4.4)$$

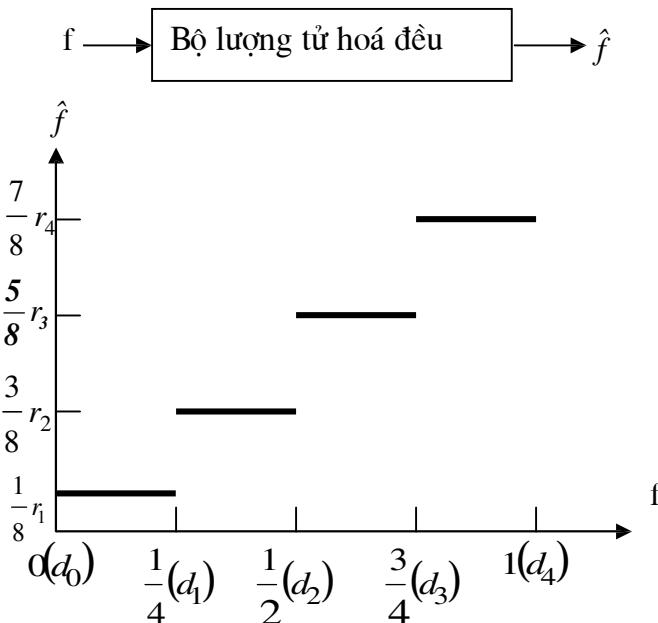
Phương pháp lượng tử hoá chân phương nhất là lượng tử hoá đều trong đó các mức lượng tử (và mức quyết định) cách đều nhau.

$$d_i - d_{i-1} = \Delta \quad 1 \leq i \leq L \quad (4.5a)$$

$$r_i = \frac{d_i + d_{i-1}}{2} \quad 1 \leq i \leq L \quad (4.5b)$$

$\Delta$  là kích thước bước nhảy bằng khoảng cách giữa hai mức lượng tử kề nhau hay hai mức quyết định kề nhau. Ví dụ về lượng tử hoá đều với  $L=4$  và  $f$  giả thiết gồm giữa 0 và 1 được trình bày ở hình 4.3. Tạp âm lượng tử  $e_Q$  thường phụ thuộc tín hiệu. Chẳng hạn tạp âm lượng tử  $e_Q$  của bộ lượng tử hoá đều (trong hình 4.3) được biểu diễn ở hình 4.4. Từ hình này thấy rằng  $e_Q$  là hàm của  $f$  và do đó nó phụ thuộc tín hiệu. Có thể làm cho tạp âm lượng tử  $e_Q$  trong bộ lượng tử hoá đều trở thành không tương quan bằng cách dùng kỹ

thuật giả tạp âm của Robert. Như sẽ thấy trong tiết 3 phép giải tương quan của nhiễu lượng tử hoá sẽ hữu dụng trong việc cải thiện chất lượng hệ mã hoá ảnh. Nó làm thay đổi đặc tính của sự xuống cấp ảnh mã hoá. Ngoài ra có thể làm giảm tạp âm lượng tử đã giải tương quan bằng cách dùng algorit phục hồi ảnh như chương 3.



**Hình 4.3 :** Ví dụ về bộ lượng tử hoá đều. Số mức lượng tử là 4,  $f$  nằm giữa 0 và 1,  $\hat{f}$  là  $f$  đã lượng tử hoá. Các mức lượng tử và bờ quyết định được ký hiệu là  $r_i$  và  $d_i$ .

Tuy lượng tử hoá đều là cách tiếp cận tự nhiên nhất, nhưng nó không phải là tối ưu. Giả sử  $f$  tập trung ở một vùng nào đó nhiều hơn ở các vùng khác. Như vậy gán nhiều mức lượng tử cho vùng đó nhiều hơn các vùng khác là hợp lý. Ta xem lại ví dụ ở hình 4.3. Nếu  $f$  ít khi rơi vào giữa  $d_0$  và  $d_1$  thì mức lượng tử  $r_1$  ít được sử dụng. Sắp xếp các mức lượng tử  $r_1, r_2, r_3$ , và  $r_4$  sao cho chúng đều nằm giữa  $d_1$  và  $d_4$  sẽ có ý nghĩa hơn. Lượng tử hoá mà các mức lượng tử và mức quyết định không cách đều gọi là lượng tử hoá không đều. Việc xác định tối ưu  $r_i$  và  $d_i$  phụ thuộc vào tiêu chuẩn sai số được sử dụng.

Tiêu chuẩn thường dùng nhất là sai số quân phương tối thiểu MMSE\* - giả thiết  $f$  là một biến ngẫu nhiên có hàm mật độ xác suất là  $p_f(f_0)$ . Dùng tiêu chuẩn MMSE ta xác định  $r_k$  và  $d_k$  bằng cách tối thiểu hoá độ méo trung bình D, với :

$$\begin{aligned}
 D &= E[d(\hat{f} - f)] = E[e_Q^2] = E[(\hat{f} - f)^2] \\
 &= \int_{f_0 = -\infty}^{\infty} p_f(f_0) (\hat{f} - f_0)^2 df_0
 \end{aligned} \tag{4.6}$$

Lưu ý rằng  $\hat{f}$  là một trong  $L$  mức lượng tử tính theo (4.1), ta có thể đếm (4.6) viết ra :

$$D = \sum_{i=1}^L \int_{f_0 = d_{i-1}}^{d_i} p_f(f_0) (r_i - f_0)^2 df_0 \tag{4.7}$$

Để tìm cực tiểu  $D$  :

$$\begin{aligned}
 \frac{\partial D}{\partial r_k} &= 0 & 1 \leq k \leq L \\
 \frac{\partial D}{\partial d_k} &= 0 & 1 \leq k \leq L-1 \\
 d_0 &= -\infty \\
 d_L &= \infty
 \end{aligned} \tag{4.8}$$

Từ (4.7) và (4.8) :

$$r_k = \frac{\int_{f_0 = d_{k-1}}^{d_k} f_0 p_f(f_0) df_0}{\int_{f_0 = d_{k-1}}^{d_k} p_f(f_0) df_0}, \quad 1 \leq k \leq L \tag{4.9a}$$

$$d = \frac{r_k + r_{k+1}}{2}, \quad 1 \leq k \leq L-1 \tag{4.9b}$$

$$d_0 = -\infty \tag{4.9c}$$

$$d_L = \infty \tag{4.9d}$$

Phương trình đầu trong (4.9) nói lên rằng mức lượng tử  $r_k$  là tâm quay (centroid) của  $p_f(f_0)$  trong khoảng  $d_{k-1} \leq f \leq d_k$ .

Những phương trình còn lại nói lên rằng mức quyết định  $d_k$  (trừ  $d_0$  và  $d_L$ ) là điểm chính giữa hai mức lượng tử  $r_k$  và  $r_{k+1}$ . Phương trình (4.9) là bộ phương trình cần cho lời giải tối ưu. Với một số hàm mật độ xác suất, trong đó có các mật độ : đều, Gauss, và Laplace, thì (4.9) cũng là bộ phương trình đủ.

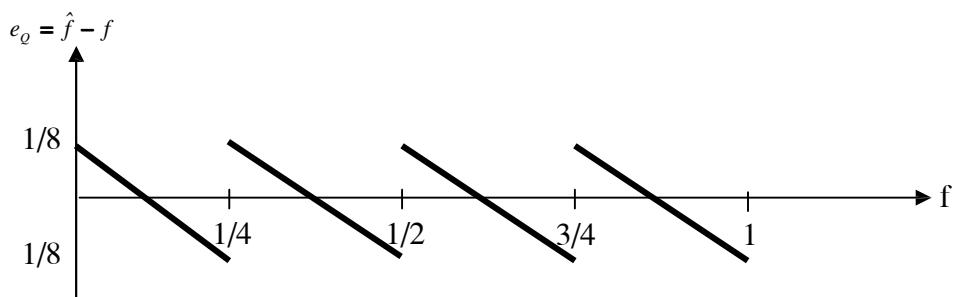
Giải (4.9) là một bài toán phi tuyến. Bài toán phi tuyến đã được giải cho một số hàm mật độ xác suất. Các lời giải khi  $p_f(f_0)$  là : đều, Gauss, Laplace, như trên bảng 1.

Bộ lượng tử hoá dựa trên tiêu chuẩn MMSE được gọi là bộ lượng tử hoá Lloyd\_Max.

Theo bảng 1, bộ lượng tử hoá đều là bộ lượng tử hoá MMSE tối ưu khi  $p_f(f_0)$  là hàm mật độ xác suất đều.

Với những mật độ xác suất khác, lời giải tối ưu là một bộ lượng tử hoá không đều. Hình 4.5 biểu diễn các mức lượng tử và mức quyết định tối ưu ứng với hàm mật độ xác suất Gauss có phương sai là 1 và  $L=4$ .

Cân đánh giá mức độ cải thiện mà bộ lượng tử hoá MMSE tối ưu đem lại so với bộ lượng tử hoá đều. Chẳng hạn xét một hàm độ xác suất Gauss có giá trị trung bình là 0 và phương sai là 1.



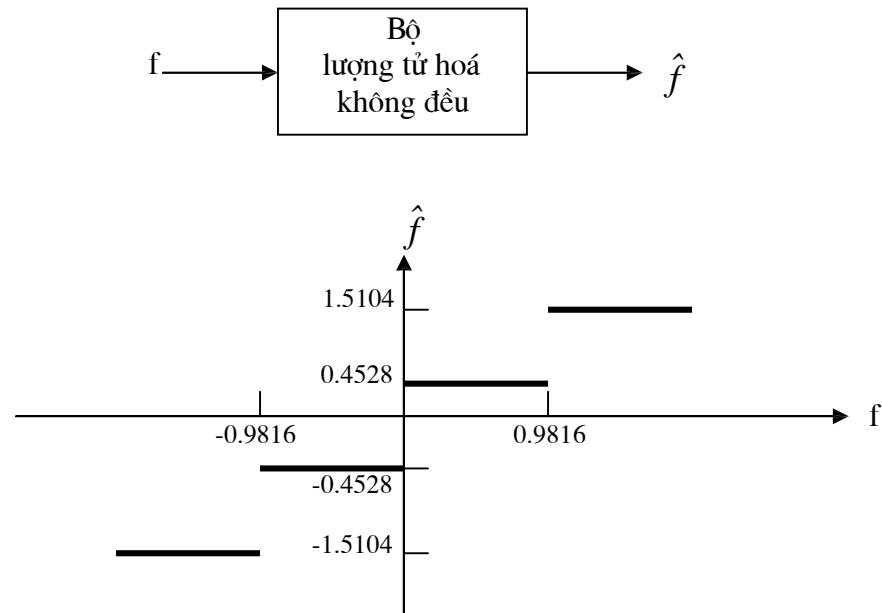
**Hình 4.4 :** Minh họa về sự phụ thuộc của tần số lượng tử vào tín hiệu.

Hình 4.6 biểu diễn méo trung bình D theo hàm của số mức lượng tử, đường liên kết ứng với bộ lượng tử hoá MMSE tối ưu, đường vẽ chấm ứng với bộ lượng tử hoá đều, trong đó các mức lượng tử  $r_i$  được chọn đối xứng đối với gốc toạ độ, các mức quyết định cực tiểu và cực đại giả thiết là  $-\infty$  và  $\infty$ , bước lượng tử  $\Delta$  được chọn để độ méo trung bình D là cực tiểu.

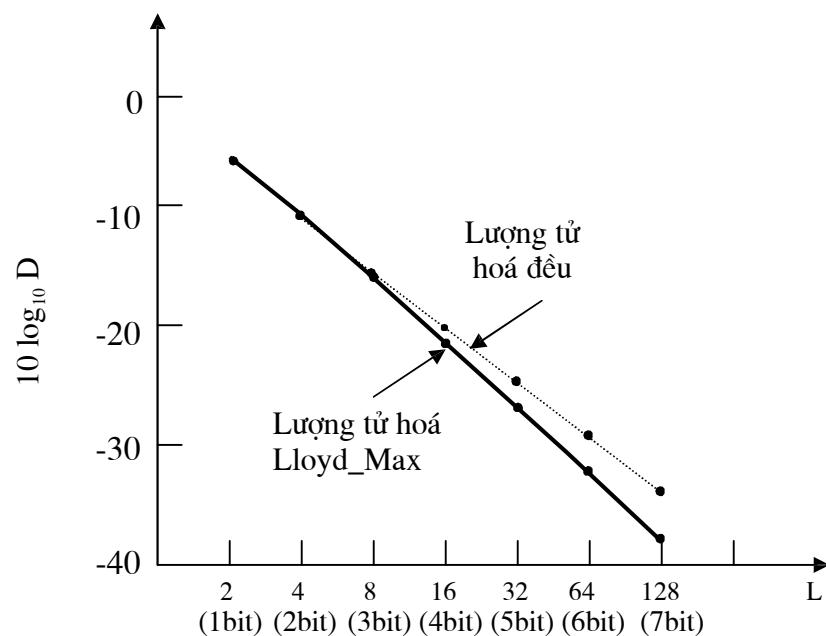
**Bảng 4.1** . Vị trí của các mức lượng tử và quyết định đối với bộ lượng tử hoá Lloyd\_Max. Với hàm mật độ xác suất đều, giả thiết  $p_f(f_0)$  đều giữa  $-1$  và  $1$ . Với hàm mật độ xác suất Gauss giả thiết trung vị bằng  $0$  và phương sai bằng  $1$ . Với hàm mật độ xác suất Laplace

$$p_f(f_0) = \frac{\sqrt{2}}{2\sigma} e^{-\frac{\sqrt{2}|f_0|}{\sigma}} \quad \text{với } \sigma = 1$$

Bit	Đều		Gauss		Laplace	
	$r_i$	$d_i$	$r_i$	$d_i$	$r_i$	$d_i$
1	-0.5000	-1.0000	-0.7979	$-\infty$	-0.7071	$-\infty$
	0.5000	0.0000	0.7979	0.0000	0.7071	0.0000
2		1.0000		$\infty$		$\infty$
	-0.7500	-1.0000	-1.5104	$-\infty$	-1.8304	$-\infty$
	-0.2500	-0.5000	-0.4528	-0.9816	-0.4198	-1.1269
	0.2500	0.0000	0.4528	0.0000	0.4198	0.0000
	0.7500	0.5000	1.5104	0.9816	1.8340	1.1269
3		1.0000		$\infty$		$\infty$
	0.8750	-1.0000	-2.1519	$-\infty$	-3.0867	$-\infty$
	-0.6250	-0.7500	-1.3439	-1.7479	-1.6725	-2.3796
	-0.3750	-0.5000	-0.7560	-1.0500	-0.8330	-1.2527
	-0.1250	-0.2500	-0.2451	-0.5005	-0.2334	-0.5332
	0.1250	0.0000	0.2451	0.0000	0.2334	0.0000
	0.3750	0.2500	0.7560	0.5005	0.8330	0.5332
	0.6250	0.5000	1.3439	1.0500	1.6725	1.2527
	0.8750	0.7500	2.1519	1.7479	3.0867	2.3769
		1.0000				
4	-0.9375	-1.0000		$\infty$		$\infty$
	-0.8125	-0.8750	-2.7326	$-\infty$	-4.4311	$-\infty$
	-0.6875	-0.7500	-2.0690	-2.4008	-3.0169	3.7240
	-0.5625	-0.6250	-1.6180	-1.8435	-2.1773	-2.5971
	-0.4375	-0.5000	-1.2562	-1.4371	-1.5778	-1.8776
	-0.3125	-0.3750	-0.9423	-1.0993	-1.1110	-1.3444
	-0.1875	-0.2500	-0.6568	-0.7995	-0.7287	-0.9198
	-0.0625	-0.1250	-0.3880	-0.5224	-0.4048	-0.5667
	0.0625	0.0000	-0.1284	-0.2582	-0.1240	-0.2664
	0.1875	0.1250	0.1284	0.0000	0.1240	0.0000
	0.3125	0.2500	0.3880	0.2582	0.4048	0.2644
	0.4375	0.3750	0.6568	0.5224	0.7287	0.5667
	0.5625	0.5000	0.9423	0.7995	1.1110	0.9198
	0.6875	0.6250	1.2562	1.0993	1.5778	1.3444
	0.8125	0.7500	1.6180	1.4371	2.1773	1.8776
	0.9375	0.8750	2.0690	1.8435	3.0169	2.5971
		1.0000	2.7326	2.4008	4.4311	3.7240
				$\infty$		$\infty$



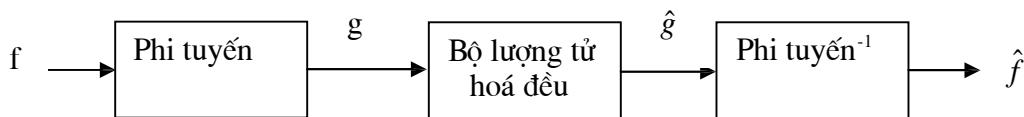
**Hình 4.5.** Ví dụ về bộ lượng tử hoá Lloyd\_Max. Số mức lượng tử là 4, hàm mật độ xác suất là Gauss với trung vị bằng 0 và phương sai bằng 1.



**Hình 4.6.** So sánh độ méo trung bình  $D = E[(\hat{f} - f)^2]$  theo hàm của số mức lượng tử  $L$  trong 2 trường hợp :

- Đường liền nét : bộ lượng tử hoá Lloyd\_Max (khi hàm mật độ xác suất là Gauss, trung vị bằng 0 và phương sai bằng 1).
- Đường vẽ chấm : bộ lượng tử hoá đều. Trục tung tính theo  $10 \log_{10} D$ .

Trên hình 4.6 nếu dùng từ mã có độ dài đều để biểu diễn các mức lượng tử t hì sự tiết kiệm bit là  $0 \sim 1/2$  bit khi L trong khoảng 2 (1 bit) và 128 (7 bit). Trong ví dụ này giả thiết hàm mật độ xác suất  $p_f(f_0)$  là Gauss. Có thể tiến hành phân tích tương tự với các hàm mật độ xác suất khác, hàm mật độ xác suất càng khác xa hàm phân b ố đều thì ưu thế của lượng tử hoá không đều so với lượng tử hoá đều càng lớn. Quan niệm : “ *bộ lượng tử hoá đều là tối ưu khi hàm mật độ xác suất phân bố đều* ” lại gợi ý cho ta một cách tiếp cận khác. Đó là, ta có thể ánh xạ  $f$  vào  $g$  bằng một phép phi tuyến sao cho  $p_g(g_0)$  là đều, ta đem lượng tử hoá  $g$  bằng một bộ lượng tử hoá đều, sau đó lại thực hiện phép ánh xạ ngược. Phương pháp này được minh họa trên hình 4.7.



**Hình 4.7.** Lượng tử hoá không đều bằng phép nén -dãn.

Phép phi tuyến này được gọi là phép nén -dãn (companding). Theo lý thuyết xác suất, một lựa chọn của phép phi tuyến (hay phép nén -dãn)  $C[\bullet]$  để tạo ra được  $p_g(g_0)$  đồng đều là :

$$g = C[f] = \int_{x=-\infty}^f p_f(x) dx^{-\frac{1}{2}} \quad (4.10)$$

$p_g(g_0)$  nhận được đồng đều trong khoảng  $-1/2 \leq g \leq 1/2$ .

Tuy (1.10) dễ giải hơn hệ phương trình phi tuyến (1.9), hệ ở hình 1.7 lại tối thiểu hoá  $D'$  :

$$D' = E[(\hat{g} - g)^2] \quad (4.11)$$

mà méo  $D'$  ở (4.11) không giống  $D$  ở (4.6).

Trong tiết này ta đã xét việc lượng tử hoá một đại lượng vô hướng  $f$ . Trong mã hoá ảnh, phải lượng tử hoá nhiều đại lượng vô hướng. Một cách tiếp cận là lượng tử hoá từng cái độc lập \_ Cách này gọi là lượng tử hoá vô hướng một nguồn vectơ. Giả sử có N vô hướng  $f_i$  với  $1 \leq i \leq N$  và mỗi vô hướng được lượng tử hoá ra  $L_i$  mức. Nếu  $L_i$  được biểu diễn bằng một luỹ thừa của 2 và nếu mỗi mức lượng tử được mã hoá với một số bit như nhau (nghĩa là với từ mã có độ dài đều) thì quan hệ giữa  $L_i$  với một số bit cần thiết  $B_i$  là :

$$L = 2^{\sum_{i=1}^N B_i} \quad (4.12a)$$

$$B_i = \log_2 L_i \quad (4.12b)$$

Tổng số bit B cần thiết để mã hoá N vô hướng là :

$$B = \sum_{i=1}^N B_i \quad (4.13)$$

Từ (4.12) và (4.13) được tổng số mức lượng tử L :

$$L = \prod_{i=1}^N L_i = 2^B \quad (4.14)$$

Xét (4.13) và (4.14) nhận thấy tổng số bit B là tổng các  $B_i$  còn tổng số mức lượng tử L là tích các  $L_i$ . Nếu có một số bit cố định B để mã hoá N vô hướng bằng phép lượng tử hoá vô hướng nguồn vectơ thì phải phân phối B cho N vô hướng. Chiến lược tối ưu để phân bổ bit phụ thuộc tiêu chuẩn sai số và hàm mật độ xác suất của các vô hướng. Chiến lược tối ưu thường dùng là cho vô hướng có phương sai lớn nhiều bit, vô hướng có phương sai bé ít bit. Ví dụ : giả sử cần tối thiểu hóa sai số quân phương

$$\sum_{i=1}^N E_i (f_i - f_i^*)^2$$

đối với  $B_i$  ( $1 \leq i \leq N$ ) trong đó  $f_i^*$  là kết quả lượng tử hoá  $f_i$ . Nếu các vô hướng có hàm mật độ xác suất giống nhau chỉ có phương sai khác nhau ta sẽ dùng một phương pháp lượng tử hoá như nhau, chẳng hạn dùng bộ lượng tử hoá Lloyd\_Max cho từng vô hướng. Khi đó lời giải gần đúng về phân bổ bit là:

$$B_i = \frac{B}{N} + \frac{1}{2} \log_2 \frac{\sigma_i^2}{\left[ \prod_{j=1}^N \sigma_j^2 \right]^{1/N}} \quad 1 \leq i \leq N \quad (4.15)$$

Trong đó  $\sigma_i^2$  là phương sai của vô hướng  $f_i$ . Từ (4.15) suy ra :

$$L_i = 2^{B_i} = 2^{B/N} \frac{\sigma_i}{\left( \prod_{j=1}^N \sigma_j \right)^{1/N}} \quad 1 \leq i \leq N \quad (4.16)$$

Theo (4.16) số mức lượng tử cho  $f_i$  tỉ lệ với  $\sigma_i$ , là độ lệch chuẩn của  $f_i$ . Tuy (4.15) là một lời giải gần đúng với một số giả thiết nhất định, nó vẫn là căn cứ tham khảo trong những bài toán phân bổ bit.  $B_i$  trong (4.15) có thể âm và nói chung không phải là số nguyên. Khi lượng tử hoá vô hướng  $B_i$  phải là một số nguyên không âm. Đó là điều kiện ràng buộc khi giải các bài toán phân bổ bit trong thực tế.

## 1.2 . Lượng tử hoá vectơ.

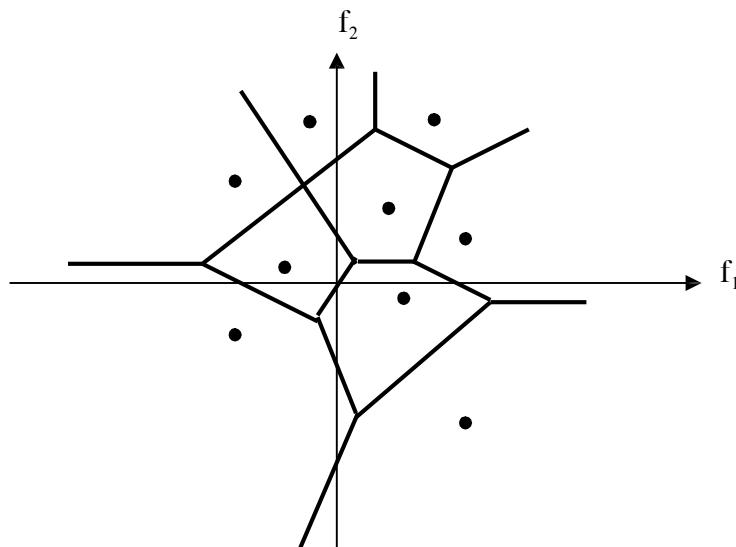
Trong tiết trên, thảo luận về lượng tử hoá vô hướng một vô hướng và một nguồn vectơ. Một cách tiếp cận khác để mã hoá nguồn vectơ là chia các vô hướng thành những khối, xem mỗi khối như một đơn vị sau đó lượng tử đồng thời những vô hướng này trong đơn vị đó. Như vậy gọi là lượng tử hoá vectơ hay lượng tử hoá khối.

Gọi  $f = [f_1, f_2, \dots, f_N]^T$  là một vectơ  $M$  chiều gồm  $N$  vô hướng  $f_i$  có giá trị thực, biên độ liên tục. Trong phép lượng tử hoá vectơ  $f$  được ánh xạ vào một vectơ  $M$  chiều khác  $r = [r_1, r_2, \dots, r_N]^T$ . Khác với  $f$  mà các phần tử có biên độ liên tục, vectơ  $r$  được chọn từ  $L$  mức lượng tử. Gọi  $\hat{f}$  là  $f$  đã được lượng tử hoá, ta biểu diễn nó bằng :

$$\hat{f} = VQ(f) = r_i f \in C_i \quad (4.17)$$

$VQ$  là toán tử lượng tử hoá vectơ  $r_i$  với  $1 \leq i \leq N$  chỉ  $L$  mức lượng tử và  $C_i$  được gọi là tế bào thứ  $i$ . Nếu  $f$  nằm trong tế bào  $C_i$ , thì  $f$  được ánh xạ vào  $r_i$ . Hình 4.8 cho một ví dụ lượng tử hoá vectơ khi  $N=2$  và  $L=9$ . Các chấm trên hình là những mức lượng tử, và các đường liền nét là đường biên tế bào.

Trong lượng tử hoá vectơ tế bào có thể có hình dạng, kích thước bất kỳ. Đó là điều khác biệt với lượng tử hoá vô hướng, mà tế bào (miền gữa 2 mức quyết định kề nhau) có thể có kích thước bất kỳ nhưng hình dạng cố định.



**Hình 4.8** . Ví dụ lượng tử hoá vectơ. Số vô hướng trong mỗi vectơ là 2, số mức lượng tử là 9.

Phép lượng tử hoá vectơ khai thác sự mềm dẻo này. Cũng như trong trường hợp vô hướng, ta định nghĩa độ méo  $d(f, \hat{f})$  là độ đo sự chênh lệch giữa  $f$  và  $\hat{f}$ . Một ví dụ của  $d(f, \hat{f})$  là  $e_Q^T e_Q$  trong đó tạp âm lượng tử  $e_Q$  định nghĩa theo :

$$e_Q = \hat{f} - f = VQ(f) - f \quad (4.18)$$

Các mức lượng tử  $r_I$  và bờ các tế bào  $C_I$  xác định bằng cách lấy cực tiểu 1 tiêu chuẩn sai số nào đó, chẳng hạn độ méo trung bình D :

$$D = E[d(f, \hat{f})] \quad (4.19)$$

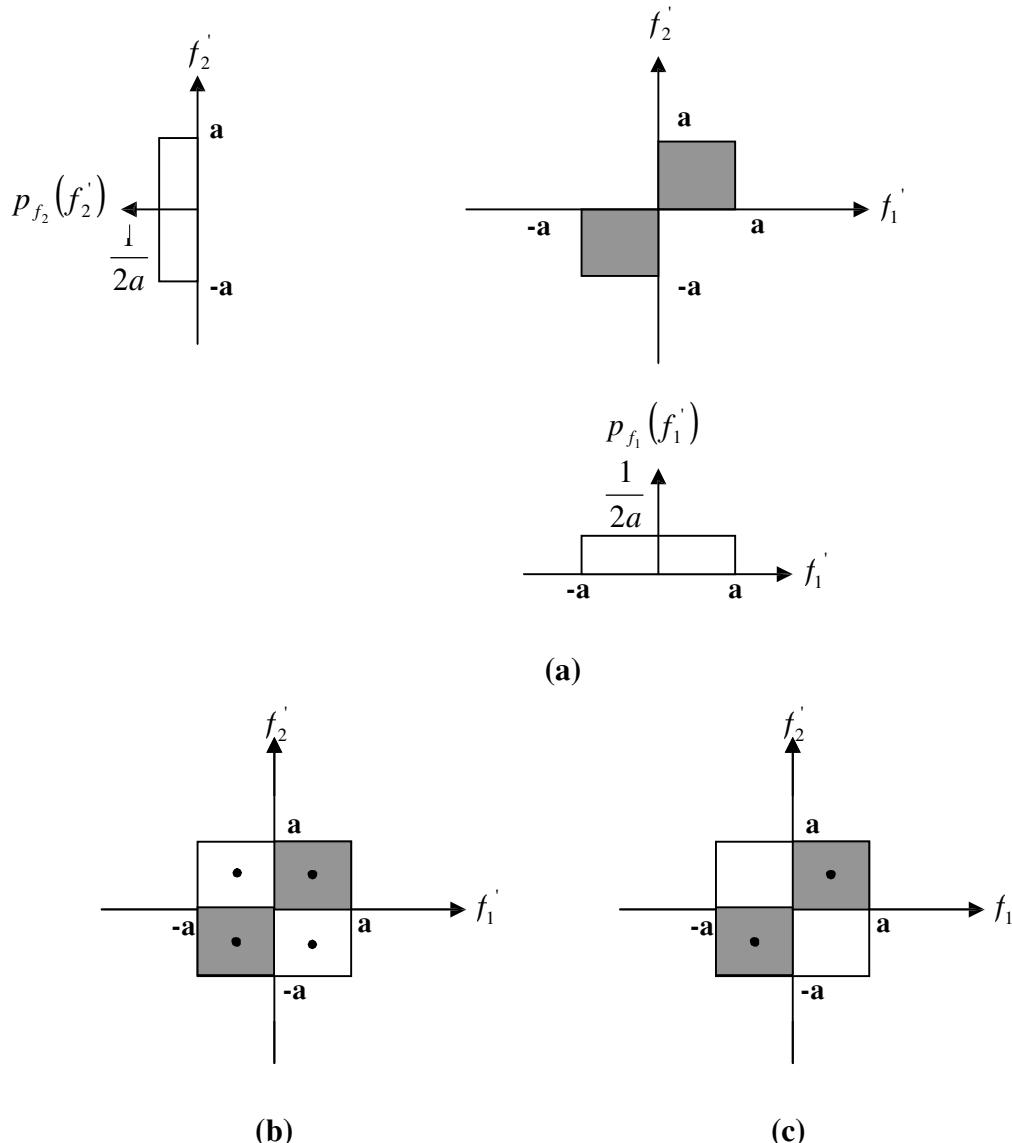
Nếu  $d(f, \hat{f})$  là  $e_Q^T e_Q$  thì từ (4.18) và (4.19) suy ra :

$$\begin{aligned} D &= E(e_Q^T e_Q) = E[(\hat{f} - f)^T (\hat{f} - f)] \\ &= \int \int \dots \int_{-\infty}^{\infty} (\hat{f} - f_0)^T (\hat{f} - f_0) p_f(f_0) df_0 \\ &= \sum_{i=1}^L \int \int \dots \int_{f_0 \in C_i} (r_i - f_0)^T (r_i - f_0) df_0 \end{aligned} \quad (4.20)$$

Độ méo trung bình ở (4.20) là sai số quân phương MSE và là dạng tổng quát của (4.7) .

Ưu điểm của  $e_Q^T e_Q$  so với lượng tử hoá vô hướng một nguồn vectơ là cải thiện chất lượng. Lượng tử hoá vectơ cho phép giảm thấp độ méo trung bình D khi giữ số mức lượng tử không đổi, hay cho giảm số mức lượng tử khi giữ độ méo trung bình D không đổi. Lượng tử hoá vectơ cải thiện chất lượng so với lượng tử hoá vô hướng bằng nhiều cách. Cách có ý nghĩa nhất là khai thác mối quan hệ thống kê giữa các vô hướng trong cùng khối.

Để minh họa việc lượng tử hoá vectơ có thể khai thác mối quan hệ thống kê ta hãy xét 2 ví dụ. Trong ví dụ thứ nhất ta khai thác mối quan hệ tuyến tính (tính tương quan). Xét 2 nguồn ngẫu nhiên  $f_1$  và  $f_2$  có hàm mật độ xác suất đồng thời  $p_{f_1, f_2}(f_1, f_2)$  như trên hình 4.9a. Hàm mật độ xác suất đồng thời có biên độ đồng đều và bằng  $1/2a^2$  trong vùng gạch chéo, bằng không ở ngoài vùng gạch chéo. Hai hàm mật độ xác suất biên  $p_{f_1}(f_1)$  và  $p_{f_2}(f_2)$  cũng được vẽ trên hình. Vì  $E[f_1 f_2] \neq E[f_1] E[f_2]$  nên  $f_1$  và  $f_2$  là tương quan hay phụ thuộc tuyến tính. Giả thiết ta lượng tử hoá riêng rẽ  $f_1$  và  $f_2$ , dùng lượng tử hoá vô hướng và tiêu chuẩn MMSE.



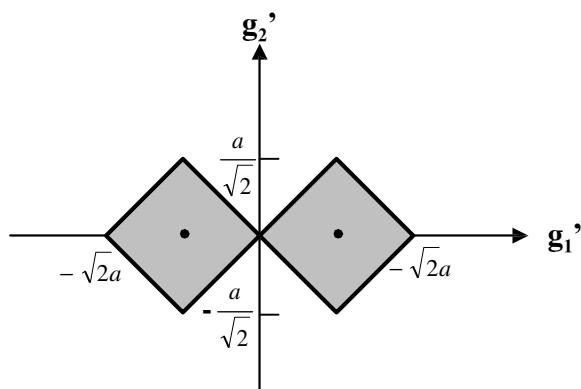
**Hình 4.9.** Minh họa việc lượng tử hóa vectơ khai thác sự phụ thuộc tuyến tính của các vô hướng trong vectơ :

- (a) Hàm mật độ xác suất  $p_{f_1 f_2}(f_1, f_2)$
- (b) Các mức lượng tử hóa (các chấm trên hình) khi lượng tử hóa vô hướng.
- (c) Các mức lượng tử hóa (các chấm trên hình) khi lượng tử hóa vectơ.

Vì mỗi vô hướng  $f_1$  và  $f_2$  đều có hàm mật độ xác suất đều nên bộ lượng tử hóa vô hướng tối ưu là lượng tử hóa đều. Nếu ta cho mỗi vô hướng có 2 mức lượng tử thì các mức lượng tử của mỗi vô hướng là  $a/2$  và  $-a/2$ . Bốn (2x2) mức lượng tử hợp thành 4

chấm trên hình 4.9b. Rõ ràng là 2 trong số 4 mức lượng tử là lãng phí. Với phép lượng tử hoá vectơ ta chỉ có thể dùng 2 mức lượng tử như trên hình 4.9c. Ví dụ này cho thấy rằng lượng tử hoá vectơ cho phép giảm số mức lượng tử mà không phải hi sinh MSE. Ta có thể loại bỏ sự phụ thuộc tuyến tính giữa  $f_1$  và  $f_2$  bằng cách đem quay hàm mật độ xác suất đi  $45^\circ$  theo chiều kim đồng hồ, kết quả của phép biến đổi toạ độ tuyến tính khả nghịch này được biểu diễn trên hình 4.10.

Trong hệ toạ độ mới  $g_1$  và  $g_2$  không tương quan, vì  $E[g_1, g_2] = E[g_1] E[g_2]$ . Trong hệ toạ độ mới này có thể đặt hai mức lượng tử vào các chấm ở trên hình bằng cách lượng tử hoá vô hướng hai đại lượng vô hướng, và khi đó ưu thế của lượng tử hoá vectơ không còn nữa.



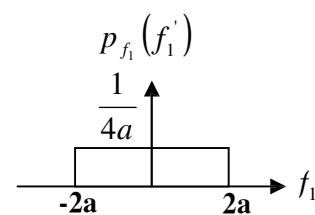
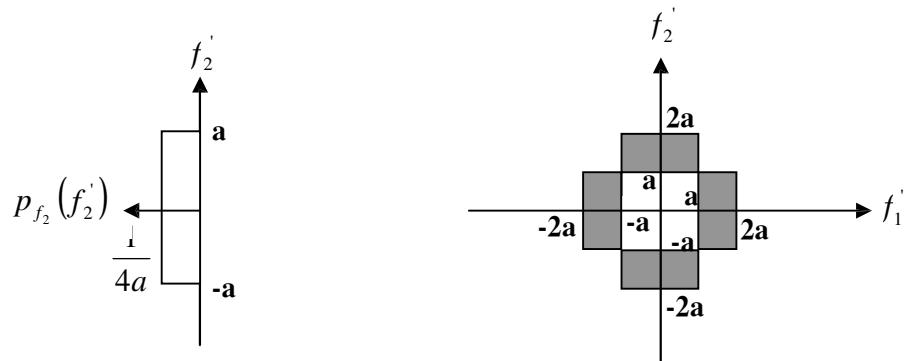
**Hình 4.10.** Kết quả loại trừ sự phụ thuộc tuyến tính giữa hai vô hướng  $f_1$  và  $f_2$  ở hình 4.9 khi thực hiện phép biến đổi tuyến tính  $f_1$  và  $f_2$ .

Loại bỏ sự phụ thuộc tuyến tính làm mất ưu thế của phép lượng tử hoá vectơ. Như vậy là phù hợp với quan điểm cho rằng lượng tử hoá vectơ có thể khai thác sự phụ thuộc tuyến tính giữa các vô hướng trong vectơ.

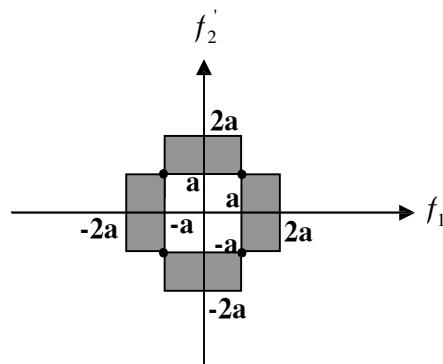
Phép lượng tử hoá vectơ cũng có thể khai thác sự phụ thuộc phi tuyến. Ta đưa ra một ví dụ minh họa.

Xét 2 biến ngẫu nhiên  $f_1$  và  $f_2$  và hàm mật độ xác suất đồng thời  $p_{f_1, f_2}(f_1, f_2)$  được biểu diễn trên hình 4.11a. Hàm mật độ xác suất vẫn là đều với biên độ bằng  $1/(8a^2)$  trong vùng gạch chéo và bằng không ngoài vùng đó. Hàm mật độ xác suất biên  $p_{f_1}(f_1)$  và  $p_{f_2}(f_2)$  cũng được vẽ trên hình 4.11a. Từ hàm mật độ xác suất đồng thời  $E[f_1, f_2] = E[f_1] E[f_2]$  và do đó  $f_1$  và  $f_2$  độc lập tuyến tính. Tuy vậy  $p_{f_1, f_2}(f_1, f_2) \neq p_{f_1}(f_1) p_{f_2}(f_2)$  nên  $f_1$  và  $f_2$  phụ thuộc thống kê.

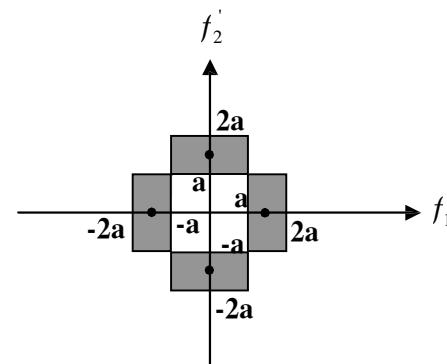
Khi mà các biến ngẫu nhiên độc lập tuy nhiên tính nhung phụ thuộc thống kê ta bảo chúng phụ thuộc phi tuyến.



(a)



(b)

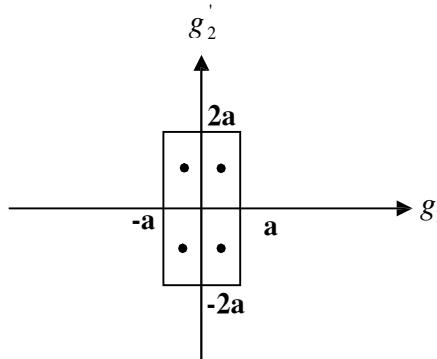


(c)

**Hình 4.11.** Minh họa việc lượng tử hoá vectơ khai thác sự phụ thuộc tuyến tính giữa các vô hướng trong vectơ :

- Hàm mật độ xác suất  $p_{f_1 f_2}(f_1, f_2)$ .
- Các mức lượng tử (các chấm) khi lượng tử hoá vô hướng .
- Các mức lượng tử (các chấm) khi lượng tử hoá vectơ.

Nếu ta lượng tử hoá  $f_1$  và  $f_2$  riêng rẽ, dùng tiêu chuẩn MSE và cho mỗi vô hướng 2 mức lượng tử, thì các mức lượng tử tối ưu cho mỗi vô hướng là  $-a$  và  $a$ . Các mức lượng tử tổng hợp trong trường hợp đó là 4 chấm trong hình 4.11b. Độ méo trung bình  $D = E[e_Q^T e_Q]$  trong ví dụ này là  $5a^2/12$ . Ví dụ này cho thấy rằng dùng lượng tử hoá vectơ có thể làm giảm MSE mà không cần tăng số mức lượng tử. Ta có thể loại bỏ sự phụ thuộc phi tuyến giữa  $f_1$  và  $f_2$  trong ví dụ này bằng một thuật toán phi tuyến khả nghịch. Kết quả của một thuật toán như vậy được biểu diễn trên hình 4.12.



**Hình 4.12.** Kết quả của việc loại bỏ sự phụ thuộc tuyến tính giữa hai vô hướng  $f_1$  và  $f_2$  ở hình 4.11.

Vì  $p_{g_1 g_2}(g_1, g_2) = p_{g_1}(g_1)p_{g_2}(g_2)$  nên  $g_1$  và  $g_2$  độc lập thống kê. Trong những trường hợp này có thể đặt hai mức lượng tử vào các chấm trên hình bằng cách lượng tử hoá vô hướng hai đại lượng vô hướng, và ưu thế của lượng tử hoá vectơ không còn nữa. Qua ví dụ này thấy rằng loại bỏ sự phụ thuộc phi tuyến làm giảm ưu thế của lượng tử hoá vectơ. Như vậy phù hợp với quan niệm cho rằng lượng tử hoá vectơ có thể khai thác sự phụ thuộc phi tuyến giữa các vô hướng trong vectơ.

Phép biến đổi tuyến tính bao giờ cũng có thể loại bỏ sự phụ thuộc tuyến tính. Về sự phụ thuộc phi tuyến đôi khi ta cũng loại bỏ được bằng một thuật toán phi tuyến khả nghịch. Nếu ta loại bỏ sự phụ thuộc tuyến tính hay phi tuyến bằng một thuật toán phi tuyến khả nghịch trước khi lượng tử hoá thì ưu thế của lượng tử hoá vectơ về khả năng khai thác sự phụ thuộc tuyến tính hay phi tuyến sẽ không còn nữa. Như vậy là phải hết sức chú ý đến quan hệ chặt chẽ giữa giai đoạn biến đổi và giai đoạn lượng tử hoá trong mã hoá ảnh. Nếu như giai đoạn biến đổi làm mất sự phụ thuộc tuyến tính hay phi tuyến giữa các vô hướng cần mã hoá thì đến giai đoạn lượng tử hoá mức độ cải thiện của lượng tử hoá

vectơ so với lượng tử hoá vô hướng sẽ giảm sút, làm cho lượng tử hoá vectơ trở lên kém hấp dẫn. Điều đó nói lên một phần tại sao trong bộ mã hoá dạng sóng sự cải thiện do lượng tử hoá vectơ đem lại rõ nét hơn trong bộ mã hoá phép biến đổi. Các vô hướng dùng trong bộ mã hoá dạng sóng, chẳng hạn các cường độ ảnh, có tính tương quan cao hơn các vô hướng trong bộ mã hoá phép biến đổi, chẳng hạn các hệ số phép biến đổi DCT. Điều này sẽ được phân tích ở các tiết 3 và 4.

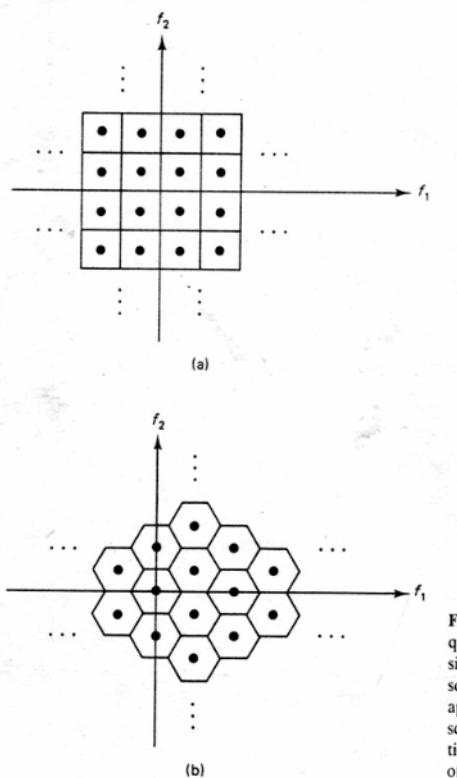
Ngoài việc khai thác sự phụ thuộc thống kê lượng tử hoá vectơ còn có thể khai thác sự tăng thứ nguyên, nghĩa là tăng số vô hướng trong vectơ. Để minh họa ta xét 2 biến ngẫu nhiên  $f_1$  và  $f_2$  có hàm mật độ xác suất đồng thời đều trong một miền hình vuông có diện tích A. Rõ ràng là  $f_1$  và  $f_2$  độc lập thống kê. Giả sử số mức lượng tử L rất lớn và do đó kích thước tế bào nhỏ hơn hình vuông trong đó hàm mật độ xác suất khác 0. Trước hết ta xét phép lượng tử hoá vô hướng  $f_1$  và  $f_2$ . Vì hàm mật độ xác suất của  $f_1$  và  $f_2$  là đều nên theo tiêu chuẩn MMSE, lượng tử hoá đều là tối ưu. Việc lượng tử hoá đều  $f_1$  và  $f_2$  riêng rẽ đưa tới những mức lượng tử và tế bào vẽ ở hình 4.13a. Trong trường hợp lượng tử hoá vô hướng tế bào có dạng hình vuông có cạnh a. Nếu đem lượng tử hoá vectơ  $f_1$  và  $f_2$  thì các mức lượng tử và tế bào như trên hình 4.13b – Tế bào có dạng lục giác. Thông qua tính toán có thể chứng minh rằng MSE trong trường hợp lượng tử hoá vectơ thấp hơn trường hợp lượng tử hoá vô hướng 4% nếu mức lượng tử như nhau. Cũng có thể chứng minh là số mức lượng tử mà lượng tử hoá vectơ yêu cầu bé hơn số mức của lượng tử hoá vô hướng 2% khi MSE như nhau. Sự cải thiện này thường nhỏ hơn nhiều so với mức cải thiện bằng lượng tử hoá vectơ khi khai thác sự phụ thuộc thống kê. Tuy vậy sự cải thiện sẽ nét hơn nhiều khi thứ nguyên (nghĩa là số vô hướng trong vectơ) tăng lên. Lưu ý rằng sự cải thiện thêm này vẫn đạt được ngay cả khi các vô hướng trong khối độc lập thống kê với nhau.

Sự cải thiện mà lượng tử hoá vectơ đem lại trong một số trường hợp cho phép mã hoá 1 vô hướng dưới 1 bit. Nếu ta mã hoá riêng rẽ từng vô hướng và cho mỗi vô hướng tối thiểu 2 mức lượng tử (nếu dùng 1 mức lượng tử thì coi như không mã hoá) thì tỷ lệ bit tối thiểu có thể là 1 bit mỗi vô hướng. Nếu dùng lượng tử hoá vectơ, có thể cho mỗi vô hướng 2 hoặc trên 2 mức lượng tử nếu xét riêng rẽ, nhưng nếu n hìn tổng hợp lại thì tốc độ bit sẽ thấp hơn một bit mỗi vô hướng.

Để minh họa điều này ta trở lại ví dụ hình 4.9. Khi lượng tử hoá vô hướng (hình 4.9b) cho mỗi vô hướng 2 mức lượng tử thì tổng lại cần đến 4 mức cho 2 vô hướng, và tỷ lệ bit là 1 bit cho mỗi vô hướng. Khi lượng tử hoá vectơ (hình 4.10c) ta cho mỗi vô

hướng 2 mức lượng tử khi xét từng vô hướng riêng rẽ và cũng chỉ có 2 mức lượng tử cho cả hai vô hướng. Trong trường hợp này tỷ lệ bit là  $1/2$  bit cho mỗi vô hướng. Nếu ta định mã hoá cường độ pixel 1 và mỗi cường độ pixel tối thiểu phải được biểu diễn bằng 2 mức lượng tử thì phương pháp lượng tử hoá vectơ là cách tiếp cận để cho tỷ lệ bit thấp hơn 1 bit/pixel. Ưu thế của lượng tử hoá vectơ so với lượng tử hoá vô hướng cũng phải trả giá : giá thành về tính toán và lưu trữ khi lượng tử hoá vectơ đắt hơn nhiều so với lượng tử hoá vô hướng .

Phần lớn yêu cầu tính toán và lưu trữ là để thiết kế và lưu trữ sách mã và để tra sách mã để nhận ra mức lượng tử phải gán cho một vectơ.



**Hình 4.13.** Minh họa khả năng khai thác sự tăng thứ nguyên của phép lượng tử. Trong trường hợp này bên lượng tử hoá vectơ ít hơn bên lượng tử hoá vô hướng 4%.

- a) Lượng tử hoá vô hướng  $f_1$  và  $f_2$  .
- b) Lượng tử hoá vectơ  $f_1$  và  $f_2$

### 1.3. Thiết kế sách mã và algorit K-means.

Khi lượng tử hoá vectơ cần xác định các mức lượng tử  $r_i$  và các tế bào  $C_i$ .

Bảng liệt kê các mức lượng tử gọi là sách mã lượng tử hay gọi tắt là sách mã. Nếu trong sách có L mức kượng tử thì ta gọi nó là sách mã L mức. Sách mã ở đầu máy phát dùng để lượng tử hoá một nguồn vectơ thành 1 trong L mức lượng tử, còn ở đầu máy thu dùng để xác định mức lượng tử theo từ mã nhận được. Theo sự thoả thuận trước của bên phát và bên thu, hai bên dùng sách mã như nhau.

Thiết kế sách mã cho lượng tử hoá vectơ là một bài toán khó. Không giống trường hợp lượng tử hoá vô hướng, các mức lượng tử bên lượng tử hoá vectơ là những vectơ, còn bờ các tế bào cũng không còn là điểm nữa. Sự khó khăn về thiết kế sách mã là một lý do để những năm 70 về trước lượng tử hoá vectơ không được xét đến khi mã hoá ảnh và tiếng nói.

Cách xác định  $r_i$  và  $C_i$  tối ưu phụ thuộc tiêu chuẩn sai số được sử dụng. Tiêu chuẩn MSE thường dùng có thể biểu diễn như độ méo trung bình  $D = E[d(f, \hat{f})]$  với  $d(f, \hat{f}) = (\hat{f} - f)^T (\hat{f} - f)$ . Thiết kế tối ưu sách mã là bài toán phi tuyến cao độ. Muốn giải bài toán đó nên khai thác 2 điều kiện cần cho lời giải bài toán sau đây :

Điều kiện 1 . Để 1 vectơ  $f$  có thể lượng tử hoá về 1 trong những mức lượng tử, bộ lượng tử hoá tối ưu phải chọn mức lượng tử  $r_i$  có méo nhỏ nhất giữa  $f$  và  $r_i$  :

$$VQ(f) = r_i \text{ nếu và chỉ nếu } d(f, r_i) \leq d(f, r_j), \quad i \neq j \quad 1 \leq j \leq L \quad (4.21)$$

Điều kiện 2 . Mỗi mức lượng tử  $r_i$  phải tối thiểu hoá được méo trung bình D trong  $C_i$  :

$$\text{Tối thiểu hoá } E[d(f, r_i) | f \in C_i] \text{ đối với } r_i. \quad (4.22)$$

Mức  $r_i$  thoả mãn (4.22) gọi là tâm quay của  $C_i$ . Nếu (4.21) không thoả mãn, thì có thể làm giảm méo trung bình D bằng cách áp đặt (4.21). Nếu (4.22) không thoả mãn, có thể làm giảm D bằng cách áp đặt (4.22). Hai điều kiện trên là điều kiện cần cho lời giải tối ưu.

Điều kiện 1 định ra 1 quy tắc để lượng tử hoá  $f$  mà không sử dụng  $C_i$  một cách tường minh. Nói cách khác độ méo  $d(f, \hat{f})$  cùng với tất cả mức lượng tử  $r_i$ , với  $1 \leq i \leq$

L, định ra mọi tế bào  $C_i$  cho  $1 \leq i \leq L$ . Điều kiện 2 chỉ ra một con đường để xác định  $r_i$  từ  $C_i$  và  $d(f, \hat{f})$ . Hai điều kiện này cho thấy rằng khi đã cho  $d(f, \hat{f})$  thì các mức lượng tử và tế bào không độc lập với nhau. Quả thực là các mức lượng tử định ra các tế bào và các tế bào định ra các mức lượng tử. Do đó chỉ riêng các mức lượng tử cũng đã đủ cho sách mã, các tin tức rõ về các tế bào không cần lưu trữ.

Hai điều kiện cần gởi ý ra một quy trình lặp để thiết kế một sách mã tối ưu. Giả sử ta bắt đầu bằng một ước lượng ban đầu của  $r_i$ . Cho  $r_i$  và độ méo  $d(f, \hat{f})$  ta xác định  $C_i$ , ít ra là bằng lý thuyết, bằng cách xác định giá trị  $r_i$  ứng với mọi giá trị có thể của  $f$ , dùng điều kiện 1 trong (4.21). Cho một ước lượng của  $C_i$  có thể xác định ra  $r_i$  bằng cách tính tâm quay của  $C_i$ , dùng điều kiện 2 trong (4.22). Giá trị  $r_i$  tìm ra là một ước lượng mới của các mức lượng tử và quá trình cứ thế tiếp tục. Quy trình lặp lại có hai khó khăn thực tế.

Trước hết, nó yêu cầu xác định  $r_i$  với mọi giá trị có thể của  $f$ . Thứ nữa, hàm  $p_r(f_\theta)$  cần để tính tâm quay của  $C_i$  thì trong thực tế không cho biết. Thay vào đó chúng ta có những vectơ huấn luyện đại biểu cho số liệu phải mã hoá. Một biến dạng của quy trình lặp trên đây, có tính đến hai khó khăn vừa nói, là algorit K\_means được Lloyd tìm ra năm 1957 cho trường hợp vô hướng ( $N = 1$ ). Năm 1965 Forgy triển khai phương pháp này cho trường hợp vectơ. Algorit này còn gọi là algorit LBG, vì Y.Linde, A.Bufo và R.M.Gray đã chứng minh rằng algorit này có thể dùng với nhiều độ đo cự ly khác nhau và đã được dùng phổ biến trong những ứng dụng lượng tử hoá vectơ vào mã hoá tiếng nói và ảnh.

Để mô tả algorit K\_means ta giả thiết là có  $M$  vectơ huấn luyện biểu thị bởi  $f_i$  cho  $1 \leq i \leq M$ . Bởi vì chúng ta ước lượng  $L$  mức lượng tử từ  $M$  vectơ huấn luyện, ta giả thiết là  $M \gg L$ . Trong những ứng dụng điển hình  $M$  cỡ  $10L$  đến  $50L$  hay hơn. Các mức lượng tử  $r_i$  được xác định bằng cách tối thiểu hoá méo trung bình, định nghĩa theo :

$$D = \frac{1}{M} \sum_{i=1}^M d(f_i, \hat{f}_i) \quad (4.23)$$

Trong algorit K\_means, chúng ta bắt đầu bằng một giá trị ước lượng ban đầu của  $r_i$  khi  $1 \leq i \leq L$ . Sau đó ta xếp lớp  $M$  vectơ huấn luyện thành  $L$  nhóm (hay lớp), mỗi lớp ứng với một mức lượng tử (dùng công thức 21). Điều này có thể thực hiện được bằng cách so sánh một vectơ huấn luyện với từng mức lượng tử và chọn ra mức ứng với méo bé nhất. Lưu ý là ta chỉ lượng tử hoá những vectơ huấn luyện đã cho chứ không phải mọi vectơ có thể có của  $f$ . Từ những vectơ trong mỗi lớp xác định ra một mức lượng tử mới.

Để thuận tiện trong kí hiệu giả thiết  $f_i \ 1 \leq i \leq M_1$  là  $M_1$  vectơ huấn luyện lượng tử hoá về mức lượng tử  $r_1$ . Ước lượng mới của  $r_1$  nhận được bằng cách tối thiểu hoá  $\sum_{i=1}^{M_1} d(f_i, r_1) / M_1$ . Nếu như  $d(f, \hat{f})$  được dùng là sai số bình phương  $(\hat{f} - f)^T (\hat{f} - f)$  thì ước lượng mới của  $r_1$  vừa đúng là trung bình của  $M_1$  vectơ  $f_i ; 1 \leq i \leq M_1$ . Các ước lượng mới của mọi mức lượng tử  $r_i$  với  $2 \leq i \leq L$  cũng nhận được bằng cách tương tự. Đó là một biến tướng của (4.22), được thực hiện để phù hợp với D trong (4.23). Như vậy là hoàn thành một bước lặp trong chu trình, và đến khi méo trung bình D ở bước sau không khác bước trước mấy thì có thể ngừng chu trình lặp. Hình 4.14 biểu diễn algorit K\_means.

Vì có sự xếp đám cho nên algorit này còn gọi là algorit xếp đám, thường gặp trong các tài liệu về nhận dạng.

Người ta đã chứng minh là algorit K\_means hội tụ về một cực tiểu tại chỗ của D trong (4.23). Để xác định  $r_i$  với D gần cực tiểu tuyệt đối có thể lặp lại algorit này với những ước lượng ban đầu khác nhau của  $r_i$  và chọn lấy tập cho D bé nhất.

Phân tích toán tốn kém nhất trong algorit K\_means là việc lượng tử hoá các vectơ huấn luyện trong mỗi chu trình lặp. Với mỗi vectơ trong M vectơ huấn luyện độ méo phải được ước lượng L lần (cứ 1 mức lượng tử 1 lần).

Như vậy trong mỗi chu trình lặp phải tính ML lần độ méo. Nếu giả thiết có N vô hướng trong vectơ, mỗi vô hướng dùng R bit và mỗi mức lượng tử được gán một từ mã có chiều dài như nhau thì quan hệ giữa L với N và R là :

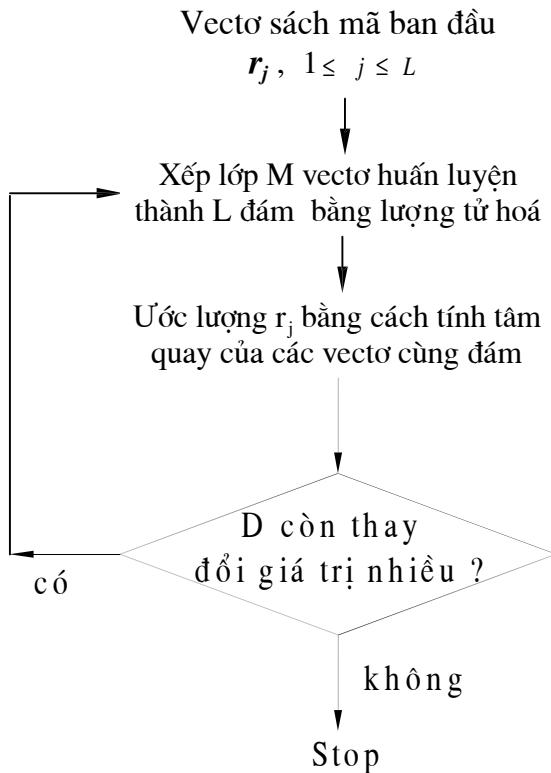
$$L = 2^B = 2^{NR} \quad (4.24)$$

B là tổng số bit dùng cho mỗi vectơ. Nếu giả thiết độ méo được dùng là sai số bình phương  $e_\Omega^T e_\Omega$  thì mỗi lần tính độ méo cần đến N phép tính số học (N phép nhân và N phép cộng). Số lượng phép tính số học cần cho bước lượng tử hoá vectơ huấn luyện trong mỗi chu trình là :

$$\text{Số lượng phép tính số học} = NML = NM \cdot 2^{NR} \quad (4.25)$$

Từ (4.25) thấy chi phí tính toán tăng theo hàm mũ N (số vô hướng trong mỗi vectơ) và R (số bit trong mỗi vô hướng).

Khi  $N = 10$  và  $R = 2$  và  $ML = 10L = 10 \cdot 2^{NR} = 10 \cdot 2^{20}$ , số lượng phép toán số học theo (25) là 100 nghìn tỉ cho mỗi bước lặp.



**Hình 4.14.** Thiết kế sách mã bằng algorit K\_means cho lượng tử hoá vectơ .

Ngoài chi phí tính toán còn chi phí lưu trữ. Giả sử mỗi vô hướng cần 1 đơn vị bộ nhớ, việc lưu trữ các vectơ huấn luyện cần đến MN đơn vị nhớ và việc lưu trữ các mức lượng tử cần đến LN đơn vị nhớ. Như vậy :

$$\text{Tổng số đơn vị bộ nhớ cần sử dụng} = (M+L)N = (M+2^{NR})N \quad (4.26)$$

Vì  $M >> N$  cho nên yêu cầu về bộ nhớ chủ yếu để lưu trữ các vectơ huấn luyện khi  $N = 10$ ,  $R = 2$ ,  $M = 10L = 10 \cdot 2^{20}$ , tính theo (4.26) số lượng đơn vị nhớ cần dùng phải cỡ 100 triệu .

Vì con số tính theo (4.26) tăng theo hàm mũ N và R do đó cả yêu cầu về tính toán và lưu trữ buộc ta chỉ sử dụng lượng tử hoá vectơ khi vectơ có số vô hướng ít, và số vô hướng có số bit ít. Trên đây thảo luận về yêu cầu về tính toán và lưu trữ khi thiết kế sách

mã. Khi thiết kế xong sách mã phải lưu trữ nó ở cả đầu máy phát và đầu máy thu. Vì sau khi thiết kế xong sách mã thì không cần đến vectơ huấn luyện nữa cho nên chỉ cần lưu trữ các mức lượng tử. Tuy thế số mức lượng tử phải lưu trữ cũng còn lớn. Trong trường hợp ở đây :

$$\text{Số đơn vị bộ nhớ cần cho sách mã} = NL = N2^{NR}. \quad (4.27)$$

Khi  $N = 10$ ,  $R = 2$ , con số tính theo (4.27) cỡ 10 triệu.

Để lượng tử hoá mỗi vectơ  $f$  phải tính độ méo  $d(f, r_i)$  cho từng bước lượng tử của  $L$  mức ở máy phát.

Vì thế cho mỗi vectơ :

$$\text{Số phép tính số học} = NL = N \cdot 2^{NR} \quad (4.28)$$

Khi  $N = 10$ ,  $R = 2$ , con số tính theo (4.28) cũng cỡ 10 triệu. Theo (4.27) và (4.28) cả số đơn vị trong bộ nhớ sách mã lẫn số thuật toán số học để lượng tử hoá 1 vectơ  $f$  đều tăng theo hàm mũ  $N$  (số vô hướng trong mỗi vectơ) và  $R$  (số bit ở mỗi vô hướng). Số thuật toán số học trên chỉ cần ở đầu phát. Ở đầu thu chỉ cần thuật toán tra bảng đơn giản.

#### **1.4. Sách mã cây và tìm kiếm nhị phân.**

Khối lượng tính toán chủ yếu khi thiết kế sách mã bằng algorit K\_means nằm ở khâu lượng tử hoá các vectơ huấn luyện. Cũng cần phải lượng tử hoá vectơ khi sách mã được dùng ở đầu phát. Nếu sách mã được thiết kế bằng thuật toán K\_means thì việc lượng tử hoá vectơ lúc thiết kế và việc truyền số liệu yêu cầu phải ước lượng độ méo giữa vectơ với từng mức trong  $L$  mức lượng tử. Quá trình này gọi là tìm kiếm đầy đủ và dẫn đến số phép tính tăng theo hàm mũ  $N$  và  $R$  (số vô hướng trong vectơ và số bit trong mỗi vô hướng). Nhiều phương pháp đã được phát triển để loại bỏ sự phụ thuộc theo hàm mũ này. Chúng làm giảm số lượng phép tính bằng cách thay đổi sách mã, bằng sự hi sinh phần nào chất lượng và cả sự tăng dung lượng bộ nhớ. Một trong những phương pháp gọi là sách mã cây. Ý cơ bản trong sách mã cây là đem chia không gian  $N$  chiều của  $f$  ra thành hai miền và dùng algorit K\_means với  $K = 2$ , sau đó lại đem chia mỗi miền ra làm hai và lại dùng algorit K\_means, cứ thế tiếp tục. Đặc biệt là khi  $L$  có thể biểu thị thành luỹ thừa của 2 thì thuật tiên thiết kế sách mã có 2 mức lượng tử  $r_1$  và  $r_2$ ,

dùng algorit K\_means. Sau đó ta xếp lớp tất cả các vectơ huấn luyện thành hai lớp, một lớp ứng với  $r_1$  và lớp kia ứng với  $r_2$ . Mỗi lớp được xử lý một cách độc lập và sách mã với hai mức lượng tử được thiết kế cho từng đám. Quá trình này cứ thế lặp đi lặp lại cho đến khi chúng ta có tất cả L mức lượng tử ở giai đoạn chót (tầng chót). Điều này được biểu diễn ở hình 4.15 cho trường hợp L = 8. Theo quy trình này thiết kế ra sách mã cây. Thoạt tiên ta hãy xét về yêu cầu tính toán và lưu trữ khi thiết kế sách mã. Ta vẫn giả thiết rằng mỗi lần tính độ đo độ méo phải mất N phép tính số học. Vì cả thảy có  $\log_2 N$  giai đoạn và độ méo chỉ xác định hai lần cho mỗi cái trong M vectơ huấn luyện ở mỗi giai đoạn mỗi chu trình lặp của algorit K\_means.

$$\text{Tổng số phép tính số học/chu trình lặp} = 2NM \log_2 L \quad (4.29)$$

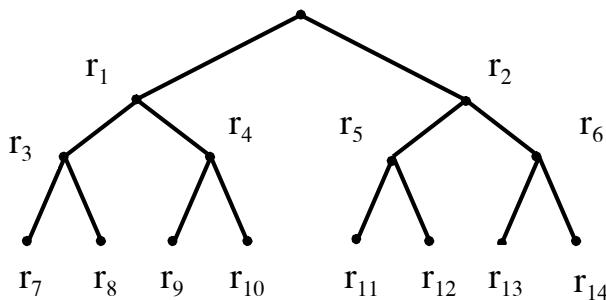
Đem con số này so sánh với số lượng tương ứng khi tính theo công thức (4.25) trong trường hợp tìm kiếm đầy đủ, thấy số phép tính giảm đi  $L/(2.\log_2 L) = 2^{NR}/(2NR)$  lần. Khi  $N = 10$ ,  $R = 2$  thì giảm được 26000 lần. Như cầu lưu trữ khi thiết kế sách mã cây đòi hỏi lớn hơn trường hợp algorit K\_means một ít, bởi vì trong cả hai trường hợp đều phải lưu trữ toàn bộ vectơ huấn luyện.

Bây giờ ta xét về số lượng phép tính khi lượng tử hoá 1 vectơ  $f$  bằng phương pháp sách mã cây. Ở giai đoạn thứ nhất ta tính độ méo giữa  $f$  và hai mức lượng tử  $r_1$  và  $r_2$  trên hình 4.15. Giả sử  $d(f, r_2) < d(f, r_1)$  thì ta chọn ra. Ở Giai đoạn hai ta tính độ méo giữa  $f$  và hai mức lượng tử  $r_5, r_6$  trên hình 4.15 và chọn ra mức lượng tử nào cho độ méo nhỏ hơn. Giả sử chọn  $r_5$  ở giai đoạn 3, ta lại so sánh  $f$  với  $r_{11}$  và  $r_{12}$ . Quá trình này cứ thế tiếp diễn cho đến giai đoạn chót. Mức lượng tử được chọn ở giai đoạn chót chính là mức lượng tử của  $f$ . Ví dụ trên nếu  $L = 8$  và  $r_{12}$  được chọn ở giai đoạn ba thì  $r_{12}$  là mức lượng tử của  $f$ . Trong quy trình này chúng ta cứ lần đi theo cây và tiến hành tìm kiếm giữa hai mức lượng tử ở điểm nút của cây. Sự tìm kiếm tiến hành giữa hai một lúc cho nên ta gọi là tìm kiếm nhị nguyên. Vì tất cả có  $\log_2 L$  giai đoạn và ở mỗi giai đoạn tính độ méo hai lần cho nên số phép tính số học cần thực hiện khi lượng tử hoá  $f$  theo phương pháp sách mã cây là :

$$\text{Số phép tính số học} = 2N \log_2 L = 2N^2 R \quad (4.30)$$

Theo công thức (4.30) thì chi phí tính toán không tăng theo hàm mũ N và R. Nếu so kết quả tính theo (4.30) với kết quả tính theo (4.28) trong trường hợp tìm kiếm đầy đủ thì chi phí cũng giảm  $2^{NR}/(2NR)$  lần. Khi  $N = 10$  và  $R = 2$  thì giảm được 26000 lần. Việc giảm bớt số lần tính toán phải trả giá. Sách mã dùng ở máy phát phải lưu trữ cả các mức

lượng tử trung gian lân các mức cuối cùng, bởi vì cần phải sử dụng các mức trung gian khi tìm kiếm.



**Hình 4.15.** Ví dụ về sách mã cây.

Vì thế cách mã trong trường hợp này có kích thước gấp đôi sách mã thiết kế theo algorit K - means. Ngoài ra trong nhiều trường hợp ứng dụng xét về độ méo trung bình thì tính năng sách mã cũng kém hơn sách mã thiết kế bằng algorit K - means. Nhưng trong nhiều trường hợp ưu điểm về giảm số phép tính có thể bù lại một cách hào phóng, nhược điểm làm tăng gấp đôi bộ nhớ và làm giảm chất lượng phần nào.

Sự tìm kiếm nhị nguyên là trường hợp riêng của một loạt phương pháp gọi là lượng tử hoá vectơ bằng tìm kiếm cây. Chẳng hạn có thể chia mỗi nút ra nhiều hơn hai nhánh. Ngoài ra khi thiết kế sách mã chúng ta có thể kết thúc một nút đặc biệt sớm hơn nếu chỉ có ít vectơ huấn luyện gán cho nút đó.

Trong tiết này ta thảo luận về lượng tử hoá vectơ. Ưu điểm của lượng tử hoá vectơ so với lượng tử hoá vô hướng là cải thiện chất lượng. Mức độ cải thiện phụ thuộc vào nhiều yếu tố, chẳng hạn mức độ sự phụ thuộc thống kê giữa các vô hướng trong vectơ. Sự cải thiện chất lượng này phải trả giá bằng chi phí cao cho tính toán và lưu trữ. Có đáng trả giá hay không là tùy theo từng trường hợp cụ thể. Lượng tử hoá vectơ chỉ có ích trong những ứng dụng mà tỷ lệ bit thấp, và sự cải thiện tính năng rất quan trọng, còn việc tăng chi phí thì không lớn lắm nhờ tỷ lệ bit thấp. Lượng tử hoá vectơ cũng có ích trong trường hợp quảng bá, khi mà số máy thu lớn gấp bội số máy phát và cho phép dùng máy phát đắt tiền. Máy thu vẫn cần lưu trữ sách mã, nhưng nó chỉ phải làm thao tác đơn giản là tra bảng để phục hồi 1 vectơ lượng tử hoá.

## 2. GÁN TỪ MÃ.

### 2.1 Gán từ mã có chiều dài đều.

Tiết 1 đã thảo luận vấn đề lượng tử hoá 1 nguồn vô hướng hoặc vectơ, kết quả của lượng tử hoá là đem lại một mức lượng tử nào đó. Để truyền cho máy thu mức lượng tử đã được chọn trong số L mức, phải gán cho mỗi mức này 1 từ mã (một chuỗi 0 và 1). Khi máy thu nhận được từ mã nó nhận ra mức lượng tử tương ứng nhờ tra sách mã. Để cho máy thu có thể nhận ra 1 mức lượng tử duy nhất, phải gán cho mỗi mức 1 từ mã riêng.

Ngoài ra, vì lúc truyền trong một dãy có thể chứa nhiều mức lượng tử, phải thiết kế từ mã sao cho máy thu nhận ra được từng từ khi truyền cả dãy. Một mã có những đặc tính như vậy gọi là có tính giải mã duy nhất. Khi  $L=4$ , gán 00 cho  $r_1$ , 01 cho  $r_2$ , 10 cho  $r_3$  và 11 cho  $r_4$  thì nhận được một mã không có tính giải mã duy nhất, bởi vì khi nhận được 1000 thì có thể hiểu ra  $r_3r_1$  hoặc  $r_2r_1r_1$ .

Sẽ rất thuận tiện nếu nghĩ rằng kết quả lượng tử hoá 1 vô hướng hoặc 1 vectơ có thể coi như một thông báo có L khả năng  $a_i$  khác nhau, ( $1 \leq i \leq L$ ) mà mỗi khả năng ứng với một lượng tử. Phương pháp đơn giản nhất để chọn từ mã là dùng từ mã có độ dài không đổi. Trong phương pháp này mỗi khả năng của thông báo được mã hoá bằng một từ mã có chiều dài giống như các khả năng khác. Một ví dụ về chọn từ mã có độ dài không đổi với  $L=8$  được biểu diễn trên bảng 4.2. Chiều dài của mỗi từ mã trong ví dụ này là  $\log_2 L = \log_2 8 = 3$ bit.

Số bit cần thiết để mã hoá 1 thông báo gọi là tỷ lệ bit. Trong ví dụ của chúng ta ở đây tỷ lệ bit là 3bit/thông báo. Nếu ta mã hoá nhiều hơn 1 thông báo thì tỷ lệ bit bình quân là tổng số bit cần thiết chia cho số thông báo. Trong trường hợp từ mã có chiều dài không đổi thì tỷ lệ bit bình quân cũng bằng tỷ lệ bit.

### 2.2 Entropy và việc gán từ mã có chiều dài biến đổi.

Gán từ mã có chiều dài không đổi tuy đơn giản nhưng thường là không tối ưu về mặt tỷ lệ bit trung bình. Giả sử có 1 vài khả năng của thông báo có xác suất được truyền nhiều hơn các khả năng khác. Vậy thì cái nào hay được truyền đi ta gán cho nó từ mã ngắn, còn cái nào ít truyền đi thi gán từ mã dài và như vậy sẽ giảm được tỷ lệ bit bình quân.

**Bảng 4.2:** Ví dụ về một bộ từ mã có chiều dài không đổi cho trường hợp bản thông báo có 8 khả năng .

Thông báo	Từ mã		
a <sub>1</sub>	0	0	0
a <sub>2</sub>	0	0	1
a <sub>3</sub>	0	1	0
a <sub>4</sub>	0	1	1
a <sub>5</sub>	1	0	0
a <sub>6</sub>	1	0	1
a <sub>7</sub>	1	1	0
a <sub>8</sub>	1	1	1

Khi từ mã được thiết kế theo đặc trưng thống kê của các khả năng thì phương pháp thiết kế gọi là mã hoá thống kê. Để thảo luận vấn đề thiết kế từ mã sao cho tỷ lệ bit bình quân thấp nhất, phải dùng khái niệm entropy định nghĩa như sau:

$$H = - \sum_{i=1}^L P_i \log_2 P_i \quad (4.31)$$

P<sub>i</sub> là xác suất để bản tin là a<sub>i</sub>. Vì  $\sum_{i=1}^L P_i = 1$  cho nên có thể chứng minh là:

$$0 \leq H \leq \log_2 L \quad (4.32)$$

Entropy H có thể coi là lượng tin tức bình quân chứa trong bản tin. Giả sử L=2. Nếu P<sub>1</sub>=1 và P<sub>2</sub> = 0 thì entropy H bằng không, và là giá trị cực tiểu ở trường hợp L=2. Trong trường hợp này xác suất để thông báo là a<sub>1</sub> là 100%, nghĩa là bản tin không chứa tin tức gì mới. Ở trạng thái cực đoan khác P<sub>1</sub> = P<sub>2</sub>=1/2, khi đó entropy H bằng 1 và là giá trị tối đa ở trường hợp L= 2, khi đó hai khả năng a<sub>1</sub> và a<sub>2</sub> của thông báo có xác suất xuất hiện như nhau, và nếu nhận được thông báo thì rõ ràng là có một tin tức mới.

Theo lý thuyết thông tin, entropy H trong(4.31) là tỷ lệ bit bình quân nhỏ nhất có thể đạt về mặt lý thuyết khi mã hoá một thông báo. Kết quả này tuy không chỉ ra được phương pháp thiết kế từ mã, nhưng rất có ích. Giả sử ta thiết kế ra những từ mã có tỷ lệ bit bình quân vừa bằng entropy. Ta biết những từ mã này là tối ưu rồi, và không cần nghiên cứu thêm.

Giả sử L có thể biểu thị thành luỹ thừa của 2 và các khả năng  $a_i$  của thông báo bằng nhau, tức là  $P_i = 1/L$  với  $1 \leq i \leq L$ .

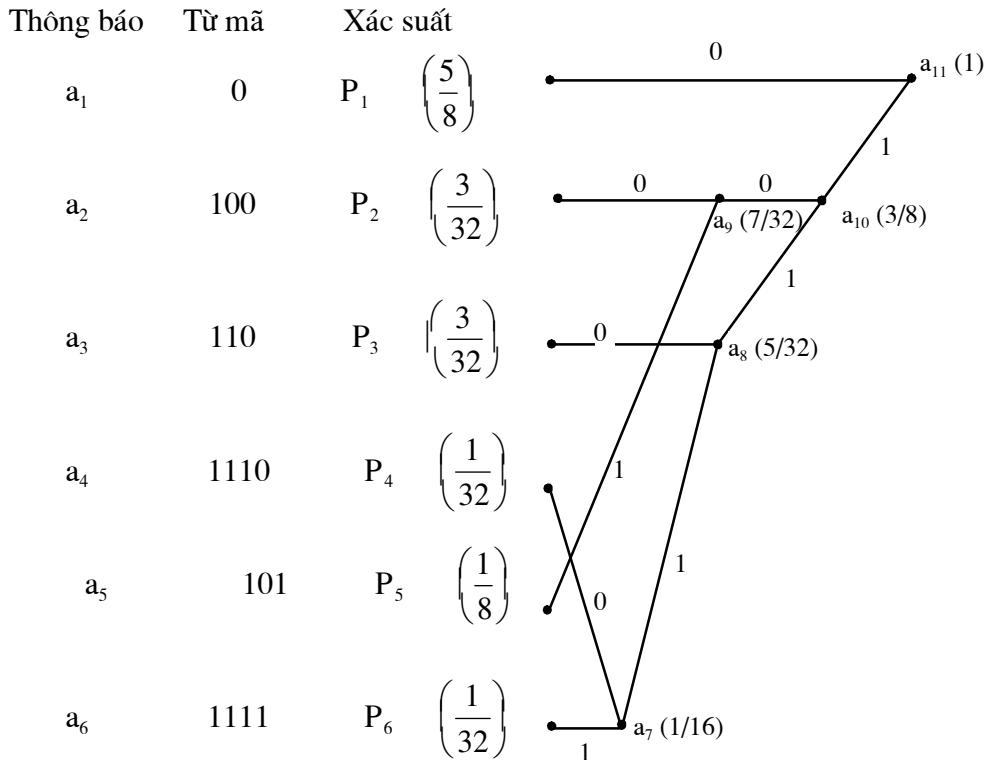
Theo (4.32) thì entropy H trong trường hợp này là  $\log_2 L$ . Vì từ mã có chiều dài không đổi đưa đến tỷ lệ bit bình quân bằng  $\log_2 L$  cho nên trong trường hợp này từ mã có chiều dài đều là lời giải tối ưu. Entropy cũng là một tiêu chuẩn để xét chất lượng phương pháp thiết kế từ mã. Nếu thiết kế ra đạt tỷ lệ bit bình quân gần sát giá trị entropy thì phương pháp thiết kế đó là có hiệu quả.

Nếu ta mã hoá từng thông báo riêng rẽ thì nói chung không thể thiết kế ra từ mã có tỷ lệ bit bình quân bằng entropy. Chẳng hạn, nếu  $L = 2$ ,  $P_1 = 1/8$ ,  $P_2 = 7/8$  entropy H của bản tin này là 0,544, nhưng không thể nào thiết kế ra từ mã đem lại tỷ lệ bit bé hơn một bit/thông báo.

Có một phương pháp thiết kế từ mã tối ưu khá đơn giản khi sử dụng, và cũng có tính giải mã duy nhất, là phương pháp mã hoá Huffman. Hình 4.16 cho một ví dụ về mã hoá Huffman.

Trong ví dụ này  $L = 6$  xác suất từng khả năng của thông báo được ghi ở các điểm nút trên hình vẽ. Trong bước 1 của mã hoá Huffman ta chọn ra hai khả năng của thông báo có xác suất thấp nhất. ở đây đó là  $a_4$  và  $a_6$ . Ta kết hợp xác suất của hai thông báo này và lập ra một nút mới có xác suất kết hợp. Ta gán 0 cho một nhánh và 1 cho nhánh còn lại. Tráo 0 và 1 chỉ ảnh hưởng đến từ mã được gán mã không ảnh hưởng tỷ lệ bit bình quân.

Bây giờ ta xem hai khả năng  $a_4$  và  $a_6$  như một khả năng  $a_7$  có xác suất kết hợp là  $1/16$ . Rồi ta chọn trong thông báo  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_5$  và  $a_7$  hai khả năng có xác suất thấp nhất. ở đây 2 khả năng đó là  $a_3$  và  $a_7$ . Ta lại đem kết hợp chúng thành một thông báo, gán 0 cho một nhánh và 1 cho nhánh còn lại. Cứ như thế tiếp tục cho đến khi chỉ còn một khả năng của thông báo với bức xác suất 1. Để xác định từ mã gán cho mỗi khả năng của thông báo ta bắt đầu bằng điểm nút cuối cùng có xác suất 1, lần theo các nhánh dẫn tới khả năng cần xét của thông báo và kết hợp những bit 0 và 1 nhặt được trên các nhánh phải đi qua. Chẳng hạn với khả năng  $a_4$  của thông báo thì từ mã là 1110. Các từ mã nhận được bằng phương pháp này được ghi lại trên hình 4.16. Các khả năng có xác suất cao đã được gán từ mã ngắn, còn các khả năng có xác suất thấp đã được gán từ mã dài. Để so sánh tính năng của phương pháp mã hoá Huffman với entropy H và với từ mã đều trong ví dụ trên ta tính tỷ lệ bit bình quân trong trường hợp từ mã đều và trường hợp từ mã Huffman và tính entropy. Kết quả như sau:



**Hình 4.16 :** Minh họa việc tạo từ mã theo mã Huffman. Bản tin với xác xuất cao hơn được gán cho những từ mã ngắn hơn.

- Từ mã đều: 3bit/bản tin
- Mã hoá Huffman:  $\frac{5}{8}.1 + \frac{3}{32}.3 + \frac{3}{32}.3 + \frac{1}{32}.4 + \frac{1}{8}.3 + \frac{1}{32}.4 = \frac{29}{16}$  bit/bản tin  
 $= 1,813$  bit/bản tin

- Entropy :

$$-\left(\frac{5}{8}\log_2\frac{5}{8} + \frac{3}{32}\log_2\frac{3}{32} + \frac{3}{32}\log_2\frac{3}{32} + \frac{1}{32}\log_2\frac{1}{32} + \frac{1}{8}\log_2\frac{1}{8} + \frac{1}{32}\log_2\frac{1}{32}\right) \\ = 1,752 \text{bit/thông báo}$$

Trong ví dụ trên mã hoá Huffman cho nhịp bit bình quân gần sát với entropy hơn và có một nhịp bit bình quân thấp hơn trường hợp từ mã đều là hơn 1 bit/bản tin. Việc thay từ mã đều bằng từ mã có chiều dài thay đổi phải trả giá bằng tốn kém cao hơn.

Nhịp bit tức thời là 1 đại lượng biến thiên. Khi những khả năng của thông báo có xác suất thấp được mã hoá kế tiếp nhau thì nhịp bit tức thời cao hơn nhịp bit bình quân khá nhiều. Trong trường hợp ngược lại nhịp bit tức thời thấp hơn nhịp bit bình quân nhiều. Khi truyền những bản tin mà từ mã có chiều dài biến đổi qua 1 hệ có nhịp bit không đổi thì phải có một tầng đệm để chứa những thông báo mà nhịp bit bình quân cao. Thêm tầng đệm thì thêm phức tạp cho hệ mã hoá và cũng gây trễ.

Nếu ta được phép gộp bất kỳ bao nhiêu thông báo tuỳ ý để gán chung từ mã thì có thể thiết kế ra từ mã với nhịp bit bình quân kề sát tới entropy H bao nhiêu cũng được. Vì lý do đó phương pháp Huffman gọi là phương pháp mã hoá entropy.

Hãy xét trường hợp  $L=2$ ,  $P_1=1/8$ ,  $P_2=7/8$  khi đó entropy  $H = 0,544$  bit. Nếu gán 1 từ mã cho 1 thông báo thì nhịp bit bình quân là 1bit/thông báo. Giả sử ta chờ đến khi có 2 thông báo và gán cho chúng 1 từ mã. Theo phương pháp Huffman sẽ nhận được từ mã như trong bảng 4.3. Khi đó nhịp bit bình quân là 0,680 bit/thông báo. Nếu tăng số lượng thông báo được mã hoá kết hợp như thế thì nhịp bit bình quân theo mã Huffman sẽ tiệm cận entropy. Trong thực tế nói chung không thể chờ một số lượng thông báo lớn, bởi vì như vậy sẽ gây trễ và cần đến 1 sách mã lớn hơn. Cho nên nhịp bit bình quân đạt được bằng phương pháp Huffman nói chung cao hơn entropy.

**Bảng 4.3 :** Mã Huffman khi mã hoá kết hợp từng cụm 2 thông báo, 2 thông báo này độc lập với nhau và mỗi thông báo có 2 khả năng, với xác suất là  $1/8$  và  $7/8$ .

Cụm thông báo	Xác suất	Mã Huffman
$a_1a_1$	$\frac{1}{64}$	0 0 0
$a_1a_2$	$\frac{7}{64}$	0 0 1
$a_2a_1$	$\frac{7}{64}$	0 1
$a_2a_2$	$\frac{49}{64}$	1

### 2.3. Tối ưu hoá kết hợp của lượng tử hoá và gán từ mã.

Ở tiết 1 bàn về vấn đề lượng tử hoá. Ở tiết 2 bàn về vấn đề gán từ mã cho các mức lượng tử. Tuy 2 vấn đề được thảo luận riêng rẽ nhưng chúng lại có quan hệ chặt chẽ với nhau. Chẳng hạn ta lượng tử hoá 1 vô hướng  $f$  có hàm mật độ xác suất không đồng đều nhưng dùng bộ lượng tử hoá đều. Các mức lượng tử trong trường hợp này có xác suất không bằng nhau do đó dùng từ mã chiều dài không đều sẽ giảm được nhịp bit bình quân so với trường hợp dùng từ mã đều. Mặt khác nếu các mức lượng tử trong khâu lượng tử hoá được chọn sao cho xác suất xuất hiện giống nhau thì dùng từ mã có chiều dài không đều sẽ không ưu việt gì hơn từ mã đều. Như vậy những điều làm trong khâu lượng tử hoá sẽ ảnh hưởng đến những việc phải làm trong khâu gán từ mã. Vì 2 khâu gắn bó với nhau như vậy cho nên tối ưu hoá riêng rẽ từng khâu không mang lại hiệu quả tổng hợp tối ưu cho bài toán. Trong tiết 1 ta xét vấn đề tối thiểu hoá độ méo khi giữ nguyên số mức lượng tử  $L$  hoặc tối thiểu hoá số mức lượng tử khi giữ nguyên độ méo. Trong thực tế chúng ta thường qua tâm tối thiểu hoá số bit chứ không phải tối thiểu hoá số mức lượng tử. Nếu ta gán từ mã đều thì số bit sẽ quy định số mức lượng tử và cả 2 bài toán được coi là tương đương. Nhưng nếu gán từ mã chiều dài thay đổi thì số mức lượng tử ít không nhất thiết kéo theo số bit ít. Chẳng hạn 4 mức lượng tử có xác suất xuất hiện không bằng nhau có thể có entropy thấp hơn 2 mức lượng tử mà xác suất xuất hiện như nhau. Tối thiểu hoá số bit lượng tử ở một độ méo trung bình đã cho, rồi tối thiểu hoá nhịp bit bình quân bằng cách thiết kế những từ mã tối ưu đối với những mức lượng tử đã cho thông thường không đem lại nhịp bit bình quân thấp nhất ở một độ méo đã cho. Tối thiểu hoá nhịp bit bình quân ở một độ méo đã cho bằng cách tối ưu hoá kết hợp hai khâu lượng tử hoá và gán từ mã là một bài toán có tính phi tuyến cao và chỉ nhận được lời giải gần đúng trong một số trường hợp cụ thể.

Trong tiết 1 và 2 thảo luận về cách lượng tử hoá và gán từ mã. Điều ta quan tâm nhất là làm sao tối thiểu hoá nhịp bit đối với một độ méo đã cho. Tất nhiên trong thực tế phải xét đến nhiều yếu tố, chẳng hạn những yêu cầu về tính toán và lưu trữ và một độ trễ chấp nhận được. Ngoài ra ở , tiết 1 và 2 ta giả thiết rằng những số liệu thống kê như  $p_f(f_0)$  và ở độ méo  $d(f, \hat{f})$  đều đã biết. Trong thực tế những số liệu thống kê này phải ước lượng và độ méo cụ thể đối với từng trường hợp ứng dụng cụ thể thường không biết trước được.

Ước lượng sai số liệu thống kê hoặc cho độ méo sai đều ảnh hưởng kết quả. Vì thế, những kết quả lý thuyết của các phương pháp lượng tử hoá và gán từ mã chỉ nên coi là những căn cứ để hướng dẫn trong việc chọn phương pháp mã hoá cho từng ứng dụng cụ thể.

### 3. MÃ HOÁ DẠNG SÓNG.

Trong tiết 1 và 2 nói về lượng tử hoá và gán từ mã là công đoạn thứ 2 và thứ 3 trong 3 công đoạn mã hoá ảnh. Nay ta nói về công đoạn thứ nhất là đem ảnh biến đổi vào 1 miền thuận lợi nhất cho lượng tử hoá và gán từ mã. Công đoạn này quyết định đại lượng nào đem ra mã hoá. Các algorit mã hoá ảnh được phân thành 3 loại tùy theo đối tượng nào trong ảnh được đem mã hoá. Trong tiết này nói về bộ mã hoá dạng sóng, trong tiết 4 và 5 sẽ nói về bộ mã hoá biến đổi và bộ mã hoá mô hình ảnh.

Trong mã hoá dạng sóng ta đem mã hoá cường độ ảnh hoặc mã hoá sự biến thiên cường độ ảnh tức là hiệu số cường độ ảnh của 2 pixel kề nhau. Ưu điểm chủ yếu của mã hoá dạng sóng là tính đơn giản. Kỹ thuật mã hoá dạng sóng có xu thế phục hồi dạng sóng 1 cách trung thực mà không đi sâu khai thác những thông tin đặc thù cho 1 loại tín hiệu, do đó nó có thể dùng rộng rãi cho nhiều loại tín hiệu khác nhau, chẳng hạn tín hiệu ảnh và tiếng nói.

Ngoài ra bộ mã hoá dạng sóng có thể làm giảm tỷ lệ bít cùng cỡ như mã hoá phép biến đổi trong một số ứng dụng cụ thể, chẳng hạn trong truyền hình số, là trường hợp yêu cầu rất cao về chất lượng hình ảnh.

Trong những ứng dụng như hội nghị video và điều khiển tàu xe từ xa yêu cầu về giảm nhịp bit rất cao và có thể cho phép hi sinh một phần chất lượng thì bộ mã hoá biến đổi ưu việt hơn bộ mã hoá dạng sóng.

Về nguyên tắc có thể dùng bất kỳ phương pháp lượng tử hoá và gán từ mã nào đã được nói đến ở các tiết 1 và 2. Tuy vậy vì lý do đơn giản người ta vẫn thích dùng lượng tử hoá vô hướng và từ mã đều.

Trong những thảo luận tiếp theo coi như chỉ dùng lượng tử hoá vô hướng và từ mã đều trừ trường hợp đặc biệt sẽ có chú thích.

Trong tiết này ta dùng các ví dụ để minh họa về tính năng của từng algorit mã hoá ảnh. Trong những trường hợp có thể sẽ cho sai số quân phương chuẩn hoá NMSE và tỉ số tín hiệu trên tạp âm SNR.

$$NMSE\% = 100 \frac{V_{ar} [\hat{f}(n_1, n_2) - f(n_1, n_2)]}{V_{ar} [f(n_1, n_2)]}\% \quad (4.33a)$$

$$SNR(\text{tính bằng dB}) = 10 \log \frac{(NMSE\%)}{100} dB \quad (4.33b)$$

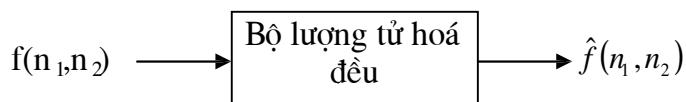
Trong đó  $f(n_1, n_2)$  là ảnh gốc,  $\hat{f}(n_1, n_2)$  là ảnh mã hoá.

### 3.1 Điều chế xung mã (PCM).

Phương pháp mã hoá dạng sóng đơn giản nhất là điều xung mã, trong đó cường độ ảnh  $f(n_1, n_2)$  được lượng tử hoá đều. Sơ đồ cơ bản của hệ PCM trên hình 4.17. Cường độ ảnh  $f(n_1, n_2)$  sau khi lượng tử hoá kí hiệu là  $\hat{f}(n_1, n_2)$ . Hệ PCM không những có thể dùng để mã hoá cường độ ảnh mà còn có thể mã hoá các hệ số biến đổi và các thông số của mô hình ảnh. Tuy vậy nó được dùng từ đầu để mã hoá dạng sóng và đến nay vẫn còn được dùng rộng rãi. Cho nên hệ PCM nếu không có gì nói thêm thì cứ coi là 1 bộ mã hoá dạng sóng. Điều này cũng áp dụng cho các bộ mã hoá dạng sóng khác như hệ điều chế delta (DM) hoặc điều chế xung mã vi sai (DPCM).

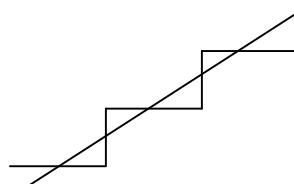
Hệ PCM cơ bản ở hình 4.17 dùng để biến 1 ảnh analog ra 1 ảnh digital  $f(n_1, n_2)$ . Khả năng phân giải không gian của ảnh digital  $f(n_1, n_2)$  trước hết là do kích thước của nó quyết định, tức là do số pixel quyết định. Kích thước của  $f(n_1, n_2)$  chọn theo yêu cầu về độ phân biệt mà mỗi trường hợp ứng dụng cụ thể đặt ra. Một ảnh digital có 1024 x 1024 pixel thì độ phân biệt tương đương với phim 35 mm. Ảnh digital có 512 x 512 pixel thì độ phân biệt tương đương với truyền hình. Ảnh digital có 256 x 256 pixel và 128 x 128 pixel dùng trong điện thoại video. Kích thước của ảnh giảm thì độ phân biệt giảm và những chi tiết ảnh sẽ mất đi. Tỷ lệ bit thường dùng cho 1 ảnh gốc digital là 8 bit/pixel. Ngoài những trường hợp yêu cầu biểu diễn ảnh gốc rất chính xác như xử lý ảnh y tế, còn nói chung hệ PCM với 8 bit/pixel đảm bảo đủ chất lượng và độ dẽ hiểu c ho nhiều trường hợp và ứng dụng. Trong sự thảo luận của chúng ta tỷ lệ bit được biểu diễn bằng bit/pixel. Cần lưu ý rằng độ đo bit/pixel có khi gây nhầm lẫn, chẳng hạn nếu ta nhận 1 ảnh digital

bằng cách lấy mẫu ảnh analog với tỷ lệ bit cao hơn nhiều so với khả năng cảm nhận của mắt người thì có thể giảm số bit/pixel mà không tạo ra sự giảm độ phân biệt nhận thấy được bằng cách đơn giản là dùng tần số lấy mẫu thấp hơn có 1 độ đo có ý nghĩa hơn, là số bit/khung hình khi mã hoá loại ảnh chỉ có khung hình, hay số bit/giây, khi mã hoá 1 dãy ảnh. Tuy vậy vì thuận tiện vẫn đo tỷ lệ bit bằng bit/pixel. Ngoài ra chúng ta sẽ nói rõ kích thước khung hình của ảnh, khi mã hoá 1 ảnh chỉ có 1 khung hình, còn trong trường hợp mã hoá dãy ảnh thì ta cho cả kích thước khung hình lẫn số khung hình/giây.



**Hình 4.17:** Hệ thống điều xung mã.

Nếu tín hiệu vào thăng giáng và nếu bước lượng tử hoá đủ nhỏ thì trong hệ PCM tạp âm lượng tử coi như tạp âm cộng. Trong những trường hợp điển hình tạp âm lượng tử có thể nhìn thấy như tạp âm ngẫu nhiên ở tỷ lệ bit 5 ~ 6 bit/pixel. Nếu ta làm giảm tỷ lệ bit xuống dưới 3 ~ 4 bit/pixel thì tạp âm lượng tử phụ thuộc tín hiệu sẽ hiện thành những vòng viền trên ảnh do các bước nhảy độ chói ở những vùng mà cường độ ảnh gốc biến thiên rất chậm. Điều đó có thể thấy ở hình 4.18.



**Hình 4.18 :** Minh họa bước nhảy độ chói gây ra các vòng viền trên ảnh khi dùng PCM mã hoá ảnh.

### Điều xung mã với lượng tử hoá không đều.

Một cách đơn giản để cải thiện tính năng hệ PCM cơ bản là dù ng lượng tử hoá không đều. Cường độ ảnh không phân bố đều trong giải động . Trong trường hợp đó dùng lượng tử hoá không đều có thể cải tiến chất lượng. Một trong những phương pháp để thực hiện lượng tử hoá phi tuyến là áp dụng thuật toán phi tuyến cho  $f(n_1, n_2)$  sau đó tiến hành lượng tử hoá đều xong rồi lại áp dụng thuật toán phi tuyến ngược. Thuật toán phi tuyến được chọn sao cho đầu ra gần đúng là có xác suất phân bố đều trên cả dải động.

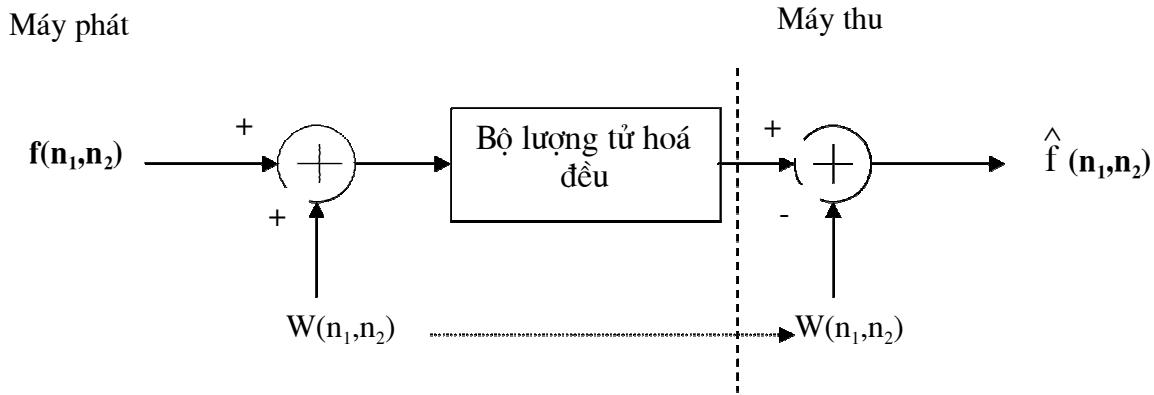
Tuy lượng tử hoá không đều có thể cải thiện tính năng hệ PCM cơ bản , nhưng trong 1 số trường hợp ứng dụng sự cải thiện không lớn. Như đã thảo luận ở tiết 3.1 (xem hình 4.6) lượng tử hoá không đều có thể làm giảm sai số quân phương non 3 dB trong trường hợp ảnh có tổ chức đồ (histogram) dạng Gauss với tỷ lệ bit tới 7 bit/pixel. Trong trường hợp mà histogram lệch xa với histogram đều thì dùng lượng tử hoá không đều có thể cải tiến tính năng của hệ PCM cơ bản 1 cách đáng kể. Chẳng hạn trong trường hợp cường độ của hình tập trung vào 1 vài dải rất hẹp mà dùng lượng tử hoá đều sẽ bỏ phí nhiều mức lượng tử. Nếu dùng lượng tử hoá không đều có thể đếm mức lượng tử đặt vào những vùng tập trung cường độ ảnh.

### Kỹ thuật tap âm giả của Robert.

Một cách khác để cải thiện tính năng hệ PCM là gỡ bỏ sự phụ thuộc tín hiệu của tạp âm lượng tử vẫn thường hiện ra dưới dạng những đường viền khi tỷ lệ bit thấp. Trong phương pháp Robert tạp âm lượng tử phụ thuộc tín hiệu được biến đổi thành tạp âm ngẫu nhiên không phụ thuộc tín hiệu. Phương pháp này được biểu diễn trên hình 4.19. Ở đây 1 tạp âm ngẫu nhiên đã biết  $\omega(n_1, n_2)$  được cộng vào ảnh gốc  $f(n_1, n_2)$  trước khi đếm lượng tử hoá ở đầu phát, rồi sau này đến máy thu lại đếm loại  $\omega(n_1, n_2)$  ra. Bởi vì cả đầu phát và đầu thu đều biết  $\omega(n_1, n_2)$  trước khi truyền ảnh, cho nên cũng chẳng cần truyền nó đi để dùng ở đầu thu. Một chuỗi tạp âm trắng có hàm mật độ xác suất đều có HMDXS đều  $P_\omega(\omega_0)$  :

$$P_0(\omega_0) = \begin{cases} \frac{1}{\Delta} & -\frac{\Delta}{2} \leq \omega_0 \leq \frac{\Delta}{2} \\ 0 & \text{ở nơi khác} \end{cases} \quad (4.34)$$

Trong đó  $\Delta$  là bước lượng tử, có thể đếm dùng như tần số ngẫu nhiên  $\omega(n_1, n_2)$ . Với  $P_0(\omega_0)$  trong (4.34) có thể chứng minh rằng ảnh phục hồi  $\hat{f}(n_1, n_2)$  có thể mô phỏng gần đúng ảnh gốc  $f(n_1, n_2)$  tuy có bị xuống cấp chút ít vì tần số ngẫu nhiên không phụ thuộc tín hiệu, nó là tần số trắng và có hàm mật độ xác suất như trong (4.34).

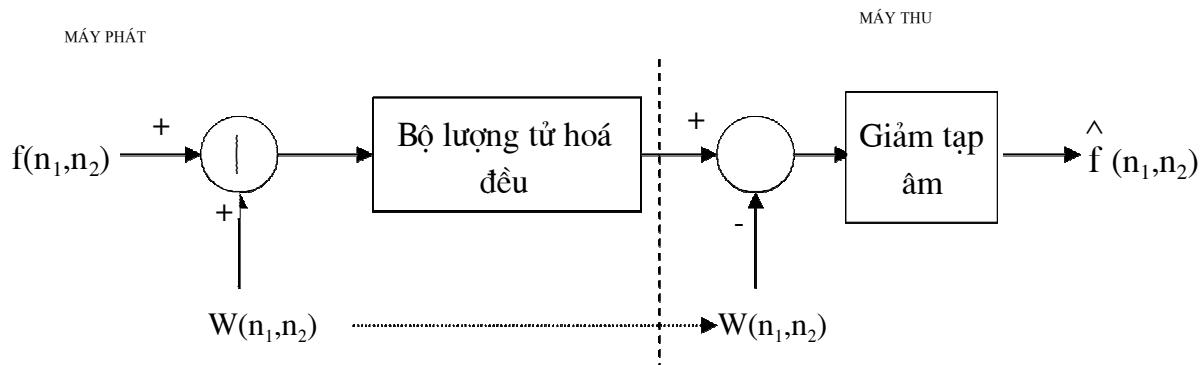


**Hình 4.19 :** Giải tương quan tần số lượng tử bằng kỹ thuật giả tần số của Robert.

Chỉ đơn giản loại bỏ sự phụ thuộc tín hiệu của tần số lượng tử đã cải thiện đáng kể chất lượng hệ PCM.

Ngoài ra nó còn cho phép sử dụng bất kỳ hệ làm giảm tần số nào cũng có thể làm giảm tần số riêng lẻ với tín hiệu.

Hình 4.20 vẽ 1 hệ có gắn bộ giảm tần số lượng tử đi kèm với kỹ thuật Robert. Hệ giảm tần số này chỉ gắn ở máy thu. Hệ này rất có ích khi dùng cho tàu vũ trụ và xe tàu điều khiển từ xa, khi đó cần máy phát đơn giản còn máy thu có thể rất phức tạp.



**Hình 4.20 :** Giảm tần số lượng tử khi dùng PCM mã hóa ánh.

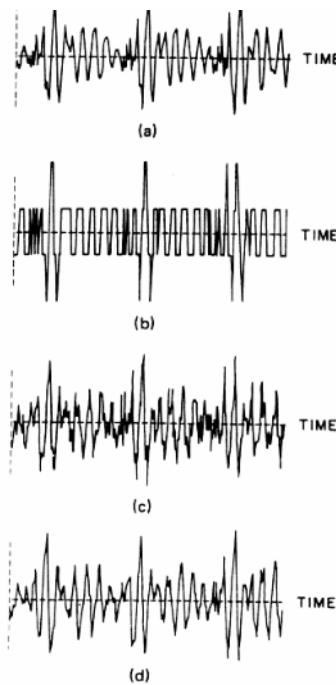
Để làm sáng tỏ hơn khái niệm loại trừ sự phụ thuộc tín hiệu của tạp âm lượng tử ta hãy xét 1 tín hiệu 1\_D (không gian 1 chiều).

Hình 4.21a vẽ 1 đoạn tín hiệu tiếng nói không có tạp âm.

Hình 4.21b vẽ dạng sóng tiến nói khi hệ PCM dùng lượng tử hoá đều với tỷ lệ bit 2 bit/mẫu. Ảnh hưởng của tạp âm phụ thuộc tín hiệu thể hiện rõ trong dạng sóng hình cầu thang.

Hình 4.21c vẽ kết quả khi cộng rồi lại khử tạp âm Rober t trong hệ PCM dùng bộ lượng tử hoá đến 2 bit/ mẫu. Kết quả có thể mô hình hoá bằng tín hiệu gốc bị giảm cấp vì tạp âm cộng ngẫu nhiên không phụ thuộc tín hiệu.

Hình 4.21d vẽ kết quả nhận được khi đã sử dụng bộ giảm tạp âm tác động vào hình sóng ở 4.21c.



**Hình 4.21 :** Ví dụ về giảm tạp âm lượng tử khi mã hoá PCM tiếng nói :

- Mẫu tín hiệu tiếng nói không có tạp âm.
- Tiếng nói mã hoá PCM với tỷ lệ 2 bit/mẫu.
- Tiếng nói mã hoá PCM tỷ lệ 2 bit/mẫu có áp dụng kỹ thuật Robert.
- Tiếng nói mã hoá PCM tỷ lệ 2 bit/mẫu có giảm tạp âm lượng tử.

Hình 4.22 cho ví dụ về 1 ảnh.

Hình 4.22a là ảnh gốc kích thước 512 x 512 pixel với tỷ lệ 8 bit /pixel.

Hình 4.22b cho kết quả của 1 hệ PCM có bộ lượng tử hoá đều với tỷ lệ 2 bit/pixel.

Hình 4.22c là kết quả của kỹ thuật Robert .

Tuy hai hình 4.22b và 4.22c có cùng độ méo (cùng MMSE) nhưng ảnh ở hình 4.22c trông có vẻ thật hơn. Ảnh ở hình 4.22d là ảnh 4.22c sau khi đã cho qua bộ lọc thích nghi Wiener .

Quá trình loại trừ sự phụ thuộc tín hiệu của tạp âm lượng tử và sau đó làm giảm tạp âm bằng 1 algorit phục hồi ảnh có thể áp dụng cho bất kỳ hệ nào có bộ lượng tử hoá đều tham gia.

Chẳng hạn có 1 hệ PCM dùng bộ lượng tử hoá không đều như trên hình 4.23a. Tạp âm giả của Robert được cộng vào trước bộ lượng tử hoá đều và sau đó lại loại bỏ ra (xem hình 4.23d). Tín hiệu  $\hat{g}(n_1, n_2)$  coi như  $g(n_1, n_2)$  bị xuống cấp vì tạp âm cộng ngẫu nhiên, độc lập với  $g(n_1, n_2)$ . Nếu muốn làm giảm tạp âm thì đem 1 bộ giảm tạp âm tác động vào  $\hat{g}(n_1, n_2)$  như trên hình 4.23b.

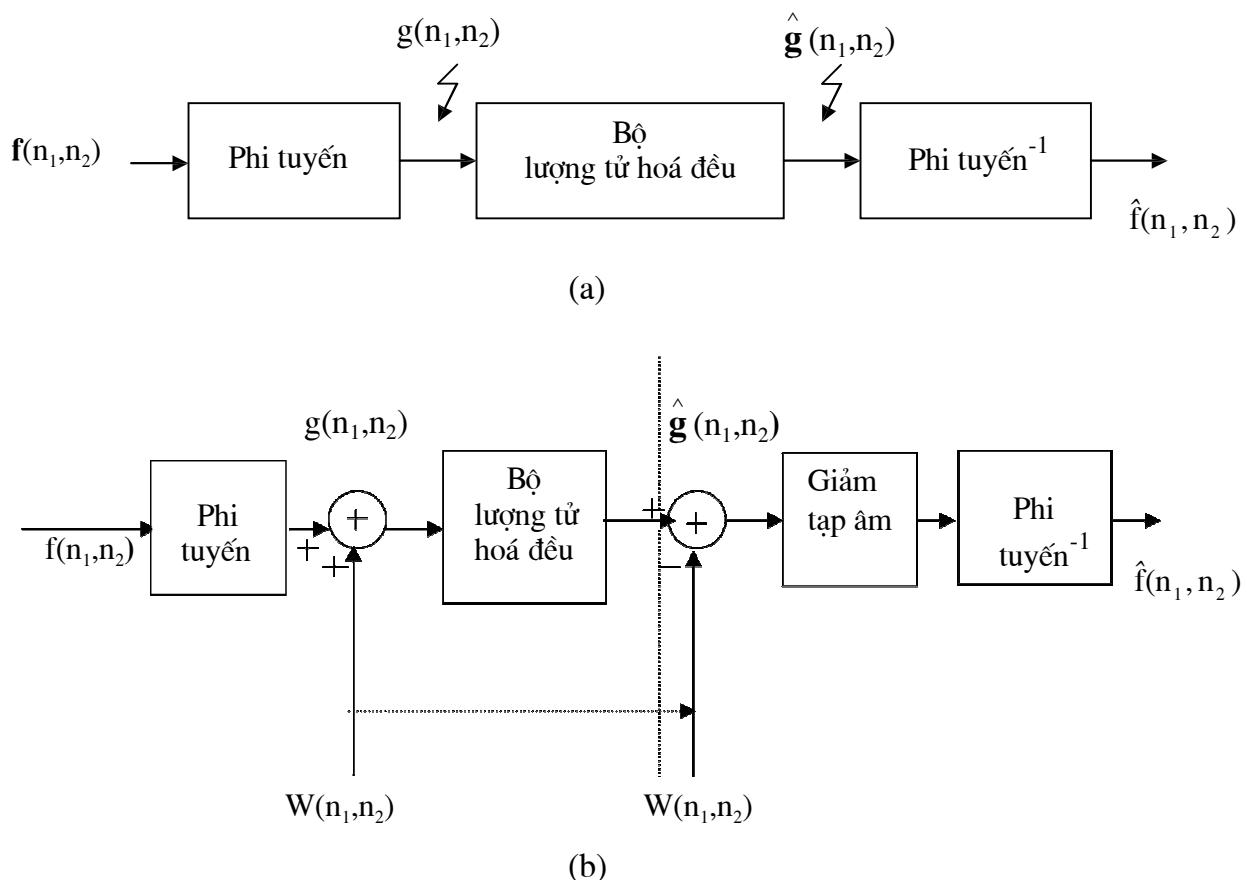
### 3.2 Điều chế delta (DM).

Trong hệ PCM cường độ ảnh được mã hoá bằng lượng tử hoá vô hướng và sự tương quan giữa cường độ các pixel không được khai thác. Có 1 cách để khai thác tương quan phần nào, mà vẫn dùng lượng tử hoá vô hướng, là điều chế DM. Trong hệ DM hiệu cường độ của hai pixel kề nhau được mã hoá bằng 1 bộ lượng tử hoá 1 bit (2 mức lượng tử). Mặc dù độ rộng dải của hiệu số tín hiệu bị tăng gấp đôi do kết quả lấy sai phân, phương sai của tín hiệu số bị giảm đáng kể do sự tương quan mạnh giữa cường độ 2 pixel kề nhau trong không gian.

Khi thảo luận về DM nên coi là các pixel trong ảnh đã được sắp xếp theo dãy sao cho  $f(n_1, n_2)$  có thể coi như tín hiệu trong không gian 1-D, tức là  $f(n)$ . Nếu  $f(n)$  nhận được bằng cách đọc 1 hàng của  $f(n_1, n_2)$  rồi đọc 1 hàng tiếp theo thì nó giữ được 1 phần tính tương quan không gian có trong  $f(n_1, n_2)$ .

**Hình 4.22 :** Ví dụ giảm tạp âm lượng tử khi mã hoá PCM.

- (a) Ảnh gốc 512 x 512 pixel
- (b) Mã hoá PCM với tỷ lệ 2 bit/pixel
- (c) Mã hoá PCM tỷ lệ 2 bit/pixel có dùng kỹ thuật Robert.
- (d) Mã hoá PCM với tỷ lệ 2 bit/pixel có giảm tạp âm lượng tử.

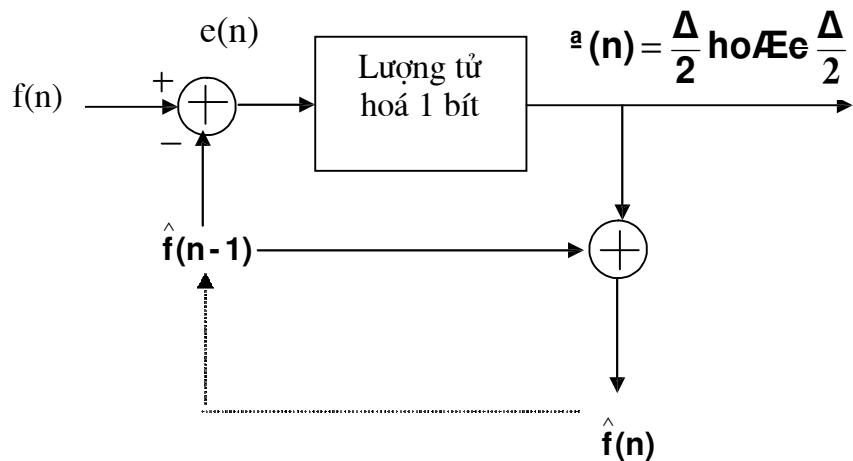


**Hình 4.23:** Giảm tạp âm lượng tử trong hệ PCM lượng tử hoá không đều:

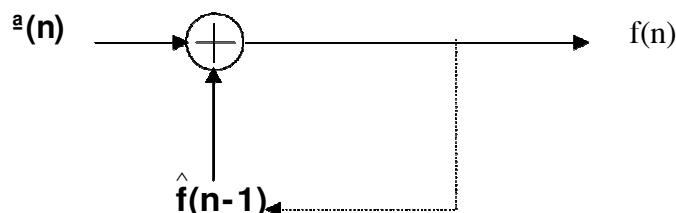
- a) Hệ PCM
- b) Hệ PCM có giảm tạp âm lượng tử .

Hình 4.24 vẽ 1 hệ DM. Trong hình này  $\hat{f}(n)$  đại biểu  $f(n)$  đã được hệ DM phục hồi. Để mã hoá  $f(n)$  đem  $f(n)$  trừ đi  $\hat{f}(n_1, n_2)$  vừa được phục hồi. Hiệu số  $e(n) = f(n) - \hat{f}(n_1, n_2)$  được lượng tử hoá về  $\Delta/2$  nếu  $e(n)$  dương và về  $-\Delta/2$  nếu  $e(n)$  âm, trong đó  $\Delta$  là 1 bước nhảy. Hiệu số  $e(n)$  được 1 bộ lượng tử hoá 1 bit mã hoá thành  $\hat{e}(n)$  và được máy phát truyền đi. Ở máy thu,  $\hat{e}(n)$  được cộng vào  $\hat{f}(n_1, n_2)$  để lấy ra  $\hat{f}(n)$ . Đầu máy phát cũng cần tín hiệu  $\hat{f}(n)$  để mã hoá  $f(n+1)$ . Đường vẽ chấm trên hình đại biểu mà sự trễ và cho biết  $\hat{f}(n)$  phải được tính sao cho có thể dùng nó để tính mẫu tiếp theo  $f(n+1)$ .

Máy phát



Máy thu



Hình 3.24: Hệ điều chế DM

Các phương trình về hệ DM trên hình 4.24 là:

$$e(n) = f(n) - \hat{f}(n-1) \quad (4.35a)$$

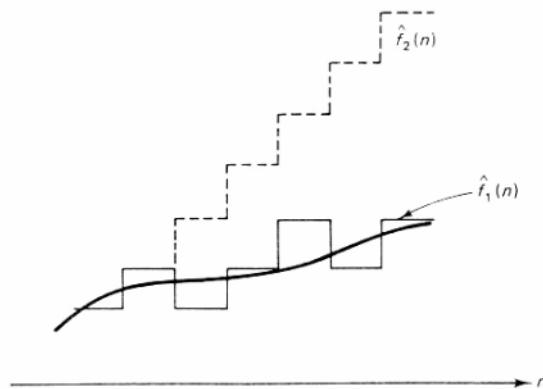
$$\hat{e}(n) = \begin{cases} \frac{\Delta}{2} & e(n) > 0 \\ -\frac{\Delta}{2} & e(n) \leq 0 \end{cases} \quad (4.35b)$$

$$\hat{f}(n) = \hat{f}(n-1) + \hat{e}(n) \quad (4.35c)$$

Từ (4.35a) và (4.35c) tính ra tạp âm lượng tử  $e_Q(n)$

$$e_Q(n) = \hat{f}(n) - f(n) = \hat{e}(n) - e(n) \quad (4.36)$$

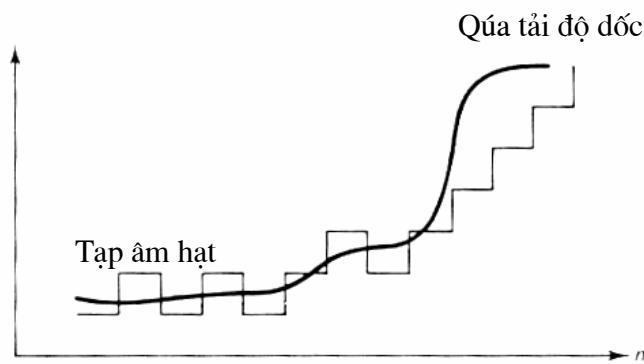
Trong DM,  $e(n)$  lượng tử hoá là hiệu giữa  $f(n)$  và  $\hat{f}(n-1)$ . Nếu dùng  $f(n-1)$  thay  $\hat{f}(n-1)$  thì máy phát không cần đến máy thu để tính  $\hat{f}(n-1)$ . Tuy vậy, tạp âm lượng tử hoá có thể tích luỹ từ pixel đến pixel kề đó. Điều này được minh họa trên hình 4.25. Đường cầu thang liên nét ( $\hat{f}_1(n)$ ) là kết quả điều chế delta khi  $\hat{f}(n-1)$ . Đường cầu thang vẽ chấm ( $\hat{f}_2(n)$ ) là kết quả khi dùng  $f(n-1)$ . Tín hiệu phục hồi khác với  $f(n)$  khá nhiều bởi vì máy thu không có  $f(n-1)$  và dùng  $\hat{f}(n-1)$  để xây dựng  $\hat{f}(n)$ , trong lúc sai số  $e(n)$  vì lượng tử hoá ở đầu máy phát lại được tạo ra từ  $f(n-1)$ .



**Hình 4.25:** Minh họa việc tích luỹ tạp âm lượng tử trong điều chế DM khi dùng  $f(n-1)$  để dự báo  $f(n)$  thay cho  $\hat{f}(n-1)$ .

- Đường bậc thang liên nét có chú thích  $\hat{f}_1(n)$  là tín hiệu phục hồi khi sử dụng  $\hat{f}(n-1)$ .
- Đường bậc thang vẽ chấm có chú thích  $\hat{f}_2(n)$  là tín hiệu phục hồi khi sử dụng  $f(n-1)$ .

Thông số quan trọng trong thiết kế DM là bước nhảy  $\Delta$ . Ta hãy xét 1 tín hiệu do DM phục hồi (hình 4.26). Trong vùng mà tín hiệu biến thiên chậm, tín hiệu phục hồi biến thiên nhanh quanh giá trị tín hiệu gốc. Đó là tạp âm hạt. Vì  $\Delta$  càng lớn thì tạp âm hạt càng nhỏ, do đó cần  $\Delta$  nhỏ. Khi tín hiệu tăng hay giảm nhanh nếu dùng  $\Delta$  nhỏ thì phải mất nhiều pixel trước khi  $\hat{f}(n)$  theo kịp  $f(n)$ . Tín hiệu phục hồi  $\hat{f}(n)$  trong vùng này bị mờ đó là hiện tượng méo quá tải độ dốc.  $\Delta$  càng bé méo này càng rõ nét, do đó lại cần  $\Delta$  to. Như vậy yêu cầu giảm tạp âm hạt và giảm méo quá tải độ dốc mâu thuẫn nhau và  $\Delta$  phải chọn sao cho dung hòa cả hai.



**Hình 4.26 :** Tạp âm hạt và méo quá tải độ dốc khi điều chế  $\Delta$ .

Hình 4.27 minh họa tính năng hệ DM.

Hình 4.27a và 4.27b phân biệt biểu diễn kết quả DM với các giá trị  $\Delta = 8\%$  và  $15\%$  của toàn dải động  $f(n_1, n_2)$ .



**Hình 4.27 :** Ví dụ mã hoá ảnh bằng điều chế DM.

- Trường hợp  $\Delta = 8\%$  của toàn dải động.
- Trường hợp  $\Delta = 15\%$  của toàn dải động.

Ảnh gốc dùng ảnh 512 x 512 pixel của hình 4.22a. Khi  $\Delta$  bé (hình 4.27a) thì tạp âm hạt nhỏ nhưng méo quá tải độ dốc lớn nên hình mờ. Khi  $\Delta$  tăng lên (hình 4.27b) thì méo quá tải độ dốc giảm nhưng những vùng tín hiệu biến đổi chậm bị tạp âm hạt rất rõ nét.

Để có ảnh phục hồi tốt bằng DM, cả tạp âm hạt lẫn méo quá tải độ dốc không đáng kể, phải dùng  $3 \sim 4$  bit/pixel. Ở DM có thể đạt tỷ lệ bit cao hơn 1 bit/pixel bằng cách lấy mẫu tín hiệu analog với tần số cao hơn tần số thường dùng để nhận được  $f(n_1, n_2)$ . Lấy mẫu tỷ lệ cao làm giảm độ dốc của tín hiệu digital  $f(n)$  cho nên có thể dùng  $\Delta$  nhỏ hơn mà vẫn không sợ tăng méo quá tải độ dốc. Ví dụ trên hình 4.28 cho ảnh mã hoá DM có tỷ lệ bit 2 bit/pixel. Để nhận được ảnh hình 4.28, kích thước ảnh nhận được giống với ảnh ở hình 4.27 nhưng tỷ lệ bit bây giờ là 2 bit/pixel.

Ảnh digital nội suy được mã hoá DM với  $\Delta = 12\%$  dài động, còn ảnh phục hồi lấy mẫu tỷ lệ thấp đi 2 lần theo phương nằm ngang. Kích thước ảnh nhận được giống với ảnh ở hình 4.27 nhưng tỷ lệ bit bây giờ là 2 bit/pixel.



**Hình 4.28** : Ảnh mã hoá DM ở tỷ lệ bit 2 bit/ pixel. Ảnh gốc là ảnh ở hình 4.22a.

### 3.3 § iQu chÔxung m· vi sai .

Điều chế xung mã vi sai (DPCM) có thể coi như DM mở rộng . DM hiệu tín hiệu  $e(n) = f(n) - \hat{f}(n-1)$  được lượng tử hoá . Tín hiệu  $\hat{f}(n-1)$  vừa mới mã hoá xong coi như dự báo của  $f(n)$ , và  $e(n)$  có thể coi như sai số giữa  $f(n)$  và dự báo của  $f(n)$ . Bên DPCM dự

báo của cường độ pixel hiện tại do nhiều cường độ pixel đã mã hoá trước cung cấp. Bên DM chỉ dùng 1 bit để mã hoá  $e(n)$ , bên DPCM dùng nhiều hơn 1 bit để mã hoá sai số.

Hình 4.29 vẽ 1 hệ DPCM. Để mã hoá cường độ pixel hiện tại  $f(n_1, n_2)$  ta dự báo  $f(n_1, n_2)$  bằng nhiều cường độ pixel phục hồi trước đó. Giá trị dự báo ký hiệu là  $f'(n_1, n_2)$ . Trên hình ta giả thiết là  $\hat{f}(n_1 - 1, n_2)$ ,  $\hat{f}(n_1, n_2 - 1)$ ,  $\hat{f}(n_1 - i, n_2 - i)$ ... đều được phục hồi trước khi mã hoá  $f(n_1, n_2)$ . Chúng ta cố giảm phương sai của  $e(n_1, n_2) = f(n_1, n_2) - f'(n_1, n_2)$  bằng cách dùng các pixel đã mã hoá trước để dự báo  $f(n_1, n_2)$ . Sai số dự báo  $e(n_1, n_2)$  được 1 hệ PCM lượng tử hoá bằng bộ lượng tử hoá đều hoặc không đều.  $e(n_1, n_2)$  đã lượng tử hoá tức là  $\hat{e}(n_1, n_2)$  được truyền đi. Ở đâu thu,  $\hat{e}(n_1, n_2)$  được kết hợp vào  $f'(n_1, n_2)$ , tức là giá trị dự báo của  $f(n_1, n_2)$ .

Bởi vì cả máy phát và máy thu đều biết các giá trị pixel đã phục hồi trước đó và cách dự báo  $f(n_1, n_2)$  trên cơ sở các pixel đã phục hồi trước cho nên máy phát và máy thu có giá trị  $f'(n_1, n_2)$  như nhau. Giá trị đã phục hồi  $\hat{f}(n_1, n_2)$  cũng cần cho đầu máy phát bởi vì nó được dùng để mã hoá cường độ các pixel chưa được mã hoá. Đường vẽ chấm trên hình cho thấy  $\hat{f}(n_1, n_2)$  vừa được tính ra để mã hoá các cường độ pixel nói trên. Cũng như bên DM, các giá trị vừa phục hồi được đem dùng để khởi phái chuyển tạp âm lượng tử đi.

Những phương trình của hệ DPCM trên hình 4.29 là :

$$e(n_1, n_2) = f(n_1, n_2) - f'(n_1, n_2) \quad (4.37a)$$

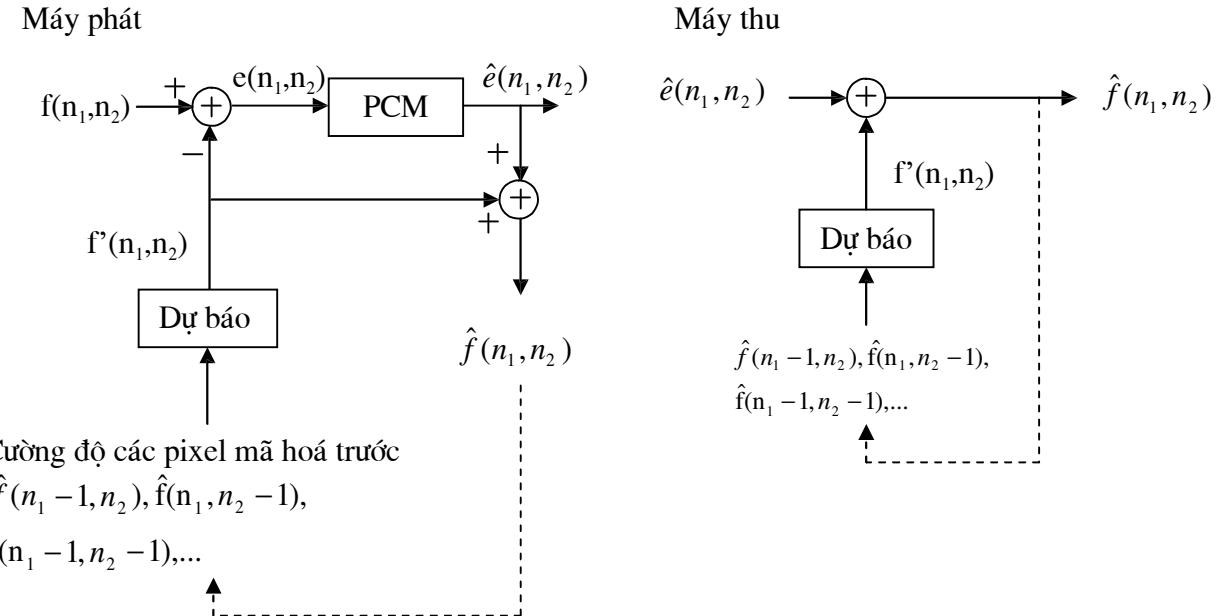
$$\hat{e}(n_1, n_2) = Q[e(n_1, n_2)] \quad (4.37b)$$

$$\hat{f}(n_1, n_2) = f(n_1, n_2) - \hat{e}(n_1, n_2) \quad (4.37c)$$

Trong đó  $Q[e(n_1, n_2)]$  là  $e(n_1, n_2)$  mà hệ PCM đã lượng tử hoá. Từ phương trình (4.37a) và (4.37c) tính ra tạp âm lượng tử  $e_Q(n_1, n_2)$  theo phương trình sau:

$$e_Q(n_1, n_2) = \hat{f}(n_1, n_2) - f(n_1, n_2) = \hat{e}(n_1, n_2) - e(n_1, n_2) \quad (4.38)$$

Hệ DPCM ở hình 4.37 có thể coi là dạng mở rộng của PCM. Khi cho  $f'(n_1, n_2) = 0$  thì hệ DPCM trở thành PCM.



**Hình 4.29**: Điều chế xung mã vi sai.

Trong hệ DPCM dự báo  $f(n_1, n_2)$  bằng cách tổ hợp tuyến tính các giá trị đã phục hồi trước.

$$f'(n_1, n_2) = \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) \hat{f}(n_1 - k_1, n_2 - k_2) \quad (4.39)$$

Trong đó  $R_a$  là miền của  $(k_1, k_2)$  trong đó  $a(k_1, k_2) \neq 0$ . Thông thường  $f'(n_1, n_2)$  nhận được bằng cách tổ hợp tuyến tính  $\hat{f}(n_1 - 1, n_2)$ ,  $\hat{f}(n_1, n_2 - 1)$  và  $\hat{f}(n_1 - 1, n_2 - 1)$ . Vì làm dự báo của  $f(n_1, n_2)$  là để giảm bớt phương sai của  $e(n_1, n_2)$ , cho nên sẽ là hợp lý khi ước lượng  $a(k_1, k_2)$  bằng cách tối thiểu hoá:

$$E[e^2(n_1, n_2)] = E\left[\left(f(n_1, n_2) - \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) \hat{f}(n_1 - k_1, n_2 - k_2)\right)^2\right] \quad (4.40)$$

Vì  $\hat{f}(n_1, n_2)$  là hàm của  $a(k_1, k_2)$  là phụ thuộc vào loại hình bộ lượng tử hoá nên giải phương trình 4.40 là một bài toán phi tuyến. Vì  $\hat{f}(n_1, n_2)$  là  $f(n_1, n_2)$  đã lượng tử hoá, do đó nó là một biểu diễn hợp lý của  $f(n_1, n_2)$  các hệ số dự báo  $a(k_1, k_2)$  nhận được bằng cách tối thiểu hoá:

$$E \left[ \left( f(n_1, n_2) - \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) f(n_1 - k_1, n_2 - k_2) \right)^2 \right] \quad (4.41)$$

Vì hàm được tối thiểu hoá ở (4.41) là 1 dạng câu phương của  $a(k_1, k_2)$  nên giải (4.41) sẽ đưa đến 1 hệ tuyến tính những phương trình có dạng như sau:

$$R_f(l_1, l_2) = \sum_{(k_1, k_2) \in R_a} a(k_1, k_2) R_f(l_1 - k_1, l_2 - k_2) \quad (4.42)$$

Trong đó  $f(n_1, n_2)$  là quá trình ngẫu nhiên dùng với hàm tương quan là  $R_f(n_1, n_2)$ .

Hình 4.30 minh họa đặc tính của 1 hệ DPCM. Hình này cho kết quả của 1 hệ DPCM ở tỷ lệ bit là 3 bit/pixel. ảnh gốc là ảnh ở hình 4.22a. Hệ PCM ở hình 4.30 dùng 1 bộ lượng tử hoá không đều. Các hệ số dự báo  $a(k_1, k_2)$  dùng trong ví dụ này là :

$$a(1,0) = a(0,1) = 0,95 \text{ và } a(1,1) = -0,95$$

Với tỷ lệ 3 bit/pixel thì kết quả của DPCM là 1 ảnh có chất lượng tốt. Vì hệ PCM là một bộ phận trong DPCM cho nên có thể dùng kỹ thuật tạp âm giả của Robert vào hệ DPCM. Tuy nhiên tín hiệu sai số  $e(n_1, n_2)$  được lượng tử hoá trong hệ DPCM biến thiên nhanh từ pixel này sang pixel khác và ảnh được phục hồi ít có những đường viền hơn bên PCM. Vì thế cho nên kỹ thuật Robert rất có ích trong hệ PCM nhưng trong hệ DPCM lại không cần thiết lắm. Ngoài ra dùng 1 hệ phục hồi ảnh để làm giảm tạp âm lượng tử trong DPCM cũng không cần thiết lắm. Cả chuỗi sai số  $e(n_1, n_2)$  và tạp âm lượng tử  $e_Q(n_1, n_2)$  đều có khổ rộng và làm giảm  $e_Q(n_1, n_2)$  trong  $\hat{e}(n_1, n_2) = e(n_1, n_2) + e_Q(n_1, n_2)$  không hiệu quả lắm .



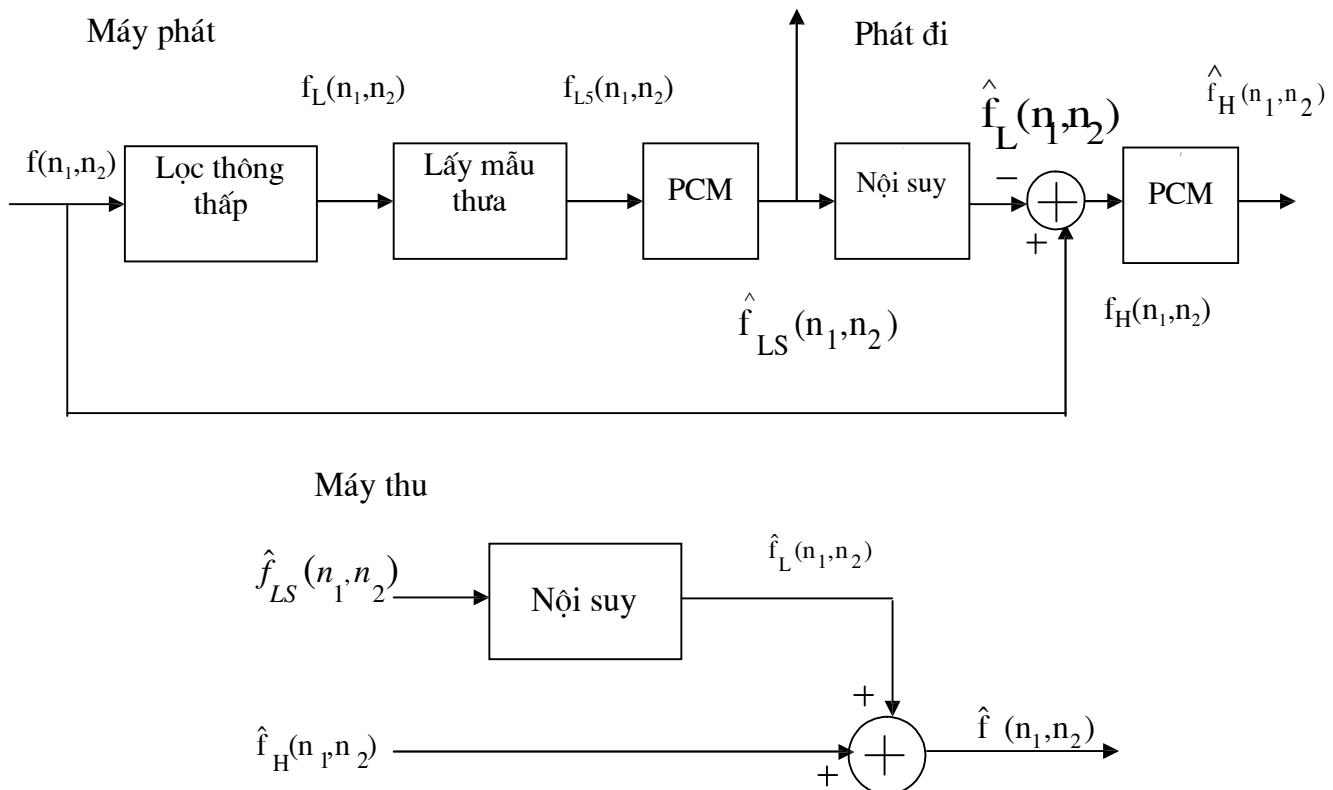
**Hình 4.30** : Ví dụ về mã hoá bằng điều xung mã vi sai ở tỷ lệ 3 bit/pixel.  
Ảnh gốc là ảnh ở hình 4.22a.

Vì dự báo  $f(n_1, n_2)$  từ các pixel lân cận gặp khó khăn trong những vùng ở ngoài rìa, khi mà độ tương phản tại chỗ tương đối cao, và tín hiệu sai số  $e(n_1, n_2)$  ở đó lớn hơn. Cùng 1 mức tạp âm thì ở vùng độ tương phản cao ít nhận thấy hơn ở vùng độ tương phản thấp. Kiến thức này được khai thác để xác định các mức lượng tử của  $e(n_1, n_2)$  trong hệ DPCM bởi vì biên độ của  $e(n_1, n_2)$  có liên quan với độ tương phản tại chỗ.

### 3.4. Cắt bé m. ho\_ 2 k\_ nh.

Trong một bộ mã hoá 2 kênh 1 ảnh  $f(n_1, n_2)$  được chia thành 2 phần là thành phần thấp và thành phần cao. Thành phần thấp  $f_L(n_1, n_2)$  chủ yếu là gồm những thành phần tần số thấp và đại biểu độ chói trung bình tại chỗ. Thành phần cao  $f_H(n_1, n_2)$  gồm chủ yếu các thành phần tần số cao và đại biểu cho độ tương phản tại chỗ của  $f(n_1, n_2)$ .

Vì thành phần thấp là 1 dạng của  $f(n_1, n_2)$  sau khi đã đi qua bộ lọc thông thấp cho nên nó sẽ bị lấy mẫu rất thừa tuỳ theo loại bộ lọc được sử dụng. Các thành phần cao có thể lượng tử hoá thô bởi vì nó không chứa tin tức về độ chói trung bình tại chỗ và vì các miền có biên độ  $f_H(n_1, n_2)$  lớn thì độ tương phản tại chỗ cao, nên ở 1 mức tạp âm đã cho, ở đấy khó nhận thấy hơn.



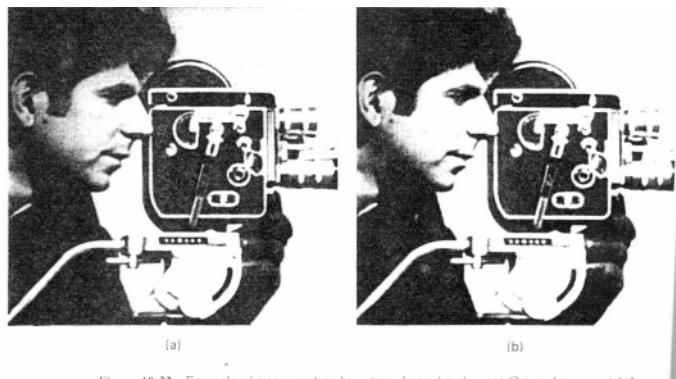
**Hình 4.31.** Bộ mã hoá ảnh hai kênh.

Hình 4.31 vẽ 1 bộ mã hoá ảnh 2 kênh. ảnh gốc  $f(n_1, n_2)$  qua bộ lọc thông thấp FIR. Thành phần thấp  $f_L(n_1, n_2)$  được lấy mẫu con với hệ số 8 x 8. Thành phần thấp đã lấy mẫu con  $f_{LS}(n_1, n_2)$  được biểu diễn rõ nét, thông thường là 8 ~ 10 bit/pixel nhưng sự đóng góp vào tỷ lệ bit tổng chỉ khoảng 0,1 ~ 0,2 bit/pixel do lấy mẫu thừa. Ước lượng của thành phần thấp  $f_L(n_1, n_2)$  có thể nhận được bằng cách nội suy  $\hat{f}_{LS}(n_1, n_2)$  và được ký hiệu là  $\hat{f}_L(n_1, n_2)$ . Thành phần cao  $f_H(n_1, n_2)$  nhận được bằng cách lấy  $f(n_1, n_2)$  trừ đi  $\hat{f}_L(n_1, n_2)$  sau đó được 1 hệ PCM lượng tử hoá. Hệ này có thể sử dụng lượng tử hoá không đều và kỹ thuật Robert. Khi chọn các mức lượng tử trong bộ lượng tử hoá không đều ta có thể khai thác đặc tính về tạp âm không hiện rõ ở vùng có độ tương phản cao tại chỗ. Dùng 3 bit/pixel để mã hoá  $f_H(n_1, n_2)$  đã có chất lượng tốt đối với những ảnh điển hình. Ở máy thu  $\hat{j}_L(n_1, n_2)$  nhận được bằng cách nội suy  $\hat{j}_{LS}(n_1, n_2)$ . Kết quả nhận được đem kết hợp với  $\hat{j}_H(n_1, n_2)$  để tạo ra ảnh phục hồi  $\hat{f}(n_1, n_2)$ . Bộ mã hoá 2 kênh giống hệ DPCM. Thành phần thấp  $\hat{j}_L(n_1, n_2)$  coi như giá trị dự báo  $f'(n_1, n_2)$  trong DPCM, thành phần cao  $f_H(n_1, n_2)$  coi như sai số  $e(n_1, n_2) = f(n_1, n_2) - f'(n_1, n_2)$  trong DPCM. Sự khác nhau chỉ ở cách lấy ra  $\hat{f}_L(n_1, n_2)$  và  $f'(n_1, n_2)$ . Trong bộ mã hoá 2 kênh người ta lấy  $\hat{f}_L(n_1, n_2)$  trực tiếp từ  $f(n_1, n_2)$ .

Vì máy thu không có  $f(n_1, n_2)$  cho nên phải truyền  $\hat{j}_L(n_1, n_2)$  đi. Trong hệ DPCM  $f'(n_1, n_2)$  lấy từ các cường độ pixel đã phục hồi trước đó, do đó không cần phải truyền nó đi. Việc truyền  $\hat{j}_L(n_1, n_2)$  tuy có làm tăng tỷ lệ bit nhưng cũng có những ưu điểm.

Trong hệ DPCM  $f'(n_1, n_2)$  nhận được bằng phương pháp đệ quy từ các cường độ pixel phục hồi trước đó cho nên những điều làm khi mã hoá pixel hiện tại sẽ ảnh hưởng đến các pixel được mã hoá về sau. Do đó mọi sai số của kênh truyền không những ảnh hưởng đến cường độ pixel hiện tại mà còn ảnh hưởng đến cường độ các pixel về sau. Ngoài ra tác động vào  $e(n_1, n_2)$  để cải thiện chất lượng ảnh có những khó khăn nhất định bởi vì thay đổi  $e(n_1, n_2)$  của pixel hiện nay sẽ ảnh hưởng đến cường độ những pixel về sau. Trong bộ mã hoá 2 kênh thì sai số kênh truyền hoặc sự tác động vào sự tương phản tại chỗ  $f_H(n_1, n_2)$  chỉ khu trú ở một vùng nhỏ.

Trong bộ mã hoá hai kênh ảnh được chia làm hai kênh, kênh thông thấp và kênh thông cao, mỗi thành phần được một bộ mã hoá riêng phù hợp với kênh đó xử lý. Tất nhiên ta cũng có thể đem ảnh chia ra nhiều dải (điển hình là 16 dải) bằng các bộ lọc thông dải rồi mã hoá tín hiệu trong mỗi giải bằng một thiết bị phù hợp với dải đó.



**Hình 4.32:** Ví dụ về mã hoá ảnh bằng bộ mã hoá hai kênh.

- a) Ảnh gốc 512x512 pixel.
- b) Ảnh mã hoá ở tỷ lệ bít  $3\frac{1}{8}$  bit/pixel .

Phương pháp này thoát tiên được dùng để mã hoá tiếng nói, sau đó phát triển ra mã hoá ảnh. Hình 4.32a là ảnh gốc digital 512x512 pixel, tỷ lệ 8 bit/pixel. Hình 4.32b là kết quả mã hoá hai kênh ở tỷ lệ  $3\frac{1}{8}$  bit/pixel.

Tần số lấy mẫu con để tạo ra  $f_{LS}(n_1, n_2)$  là 1:64 còn  $f_{LS}(n_1, n_2)$  được mã hoá ở tỷ lệ 8 bit/pixel. Như vậy là tỷ lệ để mã hoá thành phần thấp  $f_L(n_1, n_2)$  là 1/8 bit/pixel. Các thành phần cao  $f_H(n_1, n_2)$  được lượng tử hoá ở tỷ lệ 3 bit/pixel.

### 3.5. Mã hoá hình chéo.

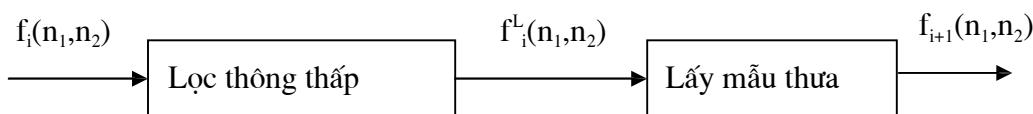
Một hình chéo là một cấu trúc số liệu cung cấp liên tiếp những tin tức cô đọng của một ảnh. Hình chéo cũng có ích trong những ứng dụng về xử lý ảnh kể cả mã hoá ảnh và phân tích ảnh.

Có nhiều cách biểu diễn ảnh có thể coi như cấu trúc hình chéo. Sau đây là một trong những cách đó: Cấu trúc hình chéo gồm một ảnh gốc và một chuỗi ảnh tiếp theo, với khả năng phân giải kém hơn (mờ hơn).

Giả sử  $f_0(n_1, n_2)$  là một ảnh gốc  $N \times N$  pixel trong đó  $N=2^M+1$  chẳng hạn 129x129, 257x257, 513x513,... Có thể từ một ảnh  $2^M \times 2^M$  pixel tạo ra một ảnh  $(2^M+1) \times (2^M+1)$  pixel.

Chẳng hạn chỉ cần lặp lại dòng cuối và cột cuối . Để đơn giản ta giả thiết là ảnh vuông. Ta lấy  $f_0(n_1, n_2)$  là ảnh ở đáy hình chóp. Ảnh ở mức trên đó nhận được bằng cách lọc thông thấp  $f_0(n_1, n_2)$  rồi tiến hành lấy mẫu con là  $f_1(n_1, n_2)$ . Vì lấy mẫu con cho nên kích thước ảnh  $f_1(n_1, n_2)$  bé hơn ảnh  $f_0(n_1, n_2)$  và nó là ảnh lớp trên kề đáy hình chóp. Ta gọi  $f_1(n_1, n_2)$  là ảnh mức 1 của hình chóp. Ảnh mức 2 nhận được bằng cách lọc thông thấp ảnh mức 1 và lấy mẫu con, kết quả là  $f_2(n_1, n_2)$  .

Cứ thế áp dụng quy trình cho các mức cao hơn như  $f_3(n_1, n_2)$ ,  $f_4(n_1, n_2)$ ...Quá trình tạo ra  $f_{i+1}(n_1, n_2)$  từ  $f_i(n_1, n_2)$  được biểu diễn trên hình 4.33 . Giả thiết ảnh ở mức k là  $f_k(n_1, n_2)$  nằm trên cùng hình chóp. Càng lên trên kích thước càng nhỏ và ảnh càng mờ (độ phân biệt trong không gian kém).



**Hình 4.33 :** Quá trình tạo ảnh  $f_{i+1}(n_1, n_2)$  ở lớp thứ  $i+1$  từ ảnh  $f_i(n_1, n_2)$  ở lớp thứ  $i$ .

Các ảnh  $f_i(n_1, n_2)$  với  $0 \leq i \leq k$  coi như những ảnh có nhiều độ phân biệt mà hình thành hình chóp.

Tùy theo loại lọc thông thấp được dùng và cách lấy mẫu con kết quả lọc, có nhiều phương án hình chóp. Trong hình chóp Gauss bộ lọc thông thấp có  $5 \times 5$  điểm đáp ứng xung  $h(n_1, n_2)$ .

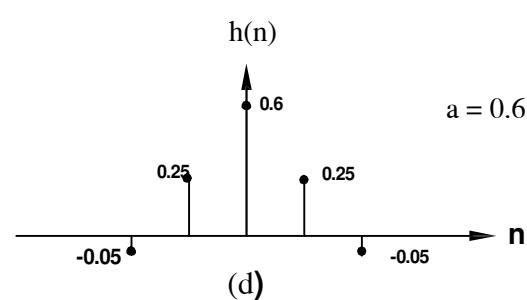
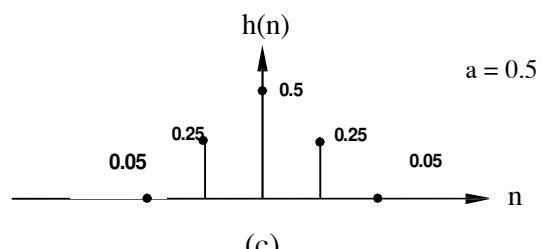
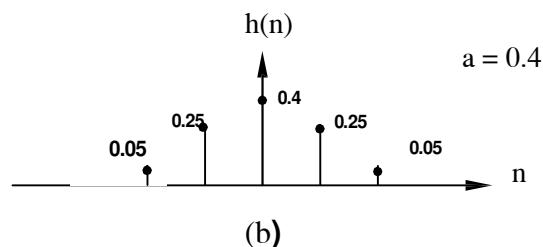
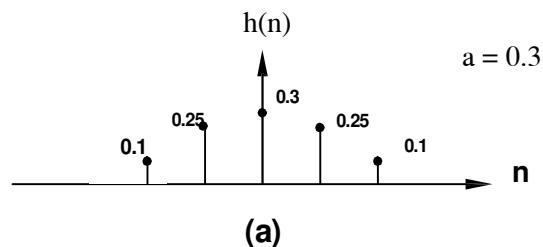
$$h(n_1, n_2) = h(n_1) h(n_2) \quad (4.44a)$$

$$h(n) = \begin{cases} a, & n = 0 \\ \frac{1}{4}, & n = \pm 1 \\ \frac{1}{4} - \frac{a}{2}, & n = \pm 2 \end{cases} \quad (4.44b)$$

Hằng số  $a$  trong (4.44b) là hằng số tự do, được chọn giữa 0,3 và 0,6. Hình 4.34 vẽ chuỗi  $h(n)$  với  $a = 0,3 ; 0,4 ; 0,5 ; 0,6$  . Khi  $a = 0,4$   $h(n)$  có dạng gần đúng Gauss và do

đó gọi là hình chóp Gauss. Cách chọn  $h(n_1, n_2)$  trong (4.44) đảm bảo  $h(n_1, n_2)$  có pha bằng không và bộ lọc thông suốt đối với thành phần 1 chiều:

$$\left[ H(0,0)=1, \sum_{n_1} \sum_{n_2} h(n_1, n_2) = 1 \right]$$

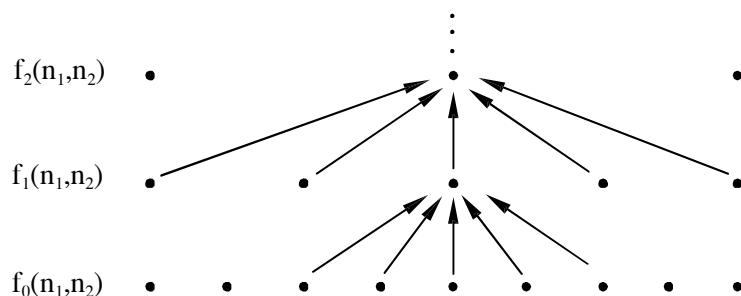


**Hình 4.34 :** Đáp ứng xung  $h(n)$  theo hàm thông số  $a$ . Bộ lọc 2\_D thông thấp  $h(n_1, n_2)$  dùng trong biểu diễn ảnh bằng hình chóp Gauss nhận được từ  $h(n)$  theo  $h(n_1, n_2) = h(n_1) h(n_2)$ .

Ảnh  $f_0^L(n_1, n_2)$  nhận được từ  $f_0(n_1, n_2) * h(n_1, n_2)$  và sau đó lấy mẫu con với hệ số 4, tức là hệ số 2 dọc  $n_1$  và hệ số 2 dọc  $n_2$ . Ảnh đã lấy mẫu có dạng:

$$f_1(n_1, n_2) = \begin{cases} f_0^L(2n_1, 2n_2) & 0 \leq n_1 \leq 2^{M-1}; 0 \leq n_2 \leq 2^{M-1} \\ 0 & \text{ở các nơi khác} \end{cases} \quad (4.45)$$

Kích thước của  $f_1(n_1, n_2)$  là  $(2^{M-1}+1) \times (2^{M-1}+1)$  pixel gần bằng  $1/4$  kích thước  $f_0(n_1, n_2)$ . Từ (4.45) thấy chỉ cần tính  $f_0^L(n_1, n_2)$  với các giá trị chẵn của  $n_1$  và  $n_2$  là được  $f_1(n_1, n_2)$ . Các ảnh ở mức cao hơn nhận được bằng cách lặp lại nhiều lần phép lọc thông thấp và lấy mẫu con. Một biểu diễn hình học của quá trình này với ảnh trong không gian 1 chiều như trên hình (4.35).



**Hình 4.35 :** Biểu diễn hình học trong không gian 1 chiều của cách tạo hình chéo Gauss.

Ví dụ biểu diễn ảnh 513 x 513 pixel bằng hình chéo Gauss như trên hình 4.36.



**Hình 4.36 :** Ví dụ về biểu diễn bằng hình chéo Gauss ảnh 513 x 513 pixel với k=4.

Biểu diễn hình chót Gauss có thể dùng để phát triển 1 phương pháp mã hóa ảnh. Để mã hóa ảnh gốc  $f_0(n_1, n_2)$  ta đem mã hóa  $f_1(n_1, n_2)$  và hiệu giữa  $f_0(n_1, n_2)$  với giá trị dự báo của nó suy từ  $f_1(n_1, n_2)$ . Giả sử ta dự báo  $f_0(n_1, n_2)$  bằng cách nội suy  $f_1(n_1, n_2)$ . Gọi ảnh nội suy ra là  $f'(n_1, n_2)$  ta tìm ra sai số đã mã hóa là  $e_0(n_1, n_2)$  từ :

$$\begin{aligned} e_0(n_1, n_2) &= f_0(n_1, n_2) - I[f_1(n_1, n_2)] \\ &= f_0(n_1, n_2) - f'(n_1, n_2) \end{aligned} \quad (4.46)$$

Trong đó  $I[.]$  là thuật toán nội suy không gian. Quá trình nội suy làm giãn kích thước  $f_1(n_1, n_2)$  và do đó kích thước  $f'(n_1, n_2)$  bằng  $f_0(n_1, n_2)$ . Một ưu điểm của mã hóa  $f_1(n_1, n_2)$  và  $e_0(n_1, n_2)$  thay cho  $f_0(n_1, n_2)$  là có thể dùng bộ mã hóa phù hợp với đặc tính của  $f_1(n_1, n_2)$  và  $e_0(n_1, n_2)$ . Nếu ta không lượng tử hóa  $f_1(n_1, n_2)$  và  $e_0(n_1, n_2)$  thì từ (4.46) có thể khôi phục nguyên vẹn  $f_0(n_1, n_2)$  bằng:

$$f_0(n_1, n_2) = I[f_1(n_1, n_2)] + e_0(n_1, n_2) \quad (4.47)$$

Khi mã hóa ảnh,  $f_1(n_1, n_2)$  và  $e_0(n_1, n_2)$  đều được lượng tử hóa và ảnh phục hồi  $\hat{f}_0(n_1, n_2)$  nhận được từ (4.47) bằng:

$$\hat{f}_0(n_1, n_2) = I[\hat{f}_1(n_1, n_2)] + \hat{e}_0(n_1, n_2) \quad (4.48)$$

Trong đó  $\hat{f}_0(n_1, n_2)$  và  $\hat{e}_0(n_1, n_2)$  là  $f_0(n_1, n_2)$  và  $e_0(n_1, n_2)$  đã lượng tử hóa

Nếu ta dừng lại ở đây thì cấu trúc của phương pháp mã hóa y hệt như bộ mã hóa 2 kênh, ảnh  $f_1(n_1, n_2)$  có thể coi như thành phần thấp được lấy mẫu con  $f_{LS}(n_1, n_2)$  và  $e_0(n_1, n_2)$  coi như thành phần cao  $f_H(n_1, n_2)$  trong hệ ở hình 4.31.

Ý tưởng cho rằng 1 ảnh có thể phân tích thành 2 thành phần có đặc tính rất khác nhau cũng có thể áp dụng cho mã hóa  $f_1(n_1, n_2)$  ta mã hóa  $f_2(n_1, n_2)$  và  $e_1(n_1, n_2)$  theo :

$$e_1(n_1, n_2) = f_1(n_1, n_2) - I[f_2(n_1, n_2)] \quad (4.49)$$

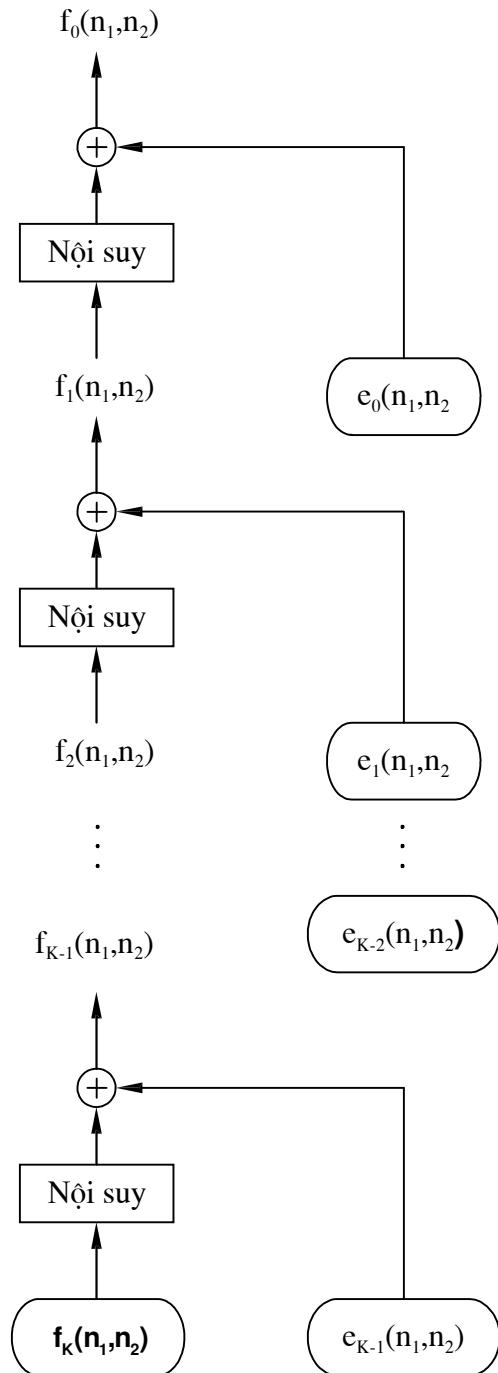
Quá trình này có thể được lặp lại. Thay vì mã hóa  $f_i(n_1, n_2)$  ta có thể mã hóa  $f_{i+1}(n_1, n_2)$  và  $e_i(n_1, n_2)$  theo :

$$e_i(n_1, n_2) = f_i(n_1, n_2) - I[f_{i+1}(n_1, n_2)] \quad (4.50)$$

Nếu ta không lượng tử hóa  $f_{i+1}(n_1, n_2)$  và  $e_i(n_1, n_2)$  thì dựa vào (4.50) ta có thể phục hồi chính xác  $f_i(n_1, n_2)$  từ  $f_{i+1}(n_1, n_2)$  và  $e_i(n_1, n_2)$  nhờ :

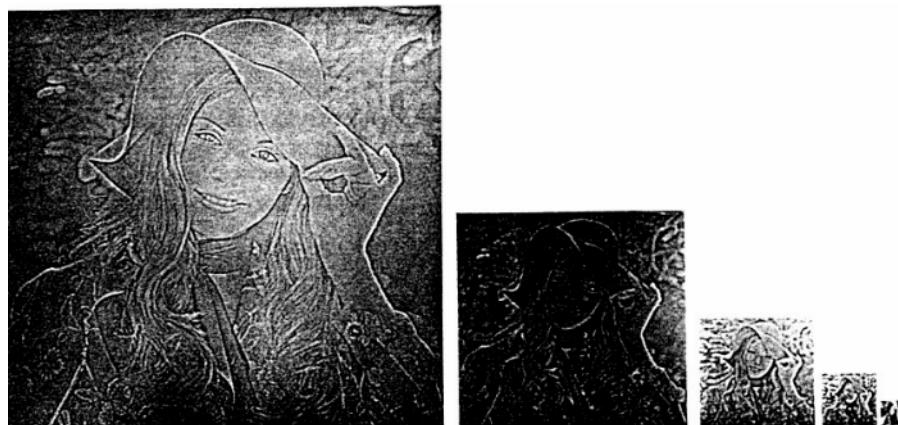
$$f_i(n_1, n_2) = I[(f_{i+1}(n_1, n_2))] + e_i(n_1, n_2) \quad (4.51)$$

Chúng ta cứ thế lặp lại quá trình cho đến khi đến đỉnh hình chóp, như trên hình 4.37. Thay vì cho mã hoá  $f_0(n_1, n_2)$  ta mã hoá  $e_i(n_1, n_2)$  với :  $0 \leq i \leq k-1$  và  $f_k(n_1, n_2)$ . Một ví dụ của  $e_i(n_1, n_2)$  với :  $0 \leq i \leq k-1$  và  $f_k(n_1, n_2)$  trong trường hợp ảnh gốc  $f_0(n_1, n_2)$  có  $513 \times 513$  pixel với  $k = 4$  được vẽ trên hình 4.38.



**Hình 4.37 :** Cách tạo hình chóp Laplacian. ảnh gốc  $f(n_1, n_2)$  được phục hồi từ  $e_i(n_1, n_2)$  với  $0 \leq i \leq k-1$  và  $f_k(n_1, n_2)$ .

Nếu  $e_i(n_1, n_2)$  và  $f_k(n_1, n_2)$  không được lượng tử hoá thì có thể phục hồi hoàn toàn  $f_0(n_1, n_2)$  bằng phép tính đệ quy phương trình (4.51) cho các giá trị  $i = k - 1, k - 2, \dots, 0$ .



**Hình 4.38 :** Ví dụ biểu diễn ảnh bằng hình chót Laplace. Ảnh gốc là ảnh  $f_0(n_1, n_2)$  với  $513 \times 513$  pixel ở hình 4.36.  $e_i(n_1, n_2)$  với  $0 \leq i \leq 3$ , và  $f_4(n_1, n_2)$ .

Lưu ý rằng phương trình (4.51) độc lập với cách chọn thuật toán nội suy  $I[.]$  phương trình (4.51) có thể dùng để phục hồi  $f_0(n_1, n_2)$  từ giá trị lượng tử hóa của  $e_i(n_1, n_2)$  và  $f_k(n_1, n_2)$ . Các ảnh  $f_k(n_1, n_2)$  và  $e_i(n_1, n_2)$  với  $0 \leq i \leq k-1$  hình thành 1 hình chót gọi là chót Laplacian trong đó  $e_i(n_1, n_2)$  là ảnh ở mức thứ  $i$  của hình chót và  $f_k(n_1, n_2)$  là ảnh ở trên đỉnh chót.

$$e_0(n_1, n_2) = f_0(n_1, n_2) - I[f_1(n_1, n_2)] \quad (4.52)$$

Trên hình 4.33  $f_1(n_1, n_2)$  là kết quả lấy mẫu con  $f_0(n_1, n_2) * h(n_1, n_2)$ . Lấy gần đúng thật toán nội suy  $I[.]$ , coi như phép toán ngược của lấy mẫu con.

$$\begin{aligned} e_0(n_1, n_2) &\approx f_0(n_1, n_2) - f_0(n_1, n_2) * h(n_1, n_2) \\ &= f_0(n_1, n_2) * (f_0(n_1, n_2) - h(n_1, n_2)) \end{aligned} \quad (4.53)$$

Bởi vì  $h(n_3, n_3)$  có đặc tính thông thấp cho nên  $e_0(n_1, n_2)$  có tính thông cao. Ta xét  $e(n_1, n_2)$  là ở mức thứ nhất của hình chót Laplacian. Theo 1 bước giống như bước đã đưa tới phương trình (4.53) và thêm một số giả thiết ta nhận được :

$$I[e_1(n_1, n_2)] \approx f_0(n_1, n_2) * h_1(n_1, n_2) \quad (4.54)$$

Trong đó :  $h_1(n_1, n_2) = h(n_1, n_2) - h(n_1, n_2) * h(n_1, n_2)$  (4.55)

Từ phương trình (4.54) kết quả nội suy  $e_1(n_1, n_2)$  sao cho kích thước của nó giống  $f_0(n_1, n_2)$  là gần đúng với kết quả lọc  $f_0(n_1, n_2)$  bằng  $h(n_1, n_2)$ .

Bởi vì  $h(n_1, n_2)$  là bộ lọc thông thấp,  $h_1(n_1, n_2)$  trong (4.55) là bộ lọc thông dải. Nếu ta tiếp tục cứ thế phân tích ta sẽ nhận thấy kết quả nội suy liên tiếp  $e_i(n_1, n_2)$  với  $1 \leq i \leq k-1$  là 1 chuỗi lọc  $f_0(n_1, n_2)$  qua nhiều bộ lọc thông dải. Khi ta tăng  $i$  từ 1 đến  $k-1$  đáp tuyến tần số của bộ lọc thông dải có dài thông ngày càng hẹp với tần số của dải thôn g thấp ngày càng giảm thấp xuống. Nếu  $h(n_1, n_2)$  có dạng Gauss thì  $h(n_1, n_2) * h(n_1, n_2)$  cũng có dạng Gauss. Nếu  $h(n_1, n_2)$  có dạng Gauss thì theo phương trình (4.55) bộ lọc thông dải sẽ có đáp ứng xung bằng hiệu của 2 hàm Gauss. Hiệu của 2 hàm Gauss gần đúng bằ ng Laplacian của Gauss, do đó có tên gọi là hình chóp Laplacian.

Từ sự thảo luận trên thấy rằng phương pháp mã hoá hình chóp có thể coi như mă hoá nhiều kênh. Trong mă hoá nhiều kênh ảnh được chia ra nhiều dải tầng hẹp và mỗi dải mă hoá bằng thiết bị phù hợp với nó.



**Hình 4.39:** Ví dụ của bộ mă hoá hình chóp Laplacian với  $K=4$  tại  $\frac{1}{2}$  bít/pixel.

Ảnh gốc sử dụng là 513x513 pixel  $f_0(n_1, n_2)$  trong hình 4.36.

Trong phương pháp mă hoá hình chóp cũng là sự lọc thông dải dưới dạng ẩn và các bộ lọc thông dải nhận được 1 cách ngẫu nhiên, còn trong bộ mă hoá nhiều kênh thì các bộ lọc thông dải được thiết kế bằng lý thuyết.

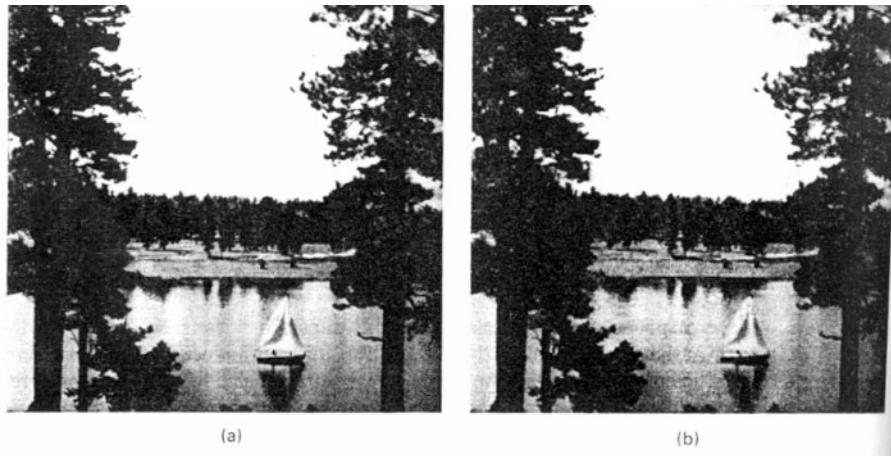
Hình 4.39 cho thấy chất lượng của 1 hệ mã hoá ảnh trong đó  $f_k(n_1, n_2)$  và  $e_i(n_1, n_2)$  với  $0 \leq i \leq k-1$  được mã hoá bằng những thiết bị phù hợp với đặc tính tín hiệu. Nói cách định tính là những ảnh ở mức cao có phương sai lớn hơn và được gán nhiều bit/pixel hơn. May mắn là chúng lại có kích thước bé. Hình 4.39 cho thấy 1 ảnh mã hoá ở tỷ lệ 1/2 bit /pixel. Ảnh gốc  $f_0(n_1, n_2)$  có 513 x 513 pixel như trên hình 4.36. Trong ví dụ này tỷ lệ bit thấp 1 bit/pixel có thể thực hiện bằng phép mã hoá entropy và khai thác nhận xét rằng phần lớn pixel của ảnh  $e_0(n_1, n_2)$  với kích thước 513 x 513 pixel được lượng tử hoá bằng không. Một ưu điểm chính của phương pháp mã hoá hình chéo là có thể truyền đi tuân tự. Thoạt tiên truyền ảnh  $f_k(n_1, n_2)$  ở đỉnh hình chéo và nội suy nó ở đầu thu ta được 1 ảnh rất mờ. Sau đó truyền  $e_{k-1}(n_1, n_2)$  tái cấu trúc  $f_{k-1}(n_1, n_2)$  có độ nét cao hơn  $f_k(n_1, n_2)$ . Cứ như thế lặp đi lặp lại quá trình thì ảnh tái cấu trúc ở đầu thu ngày càng có độ nét cao hơn và như vậy có khi không cần truyền tới ảnh gốc cũng đã có thể dừng việc truyền tín hiệu. Chẳng hạn nhìn trên 1 ảnh mờ mờ ta đã có thể xác định xem có phải là cái ta cần đến hay không, nếu không cần thì không truyền tiếp. Các ảnh từ đỉnh đến đáy hình chéo có kích thước ngày càng lớn dần, cái sau gấp 4 cái trước và khi dừng ở những ảnh kích thước nhỏ thì công tính toán cũng chưa nhiều.

Ngoài việc mã hoá ảnh, hình chéo Laplacian còn được sử dụng trong những ứng dụng khác. Chẳng hạn, như đã thảo luận ở trên, kết quả quy trình lặp phép nội suy  $e_1(n_1, n_2)$  sao cho kích cỡ của nó cũng như  $f_0(n_1, n_2)$  có thể coi như là xấp xỉ với kết quả lọc  $f_0(n_1, n_2)$  bằng Laplacian của một hàm Gauss. Như đã thảo luận ở tiết 3.3 chương 2, những điểm đi qua giá trị không của kết quả lọc  $f_0(n_1, n_2)$  bằng Laplacian của một hàm Gauss là những điểm biên trong phương pháp dò biên của Marr và Hildreth

### **3.6. M· ho· thÝch nghi vµ l- i ng tö ho· vDt-**

Các phương pháp mã hoá dạng sóng thảo luận trên đây có thể điều chỉnh để phù hợp với đặc trưng tại chỗ ở từng vùng của ảnh. Trong hệ PCM các mức lượng tử có thể chọn thích nghi. Trong hệ DM bước nhảy  $\Delta$  có thể chọn thích nghi. Ở những vùng mà cường độ biến đổi chậm thì chọn  $\Delta$  bé để giảm tạp âm hạt. Ở những vùng mà cường độ tăng hoặc giảm nhanh thì chọn  $\Delta$  to để giảm méo quá tải độ dốc. Trong hệ DPCM các hệ số dự báo và các mức lượng tử có thể chọn thích nghi. Trong hệ mã hoá 2 kênh hoặc mã hoá hình chéo cũng có thể chọn mức lượng tử thích nghi. Số bit gán cho mỗi pixel cũng có thể chọn một cách thích nghi với mọi bộ mã hoá dạng sóng mà chúng ta vừa nghiên cứu. Ở những vùng mà tín hiệu lượng tử hoá biến thiên chậm có thể cho số bit/pixel ít. Cũng có thể giữ số bit/khung cố định, nhưng cho tỷ lệ bit biến thiên theo cường độ pixel. Trong mã hoá thích nghi các thông số của bộ mã hoá được thích nghi với đặc trưng nào đấy chẳng hạn độ tương phản của ảnh ở từng vùng. Độ đo tại chỗ này có thể nhận được từ cường độ của những pixel đã mã hoá, nhưng không phải truyền nó đi.

Nếu độ đo tại chỗ lấy thẳng từ  $f(n_1, n_2)$  thì phải truyền nó đi bởi vì bên máy thu không tiếp cận được với ảnh gốc. Mã hoá thích nghi làm cho thiết bị mã hoá phức tạp hơn nhưng cải thiện chất lượng một cách đáng kể.



**Hình 4.40:** Ví dụ của ảnh được mã hoá véctơ.

- (a) ảnh gốc 512x512 pixel.
- (b) ảnh mã hoá bởi lượng tử hoá véctơ tại tỷ lệ  $\frac{1}{2}$  bít/pixel kích thước khối được sử dụng là 4x4 pixel và sách mã được thiết kế bằng cách sử dụng sự biến thiên của thuật toán K – means. NMSE = 2,7%, SNR = 15,7dB.

Hệ PCM không khai thác sự phụ thuộc thống kê giữa cường độ các pixel lân cận nhau. Có một cách để khai thác sự phụ thuộc thống kê là dùng DM, DPCM và bộ mã hoá hai kênh, khi mà người ta mã hoá hiệu số giữa  $f(n_1, n_2)$  và một giá trị dự báo của  $f(n_1, n_2)$ . Một cách khác là dùng lượng tử hoá vectơ. Như đã thảo luận ở tiết 2 lượng tử hoá vectơ có thể khai thác sự phụ thuộc thống kê giữa các thông số được mã hoá. Lượng tử hoá vectơ được xét đến khi mã hoá dạng sóng  $f(n_1, n_2)$ . Các khối được dùng gồm cường độ các pixel lân cận nhau và có kích thước nhỏ 2 x 2, 3 x 3, 4 x 4.

Lượng tử hoá vectơ thoạt tiên được dùng ở những nơi tỷ lệ bit thấp (dưới 1 bit/pixel) bởi vì kích thước khối càng lớn tỷ lệ bit càng cao thì chi phí tính toán và lưu trữ càng tăng vọt lên. Muốn ảnh có độ dẽ hiểu tốt (tuy có hi sinh một phần chất lượng) được phục hồi bằng tỷ lệ bit thấp không thể sử dụng DM, DPCM hoặc bộ mã hoá hai

kênh với lượng tử hoá vô hướng và từ mã có chiều dài không đổi. Chỉ dùng lượng tử hoá vectơ mới có thể thực hiện được với tỷ lệ bit thấp hơn 1 bit/pixel. Hình 4.40 biểu diễn ảnh được lượng tử hoá vectơ. Hình 4.40a là ảnh gốc 512 x 512 pixel. Hình 4.40b là kết quả lượng tử hoá vectơ ở tỷ lệ nửa bit/pixel. Kích thước khối là 4 x 4 pixel. Sách mã thiết kế theo algorit K\_means. Dữ liệu luyện tập được dùng là các khối của 4 ảnh khác nhau

#### **4. PHÉP MÃ HOÁ BIẾN ỔI ẢNH.**

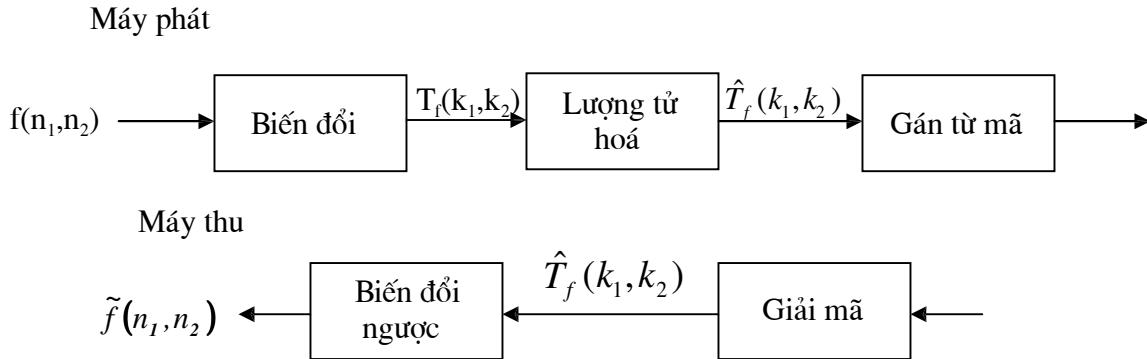
Trong phép mã hoá biến đổi ảnh, ảnh được biến đổi sang một lĩnh vực khác với lĩnh vực cường độ và người ta mã hoá các hệ số biến đổi. Ở những ứng dụng mà tỷ lệ bít thấp (dưới 1 đến 2 bit/pixel) như hội nghị video thì mã hoá hệ số biến đổi với lượng tử hoá vô hướng cho kết quả tốt hơn các phép biến đổi dạng sóng với mã hoá vô hướng nhưng chi phí tính toán đắt hơn.

Các kỹ thuật mã hoá hệ số biến đổi làm giảm sự tương quan giữa các cường độ pixel triệt để hơn các phương pháp mã hoá dạng sóng. Khi làm giảm sự tương quan thì không còn có hiện tượng mã hoá lặp đi lặp lại những thông tin thừa. Kỹ thuật mã hoá hệ số biến đổi khai thác được đặc điểm là ở một số ảnh điển hình phần lớn năng lượng tập trung vào 1 phần nhỏ của hệ số biến đổi. Đó là tính chất của (compact) năng lượng. Nhờ tính chất đó có thể mã hoá 1 số nhỏ hệ số biến đổi mà vẫn không ảnh hưởng nhiều đến ảnh, nghĩa là có thể mã hoá với tỷ lệ dưới 1 bit/pixel mà không ảnh hưởng nhiều đến chất lượng ảnh và độ đẽ hiếu.

Khi mã hoá các hệ số biến đổi có thể dùng bất kỳ phương pháp lượng tử hoá nào đã nói đến ở tiết 1 nhưng thông dụng nhất vẫn là 1 lượng tử hoá vô hướng, bởi vì nó đơn giản.

##### **4.1. Các phép biến đổi.**

Sơ đồ của một bộ mã hoá hệ số biến đổi trên hình 4.41. Ở máy phát ảnh  $f(n_1, n_2)$  được biến đổi và các hệ số biến đổi  $T_f(k_1, k_2)$  được lượng tử hoá, sau đó các hệ số đã lượng tử hoá  $\hat{T}_f(k_1, k_2)$  được mã hoá. Bên máy thu các từ mã được giải mã các hệ số biến đổi  $\hat{T}_f(k_1, k_2)$  được biến đổi ngược và ta nhận được ảnh phục hồi  $\hat{f}(n_1, n_2)$ .



**Hình 4.41.** Bộ mã hoá biến đổi ảnh.

Ta mong muốn một số tính chất của phép biến đổi có một số thuộc tính như sau: Trước hết là vì ở cả đầu phát và đầu thu đều phải thực hiện phép biến đổi ( thuận và ngược ) cho nên cần phải có những phương thức hữu hiệu để tính toán nó.

Các phép biến đổi cho mã hoá ảnh là biến đổi tuy ẩn tính và có thể biểu diễn như sau:

$$T_f(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) a(n_1, n_2; k_1, k_2) \quad (4.56a)$$

$$f(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} T_f(k_1, k_2) b(n_1, n_2; k_1, k_2) \quad (4.56b)$$

Trong đó  $f(n_1, n_2)$  là một chuỗi  $N_1 \times N_2$  điểm,  $T_f(k_1, k_2)$  cũng là một chuỗi  $N_1 \times N_2$  điểm và đại biểu cho các hệ số biến đổi  $a(n_1, n_2; k_1, k_2)$  và  $b(n_1, n_2; k_1, k_2)$  là những hàm cơ bản thoả mãn phương trình (4.56).

Từ (4.56) ta có thể suy ra rằng  $f(n_1, n_2)$  là một tổ hợp tuyến tính của các hàm cơ bản  $b(n_1, n_2; k_1, k_2)$ , và các hệ số biến đổi  $T_f(k_1, k_2)$  là những biên độ của những hàm cơ bản trong tổ hợp tuyến tính.

Khi các hàm cơ bản có một dạng nào đó có tính chất hình sin thì các hệ số biến đổi có thể giải thích như biên độ của phổ mở rộng. Xét về mặt tính toán các hàm cơ bản dùng trong mã hoá biến đổi ảnh điều đó có thể tách rời cho nên có thể biểu diễn (4.56) dưới dạng sau đây:

$$T_f(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) a_R(n_1, k_1) a_C(n_2, k_2) \quad (4.57a)$$

$$f(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} T_f(k_1, k_2) b_R(n_1, k_1) b_C(n_2, k_2) \quad (4.57b)$$

Một ví dụ của phép biến đổi theo dạng (4.57) là biến đổi Fourier rời rạc DFT:

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) e^{-j(2\pi/N_1)k_1 n_1} e^{-j(2\pi/N_2)k_2 n_2} \quad (4.58a)$$

$$f(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F(k_1, k_2) e^{j(2\pi/N_1)k_1 n_1} e^{j(2\pi/N_2)k_2 n_2} \quad (4.58b)$$

Khi các hàm cơ bản tách rời được, phép biến đổi thuận và phép biến đổi ngược có thể tính toán được bằng cách phân tích thành cột\_dòng.

Sự phân tích cột\_dòng có thể làm giảm số phép tính số học. Trong trường hợp DFT tinh 512x512 điểm thì phân tích cột\_dòng làm giảm số phép tính số học được mấy trăm lần. Với những phép đổi như DFT còn có thể khai thác tính chất hình sin của các hàm cơ bản để làm giảm khối lượng tính toán hơn nữa.

Một đặc tính nữa yêu cầu đối với phép biến đổi là giảm sự tương quan giữa các hệ số biến đổi. Đó là thuộc tính giảm tương quan. Chọn thích hợp các hàm số cơ bản  $b(n_1, n_2; k_1, k_2)$  có thể đạt mục đích này. Một thuộc tính mong muốn khác có liên quan với giảm tương quan là tính chặt (compact) năng lượng. Tập trung năng lượng vào một số ít hệ số biến đổi cho phép bỏ qua nhiều hệ số biến đổi mà không ảnh hưởng đáng kể đến ảnh, sự giảm tương quan có góp phần cho tính compact năng lượng nhưng không phải là điều kiện đủ để có compact năng lượng. Chẳng hạn trong tập âm trắng biên độ này hoàn toàn không liên quan với biên độ khác nhưng trong tập âm trắng không có compact năng lượng.

Một hệ số biến đổi có ít năng lượng thì chỉ đóng góp không đáng kể vào năng lượng của các tín hiệu. Đó là tính bảo tồn độ đo năng lượng trong lĩnh vực biến đổi. DFT có thuộc tính đó. Định lý Parseval nói rằng:

$$\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} |f(n_1, n_2)|^2 = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} |F(k_1, k_2)|^2 \quad (4.59)$$

Theo công thức này thì hệ số DFT với biên độ nhỏ chỉ đóng góp phần nhỏ năng lượng vào tín hiệu. Như vậy nếu bỏ qua các hệ số DFT có biên độ bé thì không làm méo tín hiệu một cách đáng kể. Mọi biến đổi tuyến tính dùng cho mã hoá ảnh đều có tính chất phù hợp phương trình (4.59). Có thể chứng minh rằng thuộc tính bảo tồn năng lượng có trong tất cả các phép biến đổi Unitary. Người ta đã nghiên cứu nhiều phương pháp biến đổi trong mã hoá ảnh. Chúng chỉ khác nhau về compact năng lượng và yêu cầu tính toán. Phép biến đổi KL (Karhunen\_Loeve) là phép biến đổi tuyến tính có compact năng lượng cao nhất. Trong phép biến đổi KL các hàm cơ bản  $a(n_1, n_2; k_1, k_2)$  nhận được bằng cách giải phương trình.

$$\lambda(k_1, k_2) a(n_1, n_2; k_1, k_2) = \sum_{l_1=0}^{N_1-1} \sum_{l_2=0}^{N_2-1} K_f(n_1, n_2; l_1, l_2) a(l_1, l_2; k_1, k_2) \quad (4.60)$$

Trong  $\lambda(k_1, k_2)$ :

$$K_f(n_1, n_2; l_1, l_2) = E[(f(n_1, n_2) - E[f(n_1, n_2)])(f(l_1, l_2) - E[f(l_1, l_2)])] \quad (4.61)$$

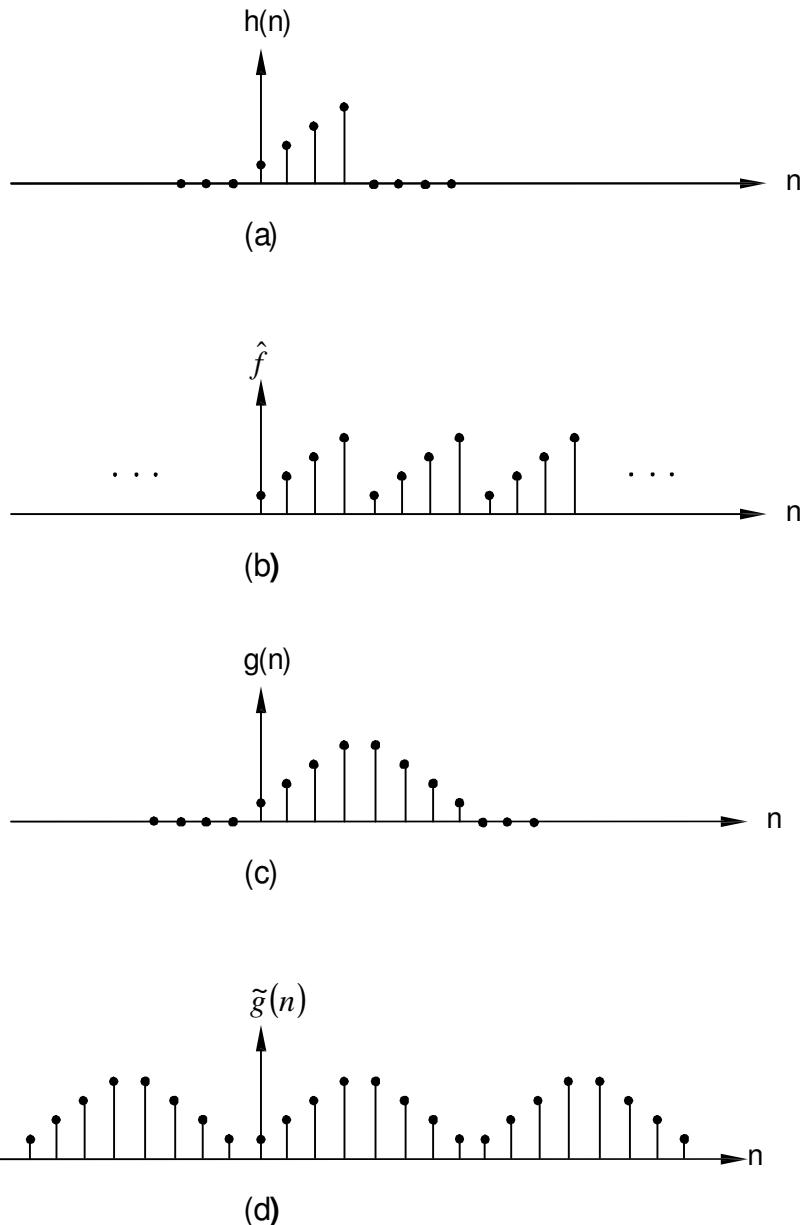
Trong phím trinh (4.60) giả thi Q róng  $f(n_1, n_2)$  lμ mét qu, trinh ngEu nhanh cã hμm hiQ biO k<sub>f</sub>(n<sub>1</sub>, n<sub>2</sub>; l<sub>1</sub>, l<sub>2</sub>) biQ. Trong phim vi cña giả thi Q nuy c, c hÖsè biO KL cã thuéc tÝh compact n" ng l- i ng cao nhE.

C, c thuéc tÝh biO KL rE hay vÒmAEIý thuyQ nh- ng cã nh- ng khä kh"n nhanh cã m"t träng trong thuc tQ v×vEY rE. Ý d" ng trong m- ho, Pnh.

ChØcä phEp biO DFT lμ cã m"t b"p c"p c"p, cã Algorit tÝh to, n rE hiQ quP vµ cã tÝh compat n" ng l- i ng tét. Cã thÓd" ng phEp biO Fourier nhanh FFT biO DFT vµ biO c"p ng- i c. DFT cã vai trB quan träng trong xö lý tÝh hiQ sè.

Cã thÓcP thiO tham tÝh compact n" ng l- i ng cña DFT m"t kh"ng phP hi sinh c, c - u kh, c lμ d" ng phEp biO c"p sin rei r"c DCT. §Óminh ho1 m"t quan hÖ gi"a DCT vµ DFT còng nh- minh ho1 sù tEpr trung n" ng l- i ng trong DCT cao h-n trong

DFT ta dī ng hñh 4.42. H̄ y x̄ d̄ y N. Trong khung gian 1 chīu ë hñh 4.42a. Ở  
 N= 4\_ C, c hÖsè F(k) cña DFT IÊy tõ nh÷ng hÖsè  $\tilde{F}(k)$  ví i  $0 \leq k \leq N-1$  cña chuçi Fourier r̄i r̄i DFS. C, c hÖsè DFSIÊy tõ chuçi  $\tilde{f}(n)$  trong hñh 4.42b.



**Hình 4.42.** Giả sử  $\Omega$  là một tập hợp compact nằm trong  $\mathbb{R}^n$  và  $\varphi$  là một hàm số liên tục có giá trị không âm trên  $\Omega$ . Khi đó,  $\varphi$  là một hàm Fourier của một hàm số liên tục có giá trị không âm trên  $\mathbb{R}^n$ .

Xem hinh 4.42b thay  $\tilde{f}(n)$  ca nhung mât nhai t<sup>1</sup>o râ n<sup>1</sup>, do vi Örem nèi ph<sup>1</sup>ng  $\tilde{f}(n)$  ví i ph<sup>1</sup>ng cuèi cña f(n). Nh<sup>1</sup>ng sù mât li<sup>1</sup>n t<sup>1</sup>c n<sup>1</sup>g găp n<sup>1</sup>ng l-<sup>1</sup> ng cho c<sub>1</sub> c<sub>2</sub> t<sup>1</sup>o cao v<sup>1</sup>l<sup>1</sup>m gi<sup>1</sup>m t<sup>1</sup>h compact n<sup>1</sup>ng l-<sup>1</sup> ng cña DFT. § Ölo<sup>1</sup>i trô nh<sup>1</sup>ng mât li<sup>1</sup>n t<sup>1</sup>c nhai t<sup>1</sup>o n<sup>1</sup>g t<sup>1</sup>o f(n) ta t<sup>1</sup>o ra 1 chuçi 2N  $\tilde{f}(n)$  nh- tr<sup>1</sup>n hinh 4.42c. Chuçi g(n)  $\tilde{f}(n)$  xong ví i  $\tilde{f}(n)$  ch<sup>1</sup>h gi<sup>1</sup>a t<sup>1</sup>o n = N-1  $\tilde{f}(n)$  = N v<sup>1</sup>l<sup>1</sup>N  $\tilde{f}(n)$  ti<sup>1</sup>n cña g(n) ho<sup>1</sup>n to<sup>1</sup>n gèng f(n). B<sup>1</sup>y gi<sup>1</sup>e n<sup>1</sup>u ta t<sup>1</sup>h 2N  $\tilde{f}(n)$  DFT cña g(n), c<sub>1</sub> c<sub>2</sub> gi<sup>1</sup>, tr<sup>1</sup>DFT G(k) còng gi<sup>1</sup>ng nh- c<sub>1</sub> c<sub>2</sub> h<sup>1</sup>s<sup>1</sup>  $\hat{G}(k)$  cña DFSI. Thay t<sup>1</sup>o chuçi  $\hat{g}(n)$  tr<sup>1</sup>n hinh 4.42d. C<sub>1</sub> c<sub>2</sub> h<sup>1</sup>ös<sup>1</sup> DCT x<sub>1</sub> c<sub>2</sub> ph<sup>1</sup>ng theo G(k).

Râ r<sup>1</sup>ng l<sup>1</sup>mu chuçi  $\hat{g}(n)$  kh<sup>1</sup>ng c<sub>1</sub> nh<sup>1</sup>ng mât li<sup>1</sup>n t<sup>1</sup>c nhai t<sup>1</sup>o v<sup>1</sup>l<sup>1</sup>m compact n<sup>1</sup>ng l-<sup>1</sup> ng c<sub>1</sub> c<sub>2</sub> h<sup>1</sup>ös<sup>1</sup> DCT hi<sup>1</sup>u qu<sup>1</sup> h<sup>1</sup>-n c<sub>1</sub> c<sub>2</sub> h<sup>1</sup>ös<sup>1</sup> DFT.

Ngo<sup>1</sup>p<sup>1</sup> c<sub>1</sub> c<sub>2</sub> ph-<sup>1</sup>ng ph<sub>1</sub>p DFT v<sup>1</sup>l<sup>1</sup>DCT c<sub>1</sub>b<sub>1</sub>n c<sub>1</sub> c<sub>2</sub> ph<sup>1</sup>p bi<sup>1</sup>nh  $\tilde{f}(n)$  kh<sub>1</sub>c nh- Haar, Hadamard, nh<sup>1</sup>ng DCT v<sup>1</sup>l<sup>1</sup>n c<sub>1</sub> t<sup>1</sup>h n<sup>1</sup>ng l-<sup>1</sup> ng cao v<sup>1</sup>l<sup>1</sup>t<sup>1</sup>h to<sub>1</sub>n d<sup>1</sup>Or<sup>1</sup>ng. Vi<sup>1</sup>Öt<sup>1</sup>h c<sub>1</sub> c<sub>2</sub> ph<sup>1</sup>p bi<sup>1</sup>nh Haar v<sup>1</sup>l<sup>1</sup> Hadamard y<sup>1</sup>u c<sub>1</sub>  $\tilde{f}(n)$  ph<sup>1</sup>p t<sup>1</sup>h h<sup>1</sup>-n nh<sup>1</sup>ng t<sup>1</sup>h compact n<sup>1</sup>ng l-<sup>1</sup> ng thua DCT, v<sup>1</sup>xv<sup>1</sup> t<sup>1</sup>o  $\tilde{f}(n)$  v<sup>1</sup>l<sup>1</sup>sau s<sup>1</sup>u th<sup>1</sup>o lu<sup>1</sup>nh t<sup>1</sup>EP chung v<sup>1</sup>l<sup>1</sup>DCT.

#### 4.2 Nh<sup>1</sup>ng kh<sub>1</sub>i ni<sup>1</sup>Öm bæsung v<sup>1</sup>l<sup>1</sup> v<sup>1</sup>Ydô minh ho<sup>1</sup> m<sup>1</sup> ho<sup>1</sup>, t<sup>1</sup>ong <sup>1</sup>nh con .

Ph<sup>1</sup>p m<sup>1</sup> ho<sup>1</sup>, t<sup>1</sup>ong <sup>1</sup>nh con, mét <sup>1</sup>nh  $\tilde{f}(n)$   $\tilde{f}(n)$  i c chia th<sup>1</sup>nh nhai <sup>1</sup>nh con hay kh<sup>1</sup>i, m<sup>1</sup>c<sub>1</sub> kh<sup>1</sup>i  $\tilde{f}(n)$  i c m<sup>1</sup> ho<sup>1</sup>, ri<sup>1</sup>ng r<sup>1</sup>. B<sup>1</sup>ng c<sub>1</sub> ch m<sup>1</sup>c<sub>1</sub> l<sup>1</sup> m<sup>1</sup> ho<sup>1</sup>, 1 <sup>1</sup>nh con c<sub>1</sub> thÓl<sup>1</sup>m b<sup>1</sup> m<sup>1</sup> ho<sup>1</sup>, th<sup>1</sup>ch nghi ví i  $\tilde{f}(n)$  tr-<sup>1</sup>ng t<sup>1</sup>i ch<sup>1</sup>c cña <sup>1</sup>nh. V<sup>1</sup>Ydô ph-<sup>1</sup>ng ph<sub>1</sub>p ch<sup>1</sup>nh m<sup>1</sup>c<sub>1</sub> l-<sup>1</sup> ng t<sup>1</sup>o v<sup>1</sup>l<sup>1</sup> g<sub>1</sub>n bit <sup>1</sup> v<sup>1</sup>ng <sup>1</sup>nh c<sub>1</sub> n<sup>1</sup>u v<sup>1</sup>l<sup>1</sup> v<sup>1</sup>ng bi<sup>1</sup>nh c<sub>1</sub> thÓkh<sub>1</sub>c nhau. Vi<sup>1</sup>Öt<sup>1</sup> m<sup>1</sup> ho<sup>1</sup>, t<sup>1</sup>ong <sup>1</sup>nh con s<sup>1</sup>l<sup>1</sup>m gi<sup>1</sup>m kh<sup>1</sup>i l-<sup>1</sup> ng t<sup>1</sup>h to<sub>1</sub>n v<sup>1</sup>l<sup>1</sup> l- u tr<sup>1</sup>, b<sup>1</sup>i v<sup>1</sup>xm<sup>1</sup>c<sub>1</sub> l<sup>1</sup>ch<sup>1</sup>ox<sup>1</sup> l<sup>1</sup>y 1 <sup>1</sup>nh con th<sup>1</sup>xkh<sup>1</sup>ng c<sub>1</sub> l- u tr<sup>1</sup> c<sub>1</sub> <sup>1</sup>nh.  $\tilde{f}(n)$  c<sub>1</sub> kh<sup>1</sup>i ni<sup>1</sup>Öm v<sup>1</sup>l<sup>1</sup>sau gi<sup>1</sup>m b<sup>1</sup> t<sup>1</sup> kh<sup>1</sup>i l-<sup>1</sup> ng t<sup>1</sup>h to<sub>1</sub>n ta h<sup>1</sup>y x<sup>1</sup> 1 <sup>1</sup>nh f(n<sub>1</sub>,n<sub>2</sub>) c<sub>1</sub>

NxN pixel, trong  $\tilde{f}(n)$  l<sup>1</sup>u<sup>1</sup> mét lu<sup>1</sup>th<sup>1</sup>nh cña 2. Ta<sup>1</sup>em chia f(n<sub>1</sub>,n<sub>2</sub>) th<sup>1</sup>nh nhai <sup>1</sup>nh con. K<sup>1</sup>ch th-<sup>1</sup> c m<sup>1</sup>c<sub>1</sub> <sup>1</sup>nh con l<sup>1</sup>u<sup>1</sup> MxM pixel, trong  $\tilde{f}(n)$  l<sup>1</sup>u<sup>1</sup> M còng l<sup>1</sup>u<sup>1</sup> mét lu<sup>1</sup>th<sup>1</sup>nh cña 2, nh- v<sup>1</sup>l<sup>1</sup>s<sup>1</sup> <sup>1</sup>nh con trong <sup>1</sup>nh l<sup>1</sup>u<sup>1</sup> N<sup>2</sup>/M<sup>2</sup>, n<sup>1</sup>u s<sup>1</sup>e ph<sup>1</sup>p t<sup>1</sup>h s<sup>1</sup>e h<sup>1</sup>ac khi t<sup>1</sup>h 1 ph<sup>1</sup>p bi<sup>1</sup>nh MxM  $\tilde{f}(n)$  l<sup>1</sup>u<sup>1</sup> M<sup>2</sup>log<sub>2</sub>M<sup>2</sup> th<sup>1</sup>x tang s<sup>1</sup>e ph<sup>1</sup>p to<sub>1</sub>n s<sup>1</sup>e h<sup>1</sup>ac ph<sup>1</sup>p t<sup>1</sup>h  $\tilde{f}(n)$   $\tilde{f}(n)$  b<sup>1</sup>c<sub>1</sub> <sup>1</sup>nh l<sup>1</sup>u<sup>1</sup> (N<sup>2</sup>/M<sup>2</sup>) M<sup>2</sup>log<sub>2</sub>M<sup>2</sup> = N<sup>2</sup>log<sub>2</sub>M<sup>2</sup>.

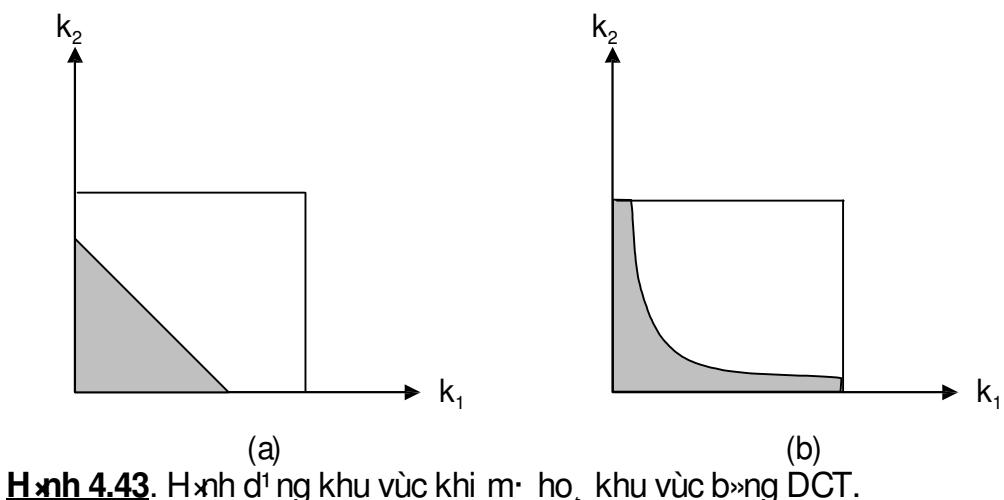
Khi t<sup>1</sup>h 1 b<sup>1</sup>c<sub>1</sub> <sup>1</sup>nh NxN  $\tilde{f}(n)$  th<sup>1</sup>x c<sub>1</sub> N<sup>2</sup>log<sub>2</sub>N<sup>2</sup> ph<sup>1</sup>p t<sup>1</sup>h s<sup>1</sup>e h<sup>1</sup>ac. Khi N = 512 v<sup>1</sup>l<sup>1</sup> M = 8 th<sup>1</sup>x s<sup>1</sup>u t<sup>1</sup>h to<sub>1</sub>n to<sup>1</sup>n b<sup>1</sup>e <sup>1</sup>nh con ch<sup>1</sup>ob<sup>1</sup>ng 1/3 kh<sup>1</sup>i l-<sup>1</sup> ng t<sup>1</sup>h to<sub>1</sub>n to<sup>1</sup>n <sup>1</sup>nh l<sup>1</sup>u<sup>1</sup>. H<sup>1</sup>-n n<sup>1</sup>a n<sup>1</sup>u c<sub>1</sub> b<sup>1</sup>e  $\tilde{f}(n)$  x<sup>1</sup> l<sup>1</sup>y th<sup>1</sup>xph<sup>1</sup>p bi<sup>1</sup>nh c<sub>1</sub> c<sub>2</sub> <sup>1</sup>nh con c<sub>1</sub> thÓt<sup>1</sup>nh h<sup>1</sup>nh song song. Tuy vi<sup>1</sup>Öt<sup>1</sup> chia <sup>1</sup>nh l<sup>1</sup>u<sup>1</sup> th<sup>1</sup>nh nhai <sup>1</sup>nh con l<sup>1</sup>u<sup>1</sup> gi<sup>1</sup>m b<sup>1</sup> t<sup>1</sup> kh<sup>1</sup>i l-<sup>1</sup> ng t<sup>1</sup>h to<sub>1</sub>n v<sup>1</sup>l<sup>1</sup> cho ph<sup>1</sup>p t<sup>1</sup>o ra b<sup>1</sup>e m<sup>1</sup> ho<sup>1</sup>, th<sup>1</sup>ch nghi ví i  $\tilde{f}(n)$  t<sup>1</sup>h t<sup>1</sup>i ch<sup>1</sup>c cña <sup>1</sup>nh nh- ng kh<sup>1</sup>ng thÓthu nhá k<sup>1</sup>ch th-<sup>1</sup> c <sup>1</sup>nh con  $\tilde{f}(n)$  v<sup>1</sup>l<sup>1</sup> t<sup>1</sup>En. Khi chia b<sup>1</sup>c<sub>1</sub> <sup>1</sup>nh ra nhai <sup>1</sup>nh m<sup>1</sup>ng nhá h<sup>1</sup>-n, ph<sup>1</sup>p bi<sup>1</sup>nh  $\tilde{f}(n)$

khai th<sub>c</sub>  $\oplus \hat{c}$  c<sub>Y</sub> h-n t<sub>Y</sub>h t-<sub>ng</sub> quan gi-<sub>a</sub> c<sub>c</sub> c-<sub>ê</sub>ng d<sub>e</sub> pixel. K<sub>Y</sub>h th-<sub>i</sub> c<sub>nh</sub> con cung nhá s<sub>u</sub> t-<sub>ng</sub> quan gi-<sub>a</sub> c<sub>c</sub> c<sub>nh</sub> con l<sub>o</sub> c<sub>ñ</sub> cung t<sub>ng</sub>. Tuy v<sub>E</sub>y v<sub>x</sub>m<sub>c</sub> c<sub>nh</sub> con  $\oplus \hat{c}$  m-<sub>ho</sub>,  $\oplus$ c l<sub>E</sub>p cho n<sub>a</sub>n t<sub>Y</sub>h t-<sub>ng</sub> quan gi-<sub>a</sub> c<sub>c</sub> c<sub>nh</sub> con l<sub>o</sub> c<sub>ñ</sub> kh<sub><</sub>ng  $\oplus \hat{c}$  khai th<sub>c</sub>. Nh-<sub>v</sub>Ey l<sub>u</sub>c<sub>a</sub> m<sub>t</sub> gi<sub>i</sub> i h<sub>1</sub>n  $\oplus$ i ví i k<sub>Y</sub>h th-<sub>i</sub> c<sub>nh</sub> con. L- u ý r<sub>r</sub>ng khi m-<sub>ho</sub>, ph<sub>D</sub>p bi<sub>O</sub>n  $\oplus$ e k<sub>Y</sub>h th-<sub>i</sub> c<sub>nh</sub> con l<sub>i</sub> n h-n khi l-<sub>i</sub> ng t<sub>o</sub> ho, vect<sub>r</sub>. K<sub>Y</sub>h th-<sub>i</sub> c<sub>kh</sub>e<sub>i</sub> d<sub>i</sub> ng trong l-<sub>i</sub> ng t<sub>o</sub> ho, vect<sub>r</sub>-th<sub><</sub>ng th-<sub>ê</sub>ng l<sub>u</sub> 2 x 2, 3 x 3 ho<sub>A</sub>E 4 x 4, do s<sub>u</sub> ph<sub>o</sub> thu<sub>c</sub> theo h<sub>um</sub> m<sub>o</sub> c<sub>n</sub>a nh-<sub>ng</sub> y<sub>a</sub>u c<sub>Q</sub>u v<sub>O</sub>t<sub>Y</sub>h to<sub>s</sub> n v<sub>u</sub>b<sub>e</sub> nhí do k<sub>Y</sub>h c<sub>i</sub> kh<sub>e</sub>i  $\oplus$ Era. V<sub>O</sub>m<sub>A</sub>E<sub>O</sub>, s<sub>u</sub> l-<sub>i</sub> ng t<sub>o</sub> ho, v<sub><</sub> h-<sub>i</sub> ng c<sub>c</sub> c<sub>h</sub>Ös<sub>e</sub> bi<sub>O</sub>n  $\oplus$ e c<sub>a</sub> - u th<sub>O</sub>v<sub>O</sub>khai th<sub>c</sub> t<sub>Y</sub>h k<sub>Q</sub> h<sub>i</sub> p hi<sub>O</sub>n h<sub>u</sub> gi-<sub>a</sub> c<sub>c</sub> c-<sub>ê</sub>ng  $\oplus$ e pixel t<sub>et</sub> h-n l-<sub>i</sub> ng t<sub>o</sub> ho, v<sub>D</sub>c<sub>r</sub>-d<sub>i</sub> ng s<sub>a</sub>ng

### MÃ HOÁ KHU VỰC VÀ MÃ HOÁ NGƯỜNG

M-<sub>ho</sub>, ph<sub>D</sub>p bi<sub>O</sub>n  $\oplus$ e c<sub>nh</sub> khai th<sub>c</sub> t<sub>Y</sub>h compact n<sub>u</sub>ng l-<sub>i</sub> ng. Ch<sub>O</sub>mét ph<sub>Q</sub>nh nhá c<sub>n</sub>a c<sub>c</sub> c<sub>h</sub>Ös<sub>e</sub> bi<sub>O</sub>n  $\oplus$ e  $\oplus \hat{c}$  m-<sub>ho</sub>. C<sub>a</sub> hai c<sub>c</sub> ch ti<sub>Q</sub> c<sub>ñ</sub>  $\oplus$ Óquy<sub>Q</sub>  $\oplus$ ph n<sub>a</sub>n m-<sub>ho</sub>, ph<sub>D</sub>p bi<sub>O</sub>n  $\oplus$ e n<sub>u</sub>o l<sub>u</sub>m-<sub>ho</sub>, khu v<sub>u</sub>c v<sub>u</sub>m-<sub>ho</sub>, ng-<sub>i</sub> ng. Khi m-<sub>ho</sub>, khu v<sub>u</sub>c ch<sub>O</sub>m-<sub>ho</sub>, c<sub>c</sub> c<sub>h</sub>Ös<sub>e</sub> trong m<sub>t</sub> khu v<sub>u</sub>c nh<sub>E</sub>  $\oplus$ nh. C<sub>a</sub> nh<sub>Q</sub>u y<sub>Q</sub>u t<sub>e</sub> c<sub>nh</sub> h-<sub>ê</sub>ng  $\oplus$ h<sub>n</sub>h d<sub>i</sub> ng v<sub>u</sub>k<sub>Y</sub>h th-<sub>i</sub> c<sub>kh</sub>e<sub>i</sub> khu v<sub>u</sub>c, ch<sub>1</sub>ag h<sub>1</sub>n ph<sub>D</sub>p bi<sub>O</sub>n  $\oplus$ e  $\oplus \hat{c}$  d<sub>i</sub> ng v<sub>u</sub>s<sub>e</sub> bit  $\oplus \hat{c}$  c<sub>ph</sub>D<sub>p</sub> d<sub>i</sub> ng.

C<sub>c</sub> c<sub>h</sub>nh th<sub>c</sub> m-<sub>ho</sub>, khu v<sub>u</sub>c th-<sub>ê</sub>ng d<sub>i</sub> ng  $\oplus$  DCT ch<sub>a</sub>n<sub>h</sub>nh d<sub>i</sub> ng c<sub>n</sub>a khu v<sub>u</sub>c nh-<sub>tr</sub><sub>a</sub>n h<sub>n</sub>h 4.43. Ch<sub>O</sub>nh-<sub>ng</sub> h<sub>Ö</sub>s<sub>e</sub> n<sub>u</sub>m trong v<sub>i</sub> ng g<sup>1</sup>ch h<sub>D</sub>  $\oplus \hat{c}$  m-<sub>ho</sub>, c<sub>B</sub>n nh-<sub>ng</sub> h<sub>Ö</sub>s<sub>e</sub> n<sub>u</sub>m ngo<sub>u</sub> v<sub>i</sub> ng n<sub>u</sub>y coi nh-<sub>b</sub>ng 0. Nh-<sub>ng</sub> h<sub>n</sub>h d<sub>i</sub> ng khu v<sub>u</sub>c n<sub>u</sub>y l<sub>u</sub> ch<sub>a</sub>n<sub>h</sub> tr<sub>a</sub>n c<sub>ñ</sub> s<sub>e</sub> c<sub>n</sub>a nh<sub>E</sub>n x<sub>E</sub> v<sub>O</sub>s<sub>u</sub> t<sub>E</sub>p trung  $\oplus$ a ph<sub>Q</sub> n<sub>u</sub>ng l-<sub>i</sub> ng t<sub>1</sub>i khu v<sub>u</sub>c th<sub>E</sub>p. Trong m-<sub>ho</sub>, ng-<sub>i</sub> ng ch<sub>O</sub>nh-<sub>ng</sub> h<sub>Ö</sub>s<sub>e</sub> bi<sub>O</sub>n  $\oplus$ e cao h-n m<sub>c</sub>ng-<sub>i</sub> ng l<sub>u</sub>  $\oplus \hat{c}$  m-<sub>ho</sub>. X<sub>E</sub> v<sub>O</sub>quan  $\oplus$ Ón t<sub>Y</sub>h compact c<sub>n</sub>a n<sub>u</sub>ng l-<sub>i</sub> ng th<sub>x</sub>m-<sub>ho</sub>, ng-<sub>i</sub> ng t<sub>et</sub> h-n m-<sub>ho</sub>, khu v<sub>u</sub>c. Trong m-<sub>ho</sub>, khu v<sub>u</sub>c m<sub>t</sub> v<sub>u</sub>i h<sub>Ö</sub>s<sub>e</sub> bi<sub>O</sub>n  $\oplus$ e c<sub>a</sub>bi<sub>a</sub>n  $\oplus$ e th<sub>E</sub>p  $\oplus \hat{c}$  m-<sub>ho</sub>, v<sub>u</sub>m<sub>t</sub> v<sub>u</sub>i h<sub>Ö</sub>s<sub>e</sub> c<sub>a</sub>bi<sub>a</sub>n  $\oplus$ e l<sub>i</sub> n l<sub>1</sub>i b<sub>P</sub>b<sub>a</sub>  $\oplus$ . Trong m-<sub>ho</sub>, ng-<sub>i</sub> ng ch<sub>O</sub>nh-<sub>ng</sub> h<sub>Ö</sub>s<sub>e</sub> bi<sub>O</sub>n  $\oplus$ e c<sub>a</sub>bi<sub>a</sub>n  $\oplus$ e l<sub>i</sub> n l<sub>u</sub>  $\oplus \hat{c}$  ch<sub>a</sub>n. Tuy v<sub>E</sub>y trong m-<sub>ho</sub>, ng-<sub>i</sub> ng v<sub>P</sub>tr<sub>Y</sub>c<sub>n</sub>a ph<sub>D</sub>p bi<sub>O</sub>n  $\oplus$ e  $\oplus \hat{c}$  m-<sub>ho</sub>, kh<sub><</sub>ng bi<sub>Q</sub> tr-<sub>i</sub> c, do  $\oplus$ a nh-<sub>ng</sub> th<sub><</sub>ng tin v<sub>O</sub>v<sub>P</sub>tr<sub>Y</sub>ph<sub>Q</sub>  $\oplus \hat{c}$  m-<sub>ho</sub>, k<sub>i</sub>m theo.



M· ho, ng-ì ng lµ mét ph-¬ng ph, p thÝch nghi. ViÖc chän hÖsè biÖn ®æ nµo ®ö biÖn ho, tuú theo ®Æ tÝnh t¹i chç cña ¶nh. Cí ng mét mœc ng-ì ng sè l-î ng hÖsè ® i c m· ho, è ¶nh con nµy vµ ¶nh con l®i cËn kh«ng gièng nhau.

NÔi ta muèn ph, t triÓn mét hÖcä tû lÖbit kh«ng ®æ, trong ®ä sè bit cho tông ¶nh con gièng nhau thxph¶i cä mét c¬ cËu ®Ùi khiÓn ®Óthay ®æ mœc ng-ì ng è tông ¶nh con vµ ®Ø ®Ùi chØh sè bit g, n cho c, c ¶nh con kh, c nhau còng nh- cho c, c hÖsè kh, c nhau trong mçi ¶nh con.

### Việc phân phối bit.

Cho mét sè bit nhÊt ®ph ®Óm· ho, mét ¶nh con ta ph¶i chia sè bit ®ä cho c, c hÖ sè biÖn ®æ ® i c m· ho, n»m trong ¶nh con. Nh- ® nãi ®Ön è tiÖ 1 nh÷ng hÖsè cä ph-¬ng sai lÍn thxnan ® i c ph®i phèi nhiÙu bit. Gi, trP- í c l-î ng cña ph-¬ng sai cho tông hÖsè biÖn ®æ trong 1 ¶nh con biÖn ®éng kh, nhiÙu tõ hÖsè nµy sang hÖsè kh. Ví i hÖsè DCT kú väng ph-¬ng sai è c, c hÖsè øng ví i tÇng thÊp lÍn h-n è c, c hÖsè øng ví i tÇng cao. Mét vÝdô vÒph®i phèi bit cho c, c hÖsè DCT cä kÝh 16x16 vµ tû lÖ bit 1/2 bit/pixel ® i c biÓu diÔn trªn hñh 4.44. Con sè n»m trong mçi « vu«ng lµ sè bi t ph®i phèi cho tông hÖsè khi m· ho, khu vực.

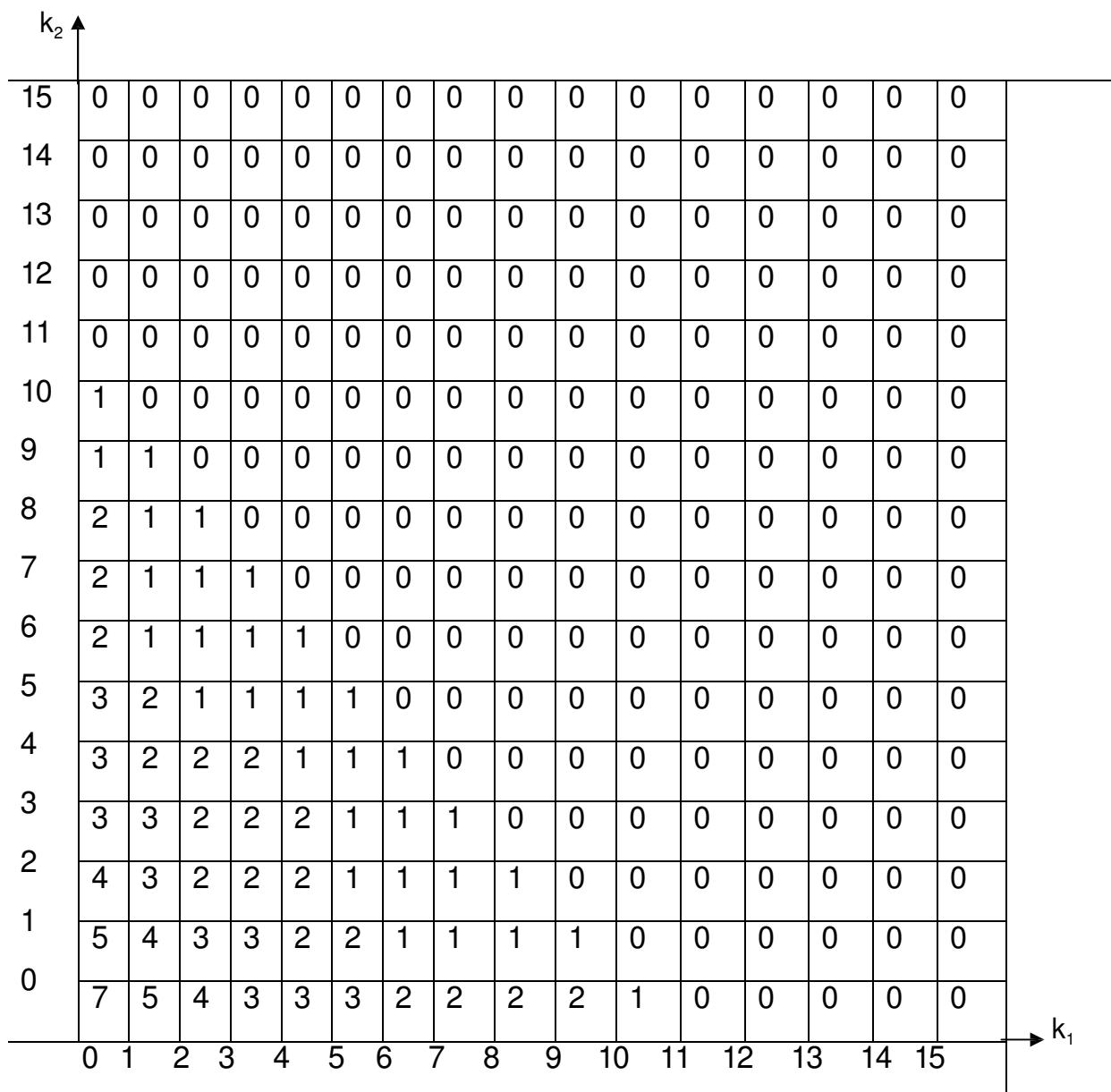
ViÖc chän c, c mœc l-î ng tö trong b- í c l-î ng tö ho, phô thuéc sè bit ® i c ph®i phèi vµ - í c l-î ng ph-¬ng sai cña hÖsè. § Ólo¹i bá sù phô thuéc vµo - í c l-î ng ph-¬ng sai mçi hÖsè ® i c chuËn ho, theo ®é lÖch chuËn vµ nh- vÝy ví i mçi sè l-î ng bit chØcä mét bé mœc l-î ng tö t-¬ng øng.

### Sự xuống cấp do mã hóa phép biến đổi.

Trong m· ho, phÊp biÖn ®æ t¹p @n l-î ng tö cä biÓu hiÖn kh, c ví i tr-êng hî p m· ho, d¹ng säng. Nãi chung ¶nh h-ëng cña t¹p @n l-î ng tö trong m· ho, phÊp biÖn ®æ ®Ý thÖhiÖn tÊp trung h-n. Sù l-î ng tö ho, 1 hÖsè biÖn ®æ ¶nh h-ëng ®Ön mäi c-êng ®é ¶nh b n trong ¶nh con.

Cä nhiÙu hñh thøc xuèng cÊp ®Ón hñh do t¹p @n l-î ng tö g®y ra khi m· ho, phÊp biÖn ®æ. Mét trong nh÷ng hñh thøc ®ä lµ gi¶m sù ®é ph®i biÖt trong kh«ng gian. Khi m· ho, DCT c, c hÖsè øng ví i tÇn cao bÞbá qua. KÕ qu¶ lµ ¶nh bÞgi¶m chi tiÖ. Mét vÝdô ® i c biÓu diÔn trªn hñh 4.45. Hñh 4.45a lµ ¶nh gèc 512x512 pixel, hñh

4.45b vµ 4.45c lµ nh¬ng ¶nh ® i c phôc h¬i khi gi÷ l¹i 1 4% vµ 8% hÖsè DCT trong t¬ng ¶nh con. KÝch th- í c ¶nh con lµ 16x16 pixel. C, c hÖsè biÕn ®i gi÷ l¹i kh¬ng ® i c l- i ng t¬ ho, vµ ® i c ch¬n t¬ m¬t khu v¬c tam gi, c nh- tr¬n h¬nh 4.43a. Xem h¬nh 4.45 th¬ng gi÷ l¹i cung Y hÖsè thx ¶nh cung m¬. Nh- ng cÙng th¬ng chØc¶n gi÷ l¹i m¬t sè l- i ng nhá hÖsè biÕn ®i lµ ® cã b¬c ¶nh kh, dÔcoi, chøng t¬ cã hiÕn t- i ng compact cña n¬ng l- i ng.



**H¬nh 4.44.** VÝdô vÒph®i ph¬i bit ®i t¬ 1/2 bit/pixel cho c, c hÖsè biÕn ®i cosin r¬i c khi m- ho, ¶nh con kÝch th- í c 16 x 16 pixel b»ng biÕn ®i DCT.



**Hình 4.45 :** Ví dụ về ảnh bị mất dữ liệu qua màng lọc DCT:

- a) Ảnh gốc 512x512 pixel
- b) Ảnh khi chia gián đoạn 14% màng lọc DCT
- c) Ảnh khi chia gián đoạn 8% màng lọc DCT

Màng lọc IFFT khử nhiễu DCT có thể làm giảm lượng thông tin hình ảnh. Số lượng thông tin hình ảnh giảm bao nhiêu phụ thuộc vào tần số cắt bỏ. Hình 4.46 là kết quả khử nhiễu bằng cách loại bỏ 2 bit cho mỗi màng lọc DCT. Hình ảnh sau khi khử nhiễu có thể thấy trong Hình 4.45b.

Hình thức xử lý khử nhiễu bằng màng lọc DCT có thể được thực hiện theo ba bước: (1) Tách khung ảnh thành các khung 16x16 pixel; (2) Áp dụng màng lọc DCT cho khung ảnh; (3) Lấy lại khung ảnh sau khi đã khử nhiễu. Ta sẽ giải thích cách khai thác khung ảnh 16x16 pixel, từ đó ta có thể xác định số lượng thông tin hình ảnh cần loại bỏ.



**Hình 4.46:** Minh ho<sup>1</sup> v<sup>o</sup>s<sup>u</sup> t<sup>o</sup>ng c<sup>,c</sup> h<sup>1</sup>t l<sup>u</sup>n t<sup>o</sup>n tr<sup>a</sup>n <sup>T</sup>nh khi l-<sup>i</sup> ng tö ho<sup>,</sup> <sup>Q</sup>u 2 bit/pixel cho m<sup>c</sup>i h<sup>o</sup>s<sup>e</sup> DCT <sup>R</sup> i c gi<sup>+</sup> l<sup>1</sup>i <sup>R</sup>o cho <sup>T</sup>nh h<sup>x</sup>nh 4.45b.

C<sup>,c</sup> th<sup>Y</sup>dô khi thi<sup>Q</sup> k<sup>O</sup>mét b<sup>e</sup> m<sup>c</sup> ho<sup>,</sup> bi<sup>Ö</sup>n <sup>R</sup> e m<sup>e</sup>t t<sup>u</sup> l<sup>Ö</sup>bit <sup>R</sup> cho th<sup>x</sup>ph<sup>T</sup> ch<sup>a</sup>n c<sup>,c</sup> th<sup><</sup>ng s<sup>e</sup> thi<sup>Q</sup> k<sup>O</sup>th<sup>Y</sup>h h<sup>i</sup>p <sup>R</sup>o<sup>c</sup> <sup>R</sup>i nh<sup>u</sup>ng s<sup>u</sup> xu<sup>è</sup>ng c<sup>Ê</sup>p do l-<sup>i</sup> ng tö ho<sup>,</sup>. Nh<sup>u</sup>ng th<sup><</sup>ng s<sup>e</sup> <sup>R</sup>a l<sup>u</sup>p ph<sup>E</sup>p bi<sup>Ö</sup>n <sup>R</sup>e d<sup>i</sup>ng <sup>R</sup> i c, k<sup>Y</sup>h th-<sup>i</sup> c <sup>T</sup>nh con, c<sup>,c</sup> ch ch<sup>a</sup>n nh<sup>u</sup>ng h<sup>o</sup>s<sup>e</sup> <sup>R</sup> i c gi<sup>+</sup> l<sup>1</sup>i, c<sup>,c</sup> ch ph<sup>o</sup>p ph<sup>e</sup>i bit, c<sup>,c</sup> ch ch<sup>a</sup>n m<sup>e</sup>c l-<sup>i</sup> ng tö. N<sup>Q</sup>u m<sup>e</sup>t h<sup>x</sup>nh th<sup>e</sup>c xu<sup>è</sup>ng c<sup>Ê</sup>p n<sup>u</sup>o tr<sup>Q</sup>m tr<sup>a</sup>ng h<sup>-n</sup> c<sup>,c</sup> h<sup>x</sup>nh th<sup>e</sup>c kh<sup>,c</sup> th<sup>x</sup>ph<sup>T</sup> thay <sup>R</sup> e th<sup><</sup>ng s<sup>e</sup> c<sup>n</sup>a m<sup>c</sup> ho<sup>,</sup> <sup>R</sup>o<sup>c</sup><sup>T</sup> thi<sup>a</sup>n s<sup>u</sup> xu<sup>è</sup>ng c<sup>Ê</sup>p n<sup>u</sup>y v<sup>u</sup>p ch<sup>Ê</sup>p nh<sup>Ë</sup>n s<sup>u</sup> xu<sup>è</sup>ng c<sup>Ê</sup>p kh<sup>,c</sup>. H<sup>x</sup>nh 4.48 cho c<sup>,c</sup> c v<sup>Y</sup>dô v<sup>o</sup>m<sup>c</sup> ho<sup>,</sup> ph<sup>E</sup>p bi<sup>Ö</sup>n <sup>R</sup>e. C<sup>,c</sup> h<sup>x</sup>nh 4.48a v<sup>u</sup> 4.48b cho k<sup>Q</sup> qu<sup>¶</sup> m<sup>c</sup> ho<sup>,</sup> DCT <sup>e</sup> t<sup>u</sup> l<sup>Ö</sup>1 bit/pixel v<sup>u</sup> 1/2 bit/pixel.

Ảnh g<sup>e</sup>c l<sup>u</sup> 512 x 512 pixel nh<sup>-</sup> tr<sup>a</sup>n h<sup>x</sup>nh 4.45a. Trong c<sup>¶</sup> hai tr<sup>-</sup>êng h<sup>i</sup>p <sup>T</sup>nh con l<sup>u</sup> 16 x 16 pixel v<sup>u</sup>p d<sup>i</sup>ng m<sup>c</sup> ho<sup>,</sup> khu v<sup>u</sup>c v<sup>i</sup> i h<sup>x</sup>nh d<sup>i</sup>ng khu v<sup>u</sup>c nh<sup>-</sup> tr<sup>a</sup>n h<sup>x</sup>nh 4.43b, k<sup>Y</sup>h th-<sup>i</sup> c khu v<sup>u</sup>c th<sup>Y</sup>h nghi v<sup>i</sup> i <sup>R</sup>A <sup>R</sup>Ón t<sup>1</sup>i ch<sup>c</sup> c<sup>n</sup>a <sup>T</sup>nh.

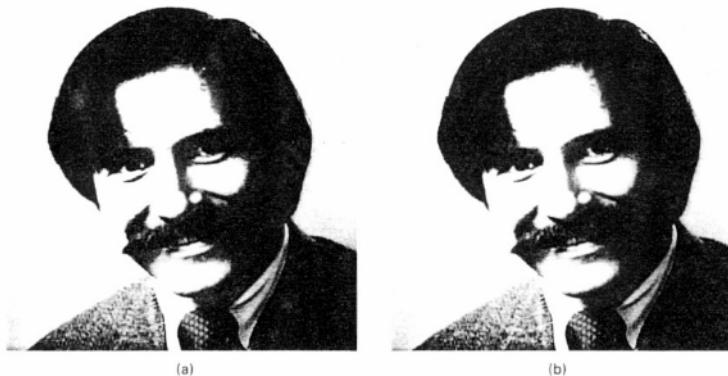


**Hình 4.47:** <sup>A</sup>nh m<sup>c</sup> ho<sup>,</sup> DCT c<sup>a</sup> hi<sup>Ö</sup>øng kh<sup>e</sup>i bi<sup>Ö</sup>øi hi<sup>Ö</sup>øi râ n<sup>D</sup>.

### 4.3. Sù lüm gi¶m hiÖu øng khèi.

Khi tû lÖbit ®i thÊp, hiÖu øng khèi, kÖ qu¶ cña sù m· ho, ®éc lËp c,c ¶nh con, trë n¤n rÊt râ nD. C,c ¶nh phôc hải bPhiÖu øng khèi s rÊt khä coi, vµ hi Öu øng khèi th-êng lµ sù xuèng cËp trQn træng nhÊt trong c,c kiÖu xuèng cËp.

Cã hai c, ch tiÖp cËn ®Ólüm gi¶m hiÖu øng khèi. ¢ c, ch thø nhÊt, xö lý hiÖu øng khèi ngay t¹i nguân. Mét vÝdô cña c, ch tiÖp cËn nøy lµ ph-¬ng ph,p phñ bê, tœc lµ thay ®éc qu, trænh ph®i m¶ng ¶nh. C, ch ph®i m¶ng ¶nh ®Ó hñh lµ ®em chia ¶nh thunh nh-¬ng vï ng kh«ng cä miÒi giao nhau. Trong ph-¬ng ph,p phñ bê, mci ¶nh con ®Ó cä chu vi g¢i l¤n c,c ¶nh con l@i cËn mét Ý. C,c pixel tr¤n ®êng chu vi ® i c m· ho, tr¤n hai hay nhiÖu vï ng. Khi phôc hải ¶nh, pixel nµo ® i c m· ho, tr¤n mét lQn s ® i c nhÊn mét c-êng ®é lµ bñh qu®i cña c,c gi, trpm· ho,. Nh- vËy nh-¬ng bÊt li¤n tœc ®éc ngét t¹i bi¤n do m· ho, g®y ra s ® i c lüm tr-n bí t nhê c,c ¶nh con ® phôc hải ® i c “s i ®an” l¹i ví i nhau. Hñh 4.49 cho vÝdô vØph-¬ng ph,p phñ bê.



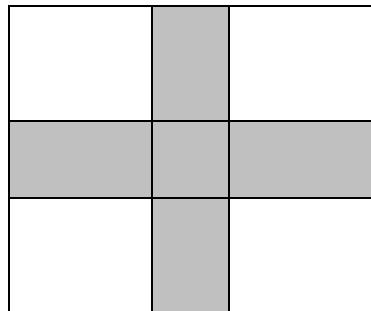
**Hñh 4.48.** VÝdô m· ho, ¶nh b»ng DCT.

a) M· ho, DCT ¢ tû lÖ1 bit/pixel. b)

M· ho, DCT ¢ tû lÖ1/2 bit/pixel

Tr¤n hñh mét ¶nh 5 x 5 pixel ® i c chia thunh 4 ¶nh con 3 x 3 (cho phñ bê 1 pixel). Pixel trong khu vực v¹ch chĐo ® i c m· ho, tr¤n mét lQn. Ph-¬ng ph,p phñ bê cä t,c dÔng khö hiÖu øng khèi rÊt tét. Tuy vËy vxcä mét sè pixel ® i c m· ho, tr¤n mét lQn cho n¤n ®Ó lµm t“ng sè pixel ph¶i m· ho,. Khi mét ¶nh 256 x 256 pixel ® i c chia

thịnh nhèng ảnh con  $16 \times 16$  pixel thay đổi ng phẳng phẳng bê lumen tông 13% sè pixel phẳng mờ hoà. Khi phôc hải ảnh, gián phẳng trả cho viền phẳng bê c, c ảnh con lumen tông khèi lật ng bit tách to, n, cản khi mờ hoà, ảnh thay đổi ng phẳng phẳng bê khung nhèng lumen tông khèi lật ng tách to, n mực cản cã xu thõi lumen tông tû lõi bit cõi cã.



**Hình 4.49.** Výdô vòphẳng bê mét pixel khi

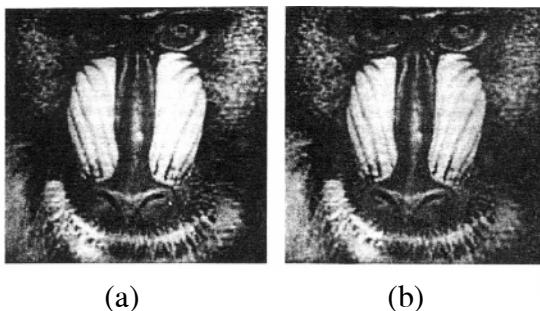
đổi ng phẳng phẳng bê  $\otimes$   
giảm hiết ọng khèi.

Mét výdô khac vòc, ch tiáp cõi chung bụi to, n giảm hiết ọng khèi ngay tui  
nguân lumen dãy ng c, c phẳng biõi  $\otimes$  khac nhau. Trong mét loai biõi  $\otimes$  truc gèi nhau gãi lõi  
lapped orthogonal transform, c, c ảnh con  $\otimes$  i c phẳng bê. Nhưng tang sè hõsè biõi  $\otimes$  vén  
nh- sè pixel trang ảnh. § i òi  $\otimes$  thuc hiết b»ng c, c h biõi diõi ảnh con ví i hõsè biõi  
 $\otimes$  ý h-n kých th- í c ảnh con. Tuy khung thôphôc hải  $\otimes$ ng ảnh con b»ng c, c hõsè biõi  
 $\otimes$  cña ảnh con, nh- ng lõi cã thôphôc hải  $\otimes$ ng hoan toan ảnh gèc b»ng tõi c, c hõsè  
biõi  $\otimes$  cña c, c ảnh con gép lõi. Vì vậy phẳng bê c, c ảnh con lumen giảm hiết ọng khèi.  
Khac ví i phẳng phẳng DCT, trong phẳng phẳng biõi  $\otimes$  truc gèi nhau (lapped  
transform) sè hõsè biõi  $\otimes$  cõi cã  $\otimes$ phôc hải ảnh chay x, c cùng b»ng kých th- í c  
ảnh.

Một cách tiếp khac để khắc phục hiệu ứng khói là đem lọc ảnh ê bé giả mờ, sau  
khi giảm ọng khèi  $\otimes$  xay ra. Trong c, ch tiáp cõi nay, quan trinh mờ hoà khung  $\otimes$ .  
Chay h-n c, c ảnh con hoan toan khung gèi lõi nhau. Bé lắc lumen kiết thung thõi v $\times$   
nhèng mõi lõi tõi ê vĩ ng biõi do phan mõng gõi ra còng gièng nhèng mõi dèc  $\otimes$ ,  
cã thịnh phan tõi sè cao. H-n n-ia bé lắc chay p dôngvuo vĩ ng biõi n-ia c, c ảnh con  $\otimes$   
trinh lumen mõi ảnh. Või bé giả mờ biõi quan trinh phan mõng dĩ ng ê bé mờ, cho n-ia

bí Q<sub>Đ</sub> i c vPtRÝc, c bÊt li<sup>ä</sup>n tÔc nh<sub>Đ</sub> t<sup>1</sup>o. K<sub>h</sub>, c ví i ph- -ng ph, p phñ bê ph- -ng ph, p l<sub>ä</sub>c kh<sub>é</sub>ng l<sub>µ</sub>m t<sup>°</sup>ng sè bit R<sub>Đ</sub> i c m- ho<sub>Đ</sub>. Ng- êi ta R<sub>Đ</sub> nghi<sup>ä</sup>n c<sub>ó</sub>u th<sub>É</sub>y r<sub>»</sub>ng ë mét tû lÖbit n<sub>µ</sub>o R<sub>Đ</sub>y thxph- -ng ph, p cã hi<sub>Ö</sub>u qu<sub>¶</sub> chèng hi<sub>Ö</sub>u øng khèi tèt h-n ph- -ng ph, p phñ bê d<sub>i</sub> ng DCT.

Hình 4.50 cho mét vÝdô vÒviÖc d<sub>i</sub> ng ph- -ng ph, p l<sub>ä</sub>c R<sub>Đ</sub>Ogi¶m hi<sub>Ö</sub>u øng khèi. Hình 4.50a l<sub>µ</sub> ¶nh cã hi<sub>Ö</sub>u øng khèi râ n<sub>µ</sub>t, c<sub>Ó</sub>n h<sub>ñ</sub>nh 4.50b l<sub>µ</sub> ¶nh R<sub>Đ</sub> R<sub>Đ</sub> i c x<sub>ö</sub> lý b<sub>»</sub>ng c<sub>ç</sub> ch x<sub>ö</sub> d<sub>ö</sub>ng b<sub>é</sub> l<sub>ä</sub>c th<sub>é</sub>ng th<sub>é</sub>p FIR 3 x 3 R<sub>Đ</sub>Øn, v<sub>µ</sub>t, c d<sub>ö</sub>ng l<sub>ä</sub>c ch<sub>Ø</sub>t, c R<sub>Đ</sub>eng v<sub>µ</sub>o nh<sub>é</sub>ng pixel tại vùng biên của các ảnh con.



**Hình 4.50.** Ví dụ về giảm hiệu ứng khói bằng phương pháp lọc :

- a) Ánh 512 x 512 pixel với hiệu ứng khói rõ nét. Ánh này mã hoá bằng bộ mã hoá DCT khu vực ở tỷ lệ 0,2 bit/pixel.
- b) Ánh (a) sau khi qua lọc để giảm hiệu ứng khói.

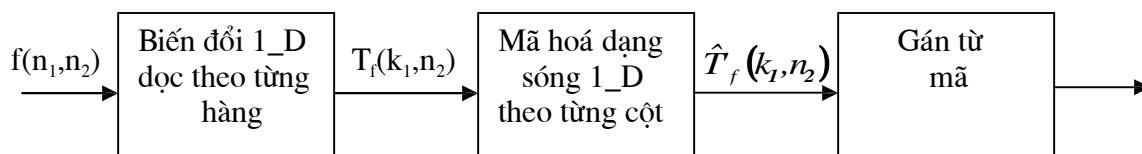
#### 4.4. Sù m- ho<sub>Đ</sub> biÖn R<sub>Đ</sub>ai lai ghEp.

Mã hoá phép biến đổi có hiệu quả tốt khi tỷ lệ bit thấp, còn mã hoá dạng sóng thì lại có ưu điểm đơn giản. Nếu ta lai cả hai phương pháp này để hình thành một bộ mã hoá lai kết hợp cả mã hoá dạng sóng và mã hoá phép biến đổi làm cho thiết bị đơn giản mà việc mã hoá ở tỷ lệ bit thấp cũng tốt hơn khi mã hoá dạng sóng.

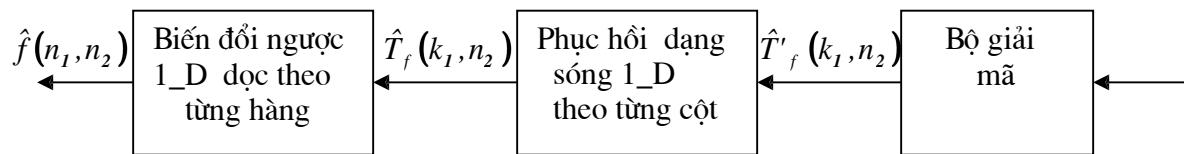
Trong bộ mã hoá lai, ảnh  $f(n_1, n_2)$  được biến đổi bằng phép biến đổi trong không gian một chiều, chẳng hạn 1\_D DCT, đọc theo một dòng hay một cột. Các hệ số  $T_f(k_1, n_2)$  sau đó được mã hoá dạng sóng, chẳng hạn bằng DPCM đọc theo một cột hay dòng. Điều này được minh họa trên hình 4.51. Phép biến đổi 1\_D làm mất tính tương

quan giữa các dòng. Sự tương quan còn lại bị DPCM khử nốt. Nhờ có phép biến đổi cho nên sự tương quan giữa các số liệu được giảm đi nhiều hơn so với trường hợp dùng bộ mã hoá dạng sóng đơn thuần. Bởi vì dùng biến đổi 1-D cho nên những thủ thuật như chọn hình dáng và kích thước khu vực trong mã hoá đơn giản hơn trường hợp mã hoá 2-D. Phép mã hoá lai trong thực tế ít dùng cho mã hoá ảnh với một khung hình, có thể là vì phương pháp này không làm giảm sự kết hợp trong các dữ liệu nhiều bằng bộ mã hoá biến đổi 2-D, và khi đưa bộ biến đổi 2-D vào cũng không phức tạp hơn bộ biến đổi lai ghép. Nhưng phép biến đổi lai ghép lại có ích khi mã hoá ảnh liên mành, như sẽ trình bày ở tiết 4.6.

Máy phát



Máy thu



**Hình 4.51.** Bộ mã hoá lai (biến đổi / dạng sóng).

#### 4.5. M· ho· thÝch nghi vµ l· i ng tö ho· vect·

Những kỹ thuật mã hoá phép biến đổi có thể làm cho thích nghi với những đặc tính tại chỗ của các ảnh con. Chẳng hạn trong mã hoá khu vực hình dạng và kích thước khu vực có thể được thích nghi. Ngoài ra số bit phân phối cho mỗi ảnh con và cho mỗi hệ số trong ảnh con đều có thể thích nghi. Chẳng hạn ta có thể mã hóa vùng ảnh có nền đều với ít hệ số và mỗi hệ số được phân ít bit hơn là ở vùng ảnh có nhiều biến đổi.

Phương pháp mã hoá có thể thích nghi liên tục bằng cách dựa vào một vài độ đo, chẳng hạn vào phu·ng sai của các cường độ pixel trong một ảnh con. Nếu có t hể lấy được giá trị độ đo từ các ảnh con đã mã hoá trước thì không cần truyền số liệu về độ đo đi. Một cách khác để tiến hành thích nghi liên tục là sắp xếp các ảnh con thành từng nhóm nhỏ và thiết kế những phương pháp mã hoá riêng cho từng nhóm. Ưu điểm của phương pháp này là với mỗi nhóm có chứa một ảnh con đã cho thì chỉ cần một số bit nhỏ để mã hoá. Nếu số bit/khung trong phép mã hoá thích nghi là cố định thì chỉ cần có

một cơ cấu điều khiển để chia cho mỗi ảnh con số bít thích hợp. Với bộ mã hoá có tỷ lệ bit biến thiên thì phải có một tầng đệm để thích ứng những sự biến thiên về tỷ lệ bit tại chỗ.

Mã hoá thích nghi cải thiện chất lượng của bộ mã hoá phép biến đổi nhưng cũng làm nó phức tạp thêm. Mã hoá phép biến đổi được dùng rộng rãi trong những ứng dụng mà tỷ lệ bit thấp. Nhiều bộ mã hoá biến đổi trong thực tế đều có tính thích nghi.

Trong mã hoá phép biến đổi cường độ các pixel trong một khối được biến đổi cả gói. Đứng về mặt nào đó mà nói thì mã hoá phép biến đổi giống như lượng tử hoá vectơ của dạng sóng. Cũng có thể dùng lượng tử hoá vectơ cho các hệ số biến đổi nhưng sự cải thiện không bù đắp được những sự phức tạp gây ra. Các hệ số biến đổi không tương quan nhiều với nhau, còn phép lượng tử hoá vectơ lại khai thác chủ yếu tính tương quan giữa các vô hướng.

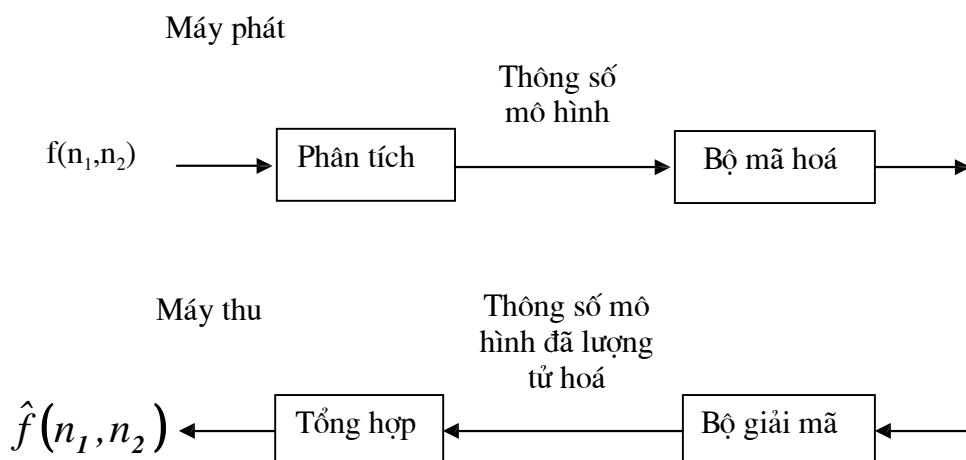
## 5. MÃ HOÁ MÔ HÌNH ẢNH.

Khi mã hoá mô hình ảnh, bức ảnh hoặc một phần của bức ảnh được mô hình hoá và các thông số của mô hình được dùng để tổng hợp ảnh. Ở máy thu nhận được các thông số mô hình đã lượng tử hoá và dùng chúng để tổng hợp lại bức ảnh. Hình 4.52 vẽ một bộ mã hoá mô hình. Về thực chất đây là một hệ phân tích / tổng hợp. Các bộ mã hoá mô hình ảnh có ưu thế là tổng hợp được những ảnh có độ dễ hiểu cao với tỷ lệ bit thấp hơn nhiều so với bộ mã hoá dạng sóng và mã hoá phép biến đổi. Tuy vậy chúng đang ở trong giai đoạn nghiên cứu và cũng còn nhiều việc phải làm trước khi ứng dụng nó vào những việc cần tỷ lệ bit thấp, chẳng hạn máy video telephon cho người điếc. Việc phát triển một hệ mô hình đơn giản có khả năng tổng hợp được ra những ảnh hiểu được không phải là việc dễ dàng. Ngoài ra ước lượng các thông số mô hình và dùng nó tổng hợp lại ảnh là rất tốn kém về mặt tính toán.

Phép mã hoá mô hình ảnh dựa trên khái niệm cho rằng để tổng hợp một ảnh hiểu được không cần phải phục hồi chi tiết các cường độ ảnh. Chẳng hạn những vùng nền của ảnh như là cỏ, trời, bức tường không cần thiết cho độ dễ hiểu của ảnh, cho nên có thể thay chúng bằng những nền tương tự được tổng hợp từ một mô hình đơn giản.

Một ví dụ khác, là người vẽ phim hoạt hình có thể vẽ một bức ảnh có độ dễ hiểu cao chỉ bằng mấy nét đơn giản. Trong mã hoá mô hình ảnh ta giữ lại những chi tiết của

ảnh cần thiết cho độ dẽ hiểu, còn những chi tiết phụ thì chỉ giữ lại những nét gân đúng, có thể tổng hợp bằng những mô hình đơn giản. Cách tiếp cận này ngược hẳn với mã hoá dạng sóng và mã hoá phép biến đổi, bởi vì ở đó ta cố phục hồi thật chính xác cường độ ảnh  $f(n_1, n_2)$ . Trong mã hoá dạng sóng và mã hoá phép biến đổi chênh lệch giữa  $f(n_1, n_2)$  và giá trị phục hồi  $\hat{f}(n_1, n_2)$  là do các thông số bị lượng tử hoá. Nếu không lượng tử hoá các thông số, thì có thể phục hồi chính xác  $f(n_1, n_2)$ . Trong mã hoá mô hình ảnh chênh lệch giữa  $f(n_1, n_2)$  và ảnh tổng hợp  $\hat{f}(n_1, n_2)$  là do cả lượng tử hoá các thông số mô hình lẫn sai số khi lập mô hình. Nói chung không thể phục hồi chính xác  $f(n_1, n_2)$  từ các thông số mô hình, ngay cả khi không tiến hành lượng tử hoá. Trong phép mã hoá mô hình, số thông số tham gia ít hơn bên mã hoá dạng sóng và mã hoá biến đổi, do đó mà mã hoá mô hình có xu thế ứng dụng tốt vào những trường hợp tỷ lệ bit thấp.



**Hình 3.52.** Độ mã hóa mô hình ảnh.

Một bức ảnh gồm nhiều vùng với những đặc tính tại chỗ khác nhau. Những vùng như cỏ, nước, trời, bức tường có cấu trúc lặp đi lặp lại giống như một bức sợi đan. Ta gọi vùng này là vùng cấu trúc sợi đan. Có hai cách tiếp cận việc mô hình hoá vùng cấu trúc sợi đan. Một cách là dùng một hoạ tiết đơn giản rồi lặp lại nó nhiều lần theo một quy luật xác định hoặc ngẫu nhiên. Một cách tiếp cận khác là coi như đây là một trường ngẫu nhiên có những đặc tính thống kê đã cho. Sự nghiên cứu cho thấy rằng khi hai vùng cấu trúc sợi đan có đặc tính thống kê bậc hai giống nhau thì đối với mắt người chúng giống như một. Có nhiều mô hình có thể dùng để mô hình hoá vùng cấu trúc sợi đan như một trường ngẫu nhiên có vài đặc tính thống kê cấp hai đã cho. Hình 4.53 là một ví dụ về vùng cấu trúc sợi đan tổng hợp bằng mô hình quá trình ngẫu nhiên.



**Hình 4.53.** Ảnh gốc 512 x 512 pixel có một vùng  
(ở góc phải, phía trên) gồm 96 x 128  
pixel đã được thay thế bằng cấu trúc  
sợi đan. Mô hình markov bậc 1 đã được  
dùng để tổng hợp cấu trúc sợi đan.

Ảnh gốc 512 x 512 pixel trong đó một vùng 96 x 128 pixel được thay thế bằng cấu trúc sợi đan dùng mô hình markov bậc nhất có ba thông số mô hình chưa biết. Mặc dù trên 10000 cường độ pixel chỉ được tổng hợp bằng ba thông số nhưng vùng cấu trúc sợi đan này cũng trộn rất khớp với toàn bộ phần còn lại của bức ảnh. Để sử dụng cách tiếp cận này người ta đã triển khai các phương pháp phân mảng bức ảnh ra thành những vùng có cấu trúc sợi đan khác nhau. Các bức ảnh có chứa những đồ vật không thể đếm mô hình hoá hoàn bằng các vùng cấu trúc sợi đan. Ở những vùng có ảnh đồ vật phải dùng những phương pháp mô hình hoá khác dựa trên khái niệm là một số đường viền đặt đúng chỗ có thể giữ lại một phần rất lớn độ dễ hiểu về các ảnh đồ vật. Dùng cách tiếp cận này ta có thể biểu diễn một số vùng của bức ảnh bằng những tập đường cong, ngoài ra những vùng còn lại thay bằng những vùng có cấu trúc sợi đan. Một trong những phương pháp để xác định ra đường cong thích hợp là đếm ảnh ánh xạ vào một ảnh 2 mức và quan sát những chỗ xuất hiện sự đổi mức. Một ưu điểm của phương pháp này là có thể dùng ảnh 2 mức để mã hoá. Có nhiều cách để ánh xạ ảnh trên thang độ xám lên ảnh 2 mức. Sau đây là một trong những cách đó.

Ảnh trên thang độ xám có thể ánh xạ vào ảnh 2 mức theo một trật tự của những luật quyết định. Trong phương pháp này những pixel được ánh xạ bởi luật quyết định mức cao, thì luật quyết định mức thấp không kiểm tra đến chúng. Mọi pixel được ánh xạ vào mức đen hay mức trắng tùy theo thời gian chúng nó tới mức quyết định thấp nhất. Ở mức cao nhất cường độ pixel được so sánh với những mức ghim cao và ghim thấp. Trong nhiều ảnh phần lớn vùng mà tin tức ở đường biên có ý nghĩa quan trọng thì cường độ chỉ có giá trị trung bình. Tuỳ theo những mức xén đã được chọn, một phần đáng kể của ảnh hai mức có thể được sinh ra tại mức này. Ở mức quyết định tiếp theo, dùng đến hai ngưỡng động trong đó có xét cả cường độ pixel, hiệu cường độ pixel và độ chói tại chỗ. Một pixel có cường độ cao chỉ được ánh xạ vào mức đen khi nó thấp hơn nhiều so với độ chói trung bình tại chỗ. Tương tự, một pixel có cường độ thấp chỉ được ánh xạ vào mức trắng khi nó cao hơn nhiều so với độ chói trung bình tại chỗ. Ở mức quyết định tiếp theo, một algorit phát hiện đường biên được áp dụng cho các pixel chưa được ánh xạ vào mức trắng hay mức đen.

Dựa trên kết quả của algorit phát hiện đường biên, theo cường độ pixel và mức chênh lệch của nó với độ chói tại chỗ thì pixel có biểu rõ sự tồn tại đường biên được ánh xạ vào mức trắng. Sau giai đoạn này còn pixel nào chưa được ánh xạ thì đó chỉ là những pixel không có giá trị miêu tả. Chúng có thể ánh xạ vào mức trắng hay mức đen theo một cách thích hợp. Chẳng hạn trong một môi trường nhiều mành, chúng có thể ánh xạ vào cùng mức với những pixel ở cùng chỗ trong mành trước. Kết quả là nhận được một ảnh hai mức. Ảnh này còn có thể được tiếp tục gia công bằng một bộ lọc median để loại bỏ một vài điểm thay đổi đột ngột, thường là do tạp âm sinh ra.

Hình 4.54 cho một bức ảnh hai mức nhận được từ ảnh theo thang độ xám có kích thước 512 x 512 pixel.



**Hình 3.54.** ảnh hai mức nhận được từ ảnh theo thang độ xám.

Có thể mã hoá thẳng ảnh hai mức dựa trên khái niệm về mức không thể thay đổi nhanh từ pixel này sang pixel lân cận.

Trên tinh thần mã hoá mô hình ảnh ta mong muốn mô hình các đường viền (nghĩa là chỗ có sự quá độ giữa hai mức) là một tập những phần tử cơ bản như đường thẳng, đa thức, đường tròn, ellip, có thể đại biểu bằng một số thông số ít hơn. Việc lựa chọn phần tử cơ bản và ước lượng thông số của chúng từ các đường viền tuỳ thuộc vào lớp đồ vật mà đường viền biểu diễn, chẳng hạn đường viền biểu diễn mặt người phải khác đường viền biểu diễn cái nhà. Mô hình hoá một loại ảnh nào đó, chẳng hạn mặt người, khi dùng cho những ứng dụng tỷ lệ bit thấp cũng tương tự như mô hình hoá tiếng nói.

Để mô hình hoá tiếng người, đã phát triển một lớp tín hiệu âm thanh, là một hệ tỷ lệ bit rất thấp mà độ dẽ hiểu tương đối khá.

Tất nhiên là tiếng nói rất khác với hình ảnh về phương pháp tạo ra ở nguồn phát âm cũng như về cách tiếp cận của giác quan. Việc mô hình hoá một loại đồ vật riêng nào đó trong một ảnh có thành một hệ thống thực tế hay không là điều còn phải chờ xem.

## **6. MÃ HOÁ ẢNH LIÊN MÀNH, MÃ HOÁ ẢNH MÂU HIỆU ỨNG LỐI KÊNH.**

### **6.1. M· ho· ¶nh li n m nh.**

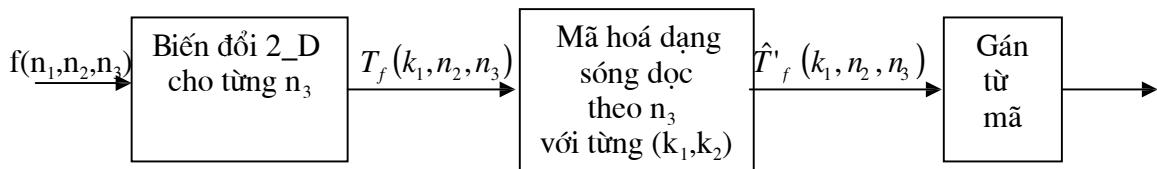
Cho đến nay chỉ mới thảo luận về ảnh một mành, chỉ khai thác tính tương quan không gian và được gọi là mã hoá trong mành (intrafram). Trong những ứng dụng như truyền, ảnh động có cả một chuỗi mành. Nhưng thường từ một mành sang mành tiếp theo ảnh không thay đổi quá nhiều cho nên tính tương quan giữa các mành lân cận cần được khai thác khi mã hoá ảnh. Việc khai thác cả tính tương quan theo không gian để mã hoá một chuỗi ảnh gọi là mã hoá liên mành.

Việc mã hoá liên mành yêu cầu lưu trữ mành trong quá trình mã hoá. Nếu dùng N mành trước để mã hoá mành hiện tại thì phải lưu trữ N mành trước đó. Ngoài ra nhất định sẽ có trễ kèm theo nếu cần cả N mành một lúc để mã hoá mành hiện tại. Điều đó hạn chế số lượng mành dùng trong mã hoá liên mành. Thông thường chỉ dùng một hoặc hai mành trước để mã hoá mành hiện tại.

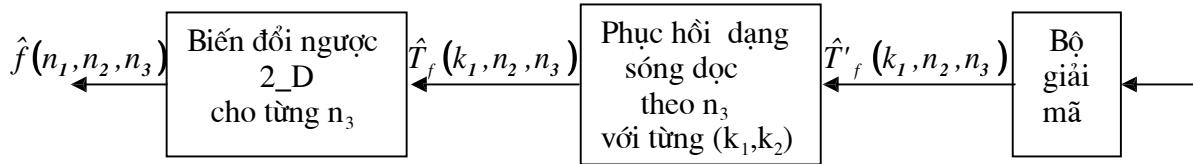
Một cách tiếp cận về mã hoá liên mành là mở rộng mã hoá liên mành 2\_D sang mã hoá liên mành 3\_D. Khi mã hoá dạng sóng bằng DPCM có thể dự báo cường độ pixel hiện tại bằng một tổ hợp tuyến tính của các cường độ pixel lân cận đã mã hoá ở cả

mành hiện tại và mành trước đó. Các phương pháp mã hoá phép biến đổi cũng có thể mở rộng một cách tự nhiên. Sự tương quan mạnh trong lĩnh vực thời gian được biểu hiện n hư một sự tập trung năng lượng ở vùng thời gian\_ tần số thấp, còn các hệ số biến đổi trong vùng thời gian\_ tần số cao có thể bỏ qua mà không làm méo dạng cường độ ảnh trong mành. Sự nghiên cứu cho thấy rằng trong một số trường hợp điển hình có thể giảm tỷ lệ bit 5 lần mà không phải hi sinh độ dẽ hiểu hoặc chất lượng nếu dùng bộ DCT 3\_D với kích thước ảnh con 16 x 16 x 16 thay cho bộ DCT 3\_D với kích thước ảnh con 16 x 16. Trong thực tế yêu cầu về lưu trữ và hiện tượng trễ gây ra làm cho trong nhiều trường h ợp dùng 16 mành đă gấp khó khăn.

Máy phát



Máy thu



Hình 4.55. Bộ mã hóa liên mành lai.

Mã hóa cũng có thể mở rộng cho mã hóa liên mành. Có thể tính biến đổi 2\_D cho mỗi mành và áp dụng mã hóa phép biến đổi như DPCM theo thứ nguyên thời gian (xem hình 4.55). Chuỗi  $f(n_1, n_2, n_3)$  trong hình đại diện cho cường độ ảnh trong các mành,  $n_1$  và  $n_2$  đại diện hai biến không gian và  $n_3$  đại diện biến thời gian. Với mỗi giá trị  $n_3$  thực hiện biến đổi 2\_D với các biến  $n_1$  và  $n_2$  và gọi kết quả là  $T_f(k_1, k_2, n_3)$ . Với mỗi  $k_1, k_2$  dùng bộ mã hóa dạng sóng để lượng tử hóa  $T_f(k_1, k_2, n_3)$ . Kết quả là  $\hat{T}'_f(k_1, k_2, n_3)$ . Ở đầu bộ giải mã thực hiện quá trình ngược. Biến đổi 2\_D thường dùng là DCT còn mã hóa dạng sóng thường dùng DPCM. So với bộ mã hóa phép biến đổi thì mã hóa lai dùng trong mã hóa liên mành có nhiều ưu điểm hơn. Khi chỉ giới hạn dùng một số ít mành như trường hợp

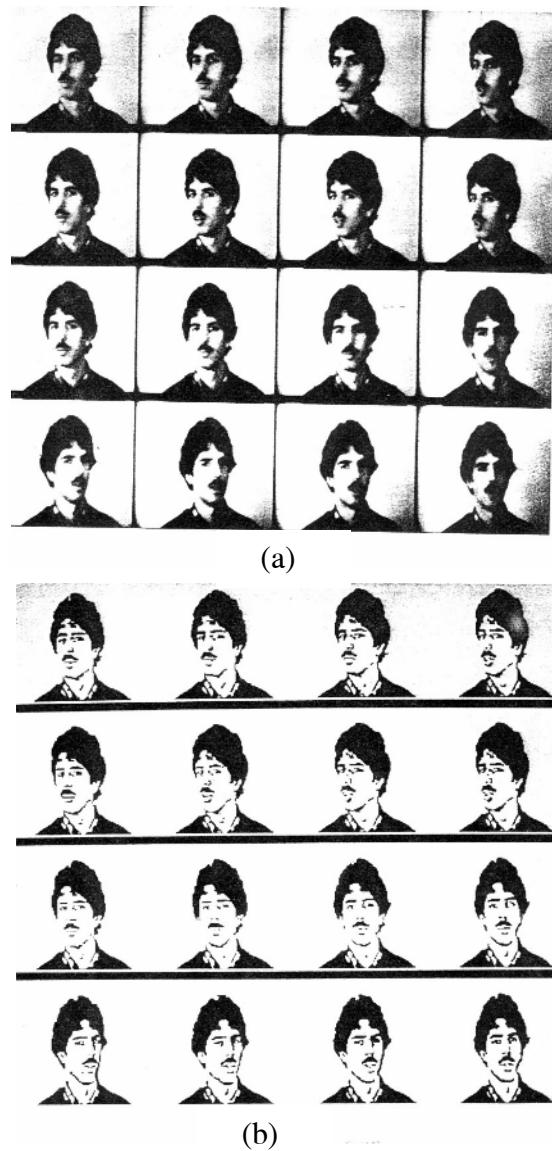
gặp trong thực tế, thì về sự làm giảm tương quan và tính compact năng lượng mã hoá phép biến đổi theo trực thời gian không còn ưu việt gì hơn mã hoá dạng sóng. Khi mã hoá lai, thoát tiên tính biến đổi 2\_D. Vì ở khâu này đã loại đi được nhiều hệ số biến đổi, nên phép mã hoá dạng sóng chỉ còn phải áp dụng cho một số nhỏ hệ số. Nếu dùng mã hoá phép biến đổi thì thoát tiên phải tính hết tất cả các hệ số biến đổi. Ngoài ra mã hoá phép biến đổi đem lại một ít trễ bởi vì khi tính một hệ số cũng phải cần đến nhiều mành. Khi mã hoá lai, mành hiện tại được dự báo bằng một hoặc hai mành đã mã hoá trước và do đó không có trễ đáng kể.

Mã hoá lai liên mành coi như một phép xử lý phụ thuộc thứ nguyên, với mỗi thứ nguyên phải có một phương pháp xử lý riêng phù hợp với đặc tính của thứ nguyên đó. Dọc hai thứ nguyên không gian có nhiều điểm số liệu và phải dùng mã hoá phép biến đổi để khai thác sự tương quan giữa các điểm số liệu. Dọc theo trực thời gian, ít điểm số liệu do đó nên dùng mã hoá dạng sóng.

Phương pháp bổ sung mành liên quan đến DPCM đem mã hoá hiệu giữa mành hiện tại và các mành đã mã hoá trước. Gọi  $f(n_1, n_2, n_3)$  là mành hiện tại và  $\hat{f}(n_1, n_2, n_3 - 1)$  là mành trước đã mã hoá. Trong dạng đơn giản nhất của bổ sung mành,  $f(n_1, n_2, n_3)$  được dự báo bằng  $\hat{f}(n_1, n_2, n_3 - 1)$  và  $e(n_1, n_2, n_3) = f(n_1, n_2, n_3) - \hat{f}(n_1, n_2, n_3 - 1)$  được lượng tử hoá. Vì  $|e(n_1, n_2, n_3)|$  thường rất bé, trừ ở những vùng nhỏ có chuyển động, chỉ những  $e(n_1, n_2, n_3)$  có biên độ vượt quá một mức ngưỡng nào đó mới được mã hoá cùng với vị trí không gian của nó. Ở phía bộ giải mã,  $e(n_1, n_2, n_3)$  đã lượng tử hoá được cộng với  $\hat{f}(n_1, n_2, n_3 - 1)$  để lập ra mành hiện tại  $f(n_1, n_2, n_3)$ . Vì số pixel ở đó  $e(n_1, n_2, n_3)$  được giữ lại phụ thuộc vào mành tại chỗ, phải có một bộ đếm để làm tròn các tỷ lệ số liệu cao hơn trung bình trong những mành có chuyển động mạnh, và các tỷ lệ số liệu thấp hơn trung bình trong những mành có ít chuyển động. Hình 4.56b cho một ví dụ của phương pháp bổ sung mành áp dụng cho cả hai mức. Hình 4.56a cho một dãy 16 mành ảnh gốc. Kích thước mỗi mành là 128 x 128 pixel. Mỗi mành được ánh xạ vào một ảnh hai mức bằng phương pháp ở tiết 5 sau đó đem các ảnh hai mức này mã hoá bằng phương pháp b ổ sung. Kết quả với tỷ lệ bit 0,08 bit/pixel được biểu diễn trên hình 4.56b.

Khi tỷ lệ mành là 15 mành/giây, tốc độ bit tổng là 20 Kbit/giây, ta lưu ý rằng rất khó từ hình 4.56 hình dung ra các mành sẽ hiện ra thế nào khi cho hiện dãy. Chẳng hạn, một sự thăng giáng cường độ từ mành này sang mành kề nó ở cùng một chỗ trên ảnh

không gây ra khó chịu khi nhìn từng mành một. Nhưng đến khi các mành được hiện ra thành một dãy video thì sự thăng giáng nói trên hiện ra thành vết nhảy khó chịu tại một chỗ. Hơn nữa những sự xuống cấp khác như hiện tượng cửa sổ bắn hay cuộn do sự tương quan giữa các mành gây ra, nếu chỉ xem từng mành một thì không nhận thấy.



**Hình 4.56:** Ví dụ về phương pháp bổ sung mành áp dụng cho ảnh hai mức

- (a) Dãy 16 mành gốc, mỗi mành  $128 \times 128$  pixel, chuỗi mành theo thứ tự con số trên hình
- (b) Các mành được phục hồi ở tỷ lệ 0,08 bít/pixel khi áp dụng phương pháp bổ sung mành cho ảnh hai mức nhận được từ ảnh theo thang độ sám trên hình 4.56a

Có một cách để cải thiện tính năng của phương pháp bổ sung mành là dự báo mành hiện tại  $f(n_1, n_2, n_3)$  bằng những algorit ước lượng sự chuyển động. Ta có thể lập ra sai số:

$$e(n_1, n_2, n_3) = f(n_1, n_2, n_3) - \hat{f}(n_1 - dx, n_2 - dy, n_3 - 1)$$

Trong đó  $dx$  và  $dy$  là những dịch chuyển ngang và dọc, là hàm số của vị trí pixel. Trong phạm vi mà sự biến thiên cường độ giữa mành hiện tại và mành trước đó đã mã hoá là do sự dịch chuyển ngang và những dịch chuyển có thể ước lượng chính xác thì có thể nhận được  $e(n_1, n_2, n_3)$  nhỏ hơn bằng cách bù chuyển động. Kết quả là, với sự bù chuyển động như khi dùng phương pháp bổ sung mành, số pixel được mã hoá tại một mức ngưỡng đã cho ít hơn. Vấn đề ước lượng những dịch chuyển (hay tốc độ chuyển động) đã được thảo luận ở tiết

4.2 chương 2. Nếu những dịch chuyển  $d_x$  và  $d_y$  không phải là số nguyên thì thay bằng những số nguyên gần nhất hoặc là phải nội suy không gian  $f^*(n_1, n_2, n_3 - 1)$ .

Nếu ta ước lượng thông số chuyển động từ  $f(n_1, n_2, n_3)$  và  $f^*(n_1, n_2, n_3 - 1)$  lúc mã hoá  $f(n_1, n_2, n_3)$  thì bộ giải mã sẽ không truy cập đến  $f(n_1, n_2, n_3)$  cho nên sẽ phải mã hoá các thông số chuyển động. Có một cách khác là ước lượng thông số chuyển động từ  $f^*(n_1, n_2, n_3 - 1)$  và  $f^*(n_1, n_2, n_3 - 2)$  và giả thiết rằng những thông số chuyển động đó có hiệu lực trong suốt khoảng thời gian từ  $n_3 - 1$  đến  $n_3$ . Trong trường hợp đó, bộ giải mã sẽ truy cập cả đến  $f^*(n_1, n_2, n_3 - 1)$  và  $f^*(n_1, n_2, n_3 - 2)$  và không cần phải mã hoá các thông số chuyển động.

Một phương pháp khác dùng bù chuyển động trong mã hoá liên mành là giảm tốc độ mành bằng cách loại bỏ bớt một số mành rồi sau này đến bộ giải mã sẽ khôi phục lại chúng từ những mành được mã hoá. Việc nội suy mành có bù chuyển động đã được thảo luận ở tiết 4.3 chương 2. Cần lưu ý rằng bỏ bớt nửa số mành chưa chắc đã giảm được tốc độ bit xuống một nửa. Bỏ bớt một số mành có tác dụng làm giảm sự tương quan giữa hai mành kề nhau đem ra mã hoá. Ngoài ra, khi phục hồi vì mành bị bỏ được thay thế bằng một mành mới tạo nên từ hai mành mã hoá, cho nên chất lượng trong một mành sẽ không thể bằng lúc mã hoá không bỏ bớt mành.

## 6.2. M· ho, Tnh mQu.

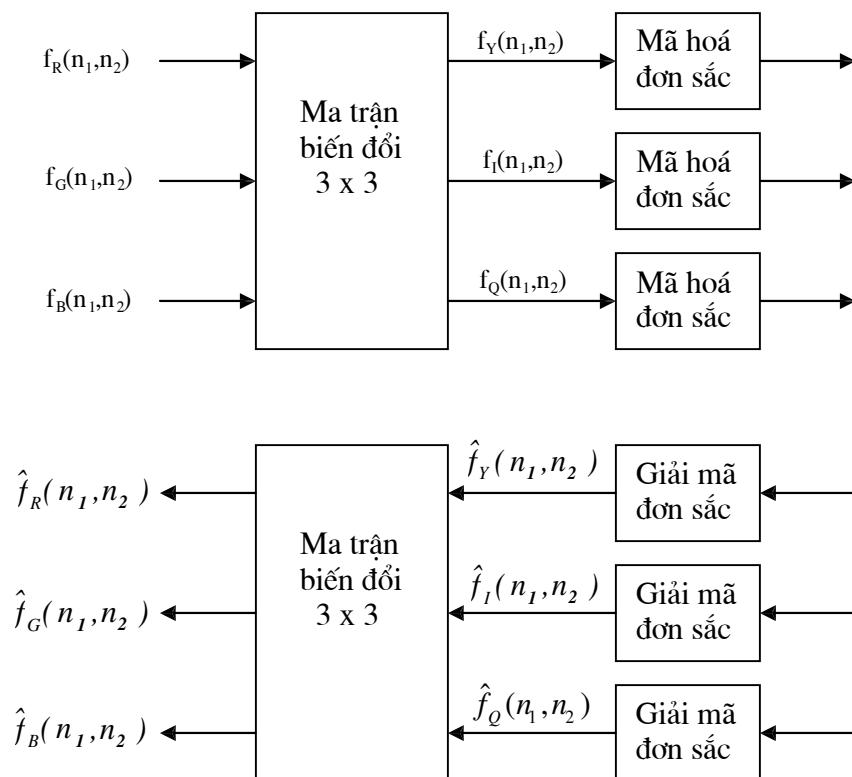
Trên đây chỉ mới nói mã hoá ảnh đen trắng. Nhiều phương pháp mã hoá ảnh đen trắng có thể mở rộng để mã hoá ảnh màu. Một ảnh màu có thể coi như ba ảnh đơn sắc  $f_R(n_1, n_2)$ ,  $f_G(n_1, n_2)$ ,  $f_B(n_1, n_2)$  đại biểu cho ba màu đỏ, lục, lam. Mỗi ảnh này coi như một ảnh đơn sắc và các phương pháp mã hoá trên đều áp dụng được. Tuy vậy ba thành phần  $f_R(n_1, n_2)$ ,  $f_G(n_1, n_2)$ ,  $f_B(n_1, n_2)$  rất tương quan với nhau và mã hoá riêng từng cái thì không hiệu quả. Có một cách để khai thác sự tương quan là đổi ba thành phần này ra thành một bộ ba thành phần khác có tính tương quan ít hơn. Chẳng hạn có thể dùng bộ  $f_Y(n_1, n_2)$ ,

$f_I(n_1, n_2)$ ,  $f_Q(n_1, n_2)$  trong đó  $f_Y(n_1, n_2)$  là độ chói và hai thành phần  $f_I(n_1, n_2)$ ,  $f_Q(n_1, n_2)$  là hai thành phần màu.

Phép biến đổi tuyến tính ma trận  $3 \times 3$  giữa RGB và YIQ có trong các sách về truyền hình. Hình 4.57 biểu diễn phép biến đổi từ RGB sang YIQ và mã hoá YIQ.

Một ưu điểm của phương pháp mã hoá YIQ thay cho RGB là những thành phần tần số cao của ảnh màu phân lõn tập trung vào thành phần Y. Do đó các thành phần I và Q có thể lấy mẫu thực với hệ số  $2 \times 2$  hoặc  $4 \times 4$  mà không ảnh hưởng đến chi tiết tần số cao của ảnh màu nếu ta dùng mã hoá dạng sóng. Nếu dùng mã hoá phép biến đổi thì khi mã hoá I và Q có thể lấy ít hệ số hơn là khi mã hoá Y. Thông thường tổng số bit phân phối cho I và Q chỉ bằng nửa số bit phân phối cho Y. Khi thêm màu không cần phải tăng tỷ lệ bit đến 3 lần. Ngoài ra tính thẩm mĩ màu dẫn đến một bức ảnh phục hồi đẹp hơn và có thể che lấp những chỗ hình ảnh bị xuống cấp.

Cũng tỷ lệ bit như nhau thì ảnh màu được mã hoá đẹp hơn bức ảnh đen trắng nhận được bằng cách sử dụng mã hoá thành phần Y của ảnh màu.



**Hình 4.57.** Mã hoá ảnh màu trong không gian YIQ.

### 6.3. Lỗi do kênh truyền gây ra.

Khi truyền ảnh, kênh truyền có thể gây ra một số lỗi. Một trường hợp thường xảy ra, và hay được dùng để nghiên cứu ảnh hưởng của kênh truyền, là bit đảo từ 0 sang 1 hay từ 1 sang 0 theo một xác suất đã cho nào đó. Ảnh hưởng của hiện tượng đảo bit lên bức ảnh được lặp lại phụ thuộc vào phần ảnh mà bit đại diện. Chẳng hạn, nếu bit bị đảo tham dự vào việc biểu diễn vị trí, thì sẽ giải mã ra một vị trí sai. Trong sự thảo luận ở đây, ta giả thiết rằng bit bị đảo đại diện cho một biên độ của dạng sóng hay một hệ số biến đổi.

Trong mã hoá PCM, bit đảo chỉ ảnh hưởng đến biên độ pixel mà nó đại biểu. Các bit đảo hiện ra như nhiều xung. Có thể giảm nhiễu này bằng những phương pháp đã thảo luận ở tiết 2.3 chương 2, như lọc trung vị, làm tròn pixel ngoại cỡ.

Trong mã hoá ảnh DPCM, ảnh hưởng của bit đảo không chỉ giới hạn ở cường độ một pixel, mà lỗi cứ lan truyền cho đến khi cường độ pixel được khởi lập lại (reinitialized) vào một thời điểm nào sau đó. Khi có đảo bit ở một pixel nào đó của tín hiệu sai số, thì pixel được tái lập ở bên máy phát khác với pixel được tái lập ở bên máy thu. Bởi vì những cường độ pixel được tái lập trong mã hoá DM hay DPCM đều được sử dụng theo kiểu đệ quy (recursively) và vì bộ mã hoá không biết có lỗi xảy ra, nên lỗi sẽ ảnh hưởng đến tất cả cường độ pixel tiếp theo cho đến khi khởi lập lại tại bộ mã hoá và bộ giải mã. Sự lan truyền lỗi là một trong những lý do làm cho giá trị dự báo bị nhân với một hệ số rò (leakage factor) nhỏ hơn 1 khi hình thành tín hiệu sai số đưa ra lượng tử hoá trong DM và DPCM. Ảnh hưởng của lỗi do kênh truyền biểu hiện như nhiều vạch, có thể làm giảm bớt bằng bộ lọc trung vị 1-D tác dụng theo phương thẳng đứng.

Một sự đảo bit trong trường hợp mã hoá ảnh biến đổi ảnh hưởng đến một hệ số biến đổi. Nhưng mỗi hệ số biến đổi ảnh hưởng đến mọi cường độ pixel trong ảnh con (subimage).

## 7. NHỮNG NHẬN XÉT BỔ SUNG.

Chương này đã thảo luận về một số phương pháp mã hoá ảnh, bây giờ ta tổng kết ưu khuyết điểm của từng loại. So với bộ mã hoá hệ số biến đổi thì các bộ mã hoá dạng sóng có ưu điểm là đơn giản. Vì chỉ có dạng sóng hoặc một đại lượng biến thiên nào đấy được mã hoá cho nên các bộ mã hoá đơn giản cả về cấu trúc lẫn cả về tính toán.

Hơn nữa trong những ứng dụng như truyền hình số, cần chất lượng ảnh cao thì bộ mã hoá dạng sóng cũng đáp ứng tốt như bộ mã hoá phép biến đổi khi tỷ lệ bit như nhau.

Các bộ mã hoá phép biến đổi về mặt tính toán đắt hơn bộ mã hoá dạng sóng bởi vì ảnh phải biến đổi ở đầu phát và biến đổi ngược ở đầu thu. Trong những ứng dụng mà tỷ lệ bit thấp như điện thoại video và điều khiển xe tàu từ xa, được phép hi sinh chất lượng thì các bộ mã hoá phép biến đổi đáp ứng tốt hơn các bộ mã hoá dạng sóng .

Các bộ mã hoá mô hình ảnh tính toán tốn kém nhất bởi vì ở đầu máy phát phải ước lượng các thông số mô hình đến đầu thu lại phải tổng hợp ảnh. Nhưng chúng có ưu thế trong những ứng dụng tỷ lệ bit rất thấp như điện thoại video cho người điếc, trong đó độ dễ hiểu là chính. Còn chất lượng ảnh có thể hi sinh để giảm tỷ lệ bit. Mã hoá mô hình ảnh còn đang ở trong giai đoạn nghiên cứu và còn nhiều việc phải làm trước khi đưa chúng ra ứng dụng thực tế.

Lượng tử hoá vectơ cải thiện chất lượng của algorit mã hoá so với lượng tử hoá vô hướng khi tỷ lệ bit như nhau. Cải thiện nhiều hay ít phụ thuộc vào rất nhiều thông số, chẳng hạn như sự phụ thuộc thống kê giữa các vô hướng trong vectơ. Lượng tử hoá vectơ cải thiện chất lượng trong các bộ mã hoá dạng sóng nhiều hơn trong các bộ mã hoá phép biến đổi. Sự cải thiện mà lượng tử hoá vectơ đem lại phải trả giá bằng tăng khối lượng tính toán và tăng yêu cầu về lưu trữ. Tuy vậy phần tính toán phức tạp ở phía máy phát. Trong những ứng dụng như sự quảng bá, khi số máy thu lớn gấp bội so với số máy phát thì tăng chi phí tính toán vì lượng tử hoá vectơ trở nên không đáng.

Các phương pháp mã hoá thích nghi thường yêu cầu tỷ lệ bit thấp hơn mã hoá không thích nghi. Tuy mã hoá thích nghi thiết bị có phức tạp hơn nhưng vì cải thiện chất lượng đáng kể cho nên có tốn kém hơn cũng đáng.

Mã hoá liên mành ứng dụng trong những trường hợp ảnh động và truyền hình khi mà chuỗi ảnh có tính tương quan theo thời gian đáng kể. Mã hoá liên mành yêu cầu phải lưu trữ mành trong quá trình mã hoá và do đó kéo theo một ít trễ.

Nói chung giảm tỷ lệ bit thì phải tăng chi phí tính toán và lưu trữ. Tỷ lệ bit lại quan hệ trực tiếp với giá cước của kênh truyền thông. Chi phí tính toán và lưu trữ làm tăng giá thành phần cứng của máy phát (bộ mã hoá) và máy thu (bộ giải mã).

Giữa giảm tải thông của kênh truyền và tăng giá thành phần cứng có mối quan hệ đối lập do đó ta phải tùy trường hợp mà lựa chọn phép mã hoá. Trong truyền hình số

nhiều máy thu dùng chung một kênh truyền do đó giảm giá thành phần cứng máy là quan trọng, còn giảm giá thành kênh thuê không quan trọng.

Về giá tiền phần cứng của máy phát và máy thu cũng là một yếu tố để lựa chọn phương pháp mã hoá. Trong truyền hình số máy thu nhiều, máy phát ít, dùng lự ợng tử hoá vectơ làm cho phần tính toán bên máy phát phức tạp lên nhiều còn máy thu không cần phức tạp do đó chọn lượng tử hoá vectơ là hợp lý. Ngược lại trong việc điều khiển từ xa các xe tàu có nhiều máy phát và ít máy thu, khi đó lại cần những phương ph áp mã hoá đơn giản ở máy phát còn máy thu cho phép sử dụng những phương pháp phức tạp để giảm tạp âm lượng tử hoá.

Từ những thảo luận trên, thấy rõ là trong thực tế có rất nhiều yếu tố ảnh hưởng đến sự lựa chọn hệ mã hoá ảnh. Phát triển một algôrit mã hoá ảnh thích hợp cho một ứng dụng cụ thể là cả một quá trình phải đắn đo, có khi phải thử đi thử lại nhiều lần và có sự can thiệp của con người. Hy vọng rằng những điều cơ bản về mã hoá ảnh số trình bày trong chương này sẽ có thể hướng dẫn độc giả trong sự lựa chọn đó.

## 7. VÀI LỜI KẾT LUẬN.

Trong các chương 1, 2, 3, và 4, chúng ta đã thảo luận những điều cơ bản về xử lý ảnh số. Chúng ta đã đưa ra một số algôrit khác nhau về xử lý ảnh và thảo luận về những khả năng và những hạn chế của nó. Cần lưu ý rằng mục tiêu của quyển sách này không phải là cung cấp những algôrit “may mắn” có thể sử dụng cho những ứng dụng cụ thể. Trái lại, mục tiêu của chúng tôi là nghiên cứu về những kiến thức cơ bản và những ý tưởng chính có thể dẫn đến sự phát triển các algôrit khác nhau. Sự thảo luận của chúng ta là để hướng dẫn cho việc phát triển algôrit xử lý ảnh hợp lý nhất đối với một ứng dụng đã cho. Với những ứng dụng khác nhau, các yếu tố phải xét đến khi phát triển một algôrit xử lý ảnh rất khác nhau, cho nên thông thường ít khi có thể tìm ra một algôrit “may mắn” phù hợp một cách lý tưởng với một ứng dụng cụ thể.

Một bước quan trọng khi phát triển một hệ xử lý ảnh trong một môi trường ứng dụng thực tế là xác định được mục tiêu tổng quát thật rõ ràng. Trong những ứng dụng nhằm xử lý ảnh cho con người xem, thì phải xét đến những thuộc tính của hệ thị giác con người. Trong những ứng dụng nhằm xử lý ảnh để cải thiện tính năng của máy móc, thì điều quan trọng là phải xét đến những thuộc tính của cái máy. Như vậy những cách tiếp cận để phát triển một hệ xử lý ảnh cụ thể rất khác nhau, tùy theo mục tiêu tổng thể của ứng dụng cụ thể.

Một bước quan trọng khác là xác định những hạn chế do môi trường ứng dụng đặt ra. Trong một vài ứng dụng, cần thiết phải xử lý ảnh theo thời gian thực . Trong một số ứng dụng khác thì đó không phải là điều bắt buộc. Rõ ràng là cách tiếp cận để phát triển một hệ xử lý ảnh chịu nhiều ảnh hưởng của những hạn chế do bối cảnh ứng dụng đặt ra.

Một bước quan trọng thứ ba là thu lượm thông tin về ảnh cần xử lý. Thông tin này sẽ được khai thác khi triển khai xử lý ảnh. Chẳng hạn, nếu trong ảnh phải xử lý chứa nhiều hình ảnh nhà cao tầng, có nhiều đường nét nằm ngang và thẳng đứng, thông tin này có thể dùng để thiết kế một bộ lọc thông thấp thích nghi để giảm nhiễu nền.

Khi đã cho mục tiêu tổng thể, các điều kiện hạn chế và thông tin về lớp ảnh phải xử lý, thì cách tiếp cận hợp lý để phát triển algôrit xử lý ảnh là xác định xem thử chỉ có một hay là có nhiều phương pháp khác nhau, chẳng hạn một phương pháp nào đó trong quyển sách này, có thể giải được bài toán. Thông thường thì cách làm việc hợp lý thoạt tiên hãy tìm cách thích ứng những phương pháp đã biết để giải bài toán. Nếu những phương pháp đã có không áp dụng được, hoặc là hiệu năng của những phương pháp đã có không đủ để đáp ứng những yêu cầu của bài toán, thì phải phát triển những cách tiếp cận mới. Lịch sử xử lý ảnh số chưa dài, do đó cũng còn nhiều chỗ cho các cách tiếp cận và phương pháp mới. Chúng tôi hy vọng rằng những điều cơ bản về xử lý ảnh số trình bày trong quyển sách này sẽ tạo nên một cơ sở kiến thức, để đọc giả có thể đọc thêm nhiều tài liệu về các chuyên mục, và các ứng dụng kết quả nghiên cứu lý thuyết vào những vấn đề thực tiễn, cũng như về các tiến bộ mới trong nghiên cứu và triển khai thuộc lĩnh vực này.

## PHỤ LỤC

### QUÁ TRÌNH NGẪU NHIÊN

#### 1. BIẾN NGẪU NHIÊN

Một biến ngẫu nhiên thực  $x$  là biến có thể có mọi giá trị thực một cách ngẫu nhiên, chẳng hạn kết quả tung đồng xu. Nó hoàn toàn được đặc trưng bởi hàm mật độ xác suất  $p_x(x_0)$ . Cúc chú  $x$  ở  $p_x(x_0)$  ký hiệu biến ngẫu nhiên  $x$ , còn  $x_0$  ký hiệu một giá trị cụ thể của  $x$ . Xác suất để  $x$  nằm giữa  $a$  và  $b$  là:

$$Prob[a \leq x \leq b] = \int_{x_0=a}^b p_x(x_0) dx_0. \quad (1)$$

Vì xác suất để một sự kiện chắc chắn xảy ra là 1, nên:

$$Prob[-\infty \leq x \leq \infty] = \int_{x_0=-\infty}^{\infty} p_x(x_0) dx_0 = 1. \quad (2)$$

Kỳ vọng  $E[f(x)]$  của hàm của biến ngẫu nhiên  $x$  là:

$$E[f(x)] = \int_{x_0=-\infty}^{\infty} f(x_0) p_x(x_0) dx_0. \quad (3)$$

Kỳ vọng trên được định nghĩa trên một toán tử tuyến tính và thoả mãn

$$E[f(x) + g(x)] = E[f(x)] + E[g(x)] \quad (4)$$

và

$$E[cf(x)] = cE[f(x)] \quad (5)$$

trong đó  $c$  là một hằng số vô hướng

$E[x^n]$ , momen thứ  $n$  của biến ngẫu nhiên  $x$ , là:

$$E[x^n] = \int_{x_0=-\infty}^{\infty} x_0^n p_x(x_0) dx_0. \quad (6)$$

Momen thứ nhất gọi là trung vị hay trung bình của x. Từ (6)

$$E[x] = \int_{x_0=-\infty}^{\infty} x_0 p_x(x_0) dx_0. \quad (7)$$

Phương sai của x được định nghĩa bởi:

$$Var[x] = E[(x - E[x])^2] = E[x^2 - 2xE[x] + E^2[x]] = E[x^2] - E^2[x] \quad (8)$$

Độ lệch chuẩn của x, ký hiệu là s.d. [x], được định nghĩa là

$$s.d.[x] = (Var[x])^{1/2} \quad (9)$$

Hai biến ngẫu nhiên x và y, hoàn toàn được đặc trưng bởi hàm mật độ xác suất đồng thời  $p_{x,y}(x_0, y_0)$ . Chúng là độc lập thống kê với nhau nếu chúng thoả mãn:

$$p_{x,y}(x_0, y_0) = p_x(x_0)p_y(y_0). \quad \text{với mọi } (x_0, y_0) \quad (10)$$

Kỳ vọng của một hàm hai biến ngẫu nhiên được định nghĩa là:

$$E[f(x, y)] = \int_{x_0=-\infty}^{\infty} \int_{y_0=-\infty}^{\infty} f(x_0, y_0) p_{x,y}(x_0, y_0) dx_0 dy_0. \quad (11)$$

Hai biến ngẫu nhiên x, y là độc lập tuyến tính nếu:

$$E[xy] = E[x]E[y]. \quad (12)$$

Độc lập thống kê kéo theo độc lập tuyến tính, nhưng độc lập tuyến tính không kéo theo độc lập thống kê.

Hàm mật độ xác suất của một biến ngẫu nhiên khi cho trước x là một biến ngẫu nhiên y được ký hiệu bằng  $p_{x|y}(x_0|y_0)$  và định nghĩa là:

$$p_{x|y}(x_0|y_0) = p_{x,y}(x_0, y_0) / p_y(y_0). \quad (13)$$

Nếu x và y là độc lập thống kê, thì biết y không cho ta biết gì về x, và  $p_{x|y}(x_0|y_0)$  rút lại chỉ còn là  $p_x(x_0)$ . Kỳ vọng  $E[f(x)|y]$  của một hàm x khi có điều kiện y được định nghĩa là:

$$E[f(x)|y] = \int_{x_0=-\infty}^{\infty} f(x_0) p_{x|y}(x_0|y) dx_0. \quad (14)$$

Biến ngẫu nhiên phức w được định nghĩa bởi:

$$w = x + jy \quad (15)$$

khi x và y là hai biến ngẫu nhiên định nghĩa như trên. Kỳ vọng E [f(w)] của một hàm của w là:

$$\begin{aligned} E[f(w)] &= E[f(x + jy)] \\ &= \int_{x_0=-\infty}^{\infty} \int_{y_0=-\infty}^{\infty} f(x_0 + jy_0) p_{x,y}(x_0, y_0) dx_0 dy_0. \end{aligned} \quad (16)$$

Trung vị của w được định nghĩa bởi:

$$E[\omega] = E[x + jy] = E[x] + jE[y]. \quad (17)$$

Phương sai của w được định nghĩa bởi:

$$\begin{aligned} Var[w] &= E[(w - E[\omega])(w - E[w])^*] \\ &= E[\omega\omega^*] - E[\omega]E^*[ ] \\ &= E[x^2 + y^2] - (E^2[x] + E^2[y]) \\ &= Var[x] + Var[y] \end{aligned} \quad (18)$$

Lưu ý rằng phương sai của w là thực và không âm, tuy w có thể là phức.

Nhiều biến ngẫu nhiên x, y, z, .. hoàn toàn được đặc trưng bởi hàm mật độ xác suất đồng thời của chúng  $p_{x,y,z\dots}(x_0, y_0, z_0, \dots)$ . Các định nghĩa của kỳ vọng và độc lập tuyến tính. Độc lập thống kê cũng tương tự như trường hợp hai biến ngẫu nhiên.

## 2. QUÁ TRÌNH NGẦU NHIÊN

Một tập biến ngẫu nhiên với số lượng vô hạn gọi là một quá trình ngẫu nhiên. Nếu các biến ngẫu nhiên là thực, thì tập đó gọi là quá trình ngẫu nhiên thực. Nếu các biến ngẫu nhiên là phức, thì tập đó gọi là quá trình ngẫu nhiên phức. Mọi kết quả trong tiết này đều ứng với quá trình ngẫu nhiên phức, trừ khi có lời dẫn riêng. Các kết quả đối

với quá trình ngẫu nhiên thực là những trường hợp đặc biệt của quá trình ngẫu nhiên phức.

Ta ký hiệu một tập biến ngẫu nhiên phức với số lượng vô hạn là  $x(n_1, n_2)$ , trong đó  $x(n_1, n_2)$  với một  $(n_1, n_2)$  cụ thể là một biến ngẫu nhiên. Quá trình ngẫu nhiên được đặc trưng đầy đủ bởi hàm mật độ xác suất đồng thời củ a tất cả các biến ngẫu nhiên. Nếu ta lấy một mẫu của quá trình ngẫu nhiên  $x(n_1, n_2)$ , thì đây là một dãy 2-D. Ta gọi dãy 2-D này là tín hiệu ngẫu nhiên và cũng ký hiệu nó là  $x(n_1, n_2)$ . Ta sẽ tùy theo bối cảnh mà hiểu  $x(n_1, n_2)$  lúc nào là ký hiệu của quá trình ngẫu nhiên còn lúc nào chỉ là ký hiệu một mẫu của quá trình đó. Hoặc là, trong trường hợp cụ thể nói rõ ý nghĩa ký hiệu. Tập của tất cả các mẫu có thể được gọi là tập hợp của quá trình ngẫu nhiên  $x(n_1, n_2)$ .

Hàm tự tương quan, hay gọi tắt là hàm tương quan của quá trình ngẫu nhiên  $x(n_1, n_2)$ , ký hiệu là  $R_x(n_1, n_2; k_1, k_2)$  được định nghĩa theo:

$$R_x(n_1, n_2; k_1, k_2) = E[x(n_1, n_2)x^*(k_1, k_2)] \quad (19)$$

Tương quan là kỳ vọng của tích hai biến ngẫu nhiên phức  $x(n_1, n_2)$  và  $x^*(k_1, k_2)$ . Hàm tự hiệp biến, hay gọi tắt là hàm hiệp biến của  $x(n_1, n_2)$ , ký hiệu là  $\gamma_x(n_1, n_2; k_1, k_2)$  được định nghĩa theo:

$$\begin{aligned} \gamma_x(n_1, n_2; k_1, k_2) &= E[(x(n_1, n_2) - E[x(n_1, n_2)])(x(k_1, k_2) - E[x(k_1, k_2)])^*] \\ &= E[x(n_1, n_2)x^*(k_1, k_2)] - E[x(n_1, n_2)]E^*[x(k_1, k_2)] \\ &= R_x(n_1, n_2; k_1, k_2) - E[x(n_1, n_2)]E^*[x(k_1, k_2)] \end{aligned} \quad (20)$$

Một quá trình ngẫu nhiên  $x(n_1, n_2)$  được gọi là quá trình có trung vị bằng không nếu:

$$E[x(n_1, n_2)] = 0 \quad \text{với mọi } (n_1, n_2) \quad (21)$$

Với một quá trình ngẫu nhiên có trung vị bằng không:

$$R_x(n_1, n_2; k_1, k_2) = \gamma_x(n_1, n_2; k_1, k_2). \quad (22)$$

Một quá trình ngẫu nhiên  $x(n_1, n_2)$  có trung vị khác không bao giờ cũng có thể biến đổi thành một quá trình ngẫu nhiên có trung vị bằng không bằng cách đem  $x(n_1, n_2)$  trừ đi  $E[x(n_1, n_2)]$ . Chúng tôi giả định  $x(n_1, n_2)$  bao giờ cũng là một quá trình trung vị bằng không, (22) luôn luôn có hiệu lực, trừ khi có lời dẫn mới.

Quá trình ngẫu nhiên  $x(n_1, n_2)$  được gọi là dừng nếu hàm mật độ xác suất đồng thời không phụ thuộc điểm gốc của chỉ số  $(n_1, n_2)$

$$p_{x(n_1, n_2)(n_1, n_2), \dots}(x_1, x_2, \dots) = p_{x(n_1+k_1, n_2+k_2)(n_1+k_1, n_2+k_2), \dots}(x_1, x_2, \dots). \quad (23)$$

với mọi giá trị cố định của  $k_1$  và  $k_2$ .

Với một quá trình ngẫu nhiên dừng  $x(n_1, n_2)$ ,  $E[x(n_1, n_2)]$  là một hằng số độc lập với  $n_1$  và  $n_2$ , còn  $R_x(n_1, n_2; k_1, k_2)$  chỉ là hàm của  $n_1 - k_1$  và  $n_2 - k_2$

$$E[x(n_1, n_2)] = m_x \quad \text{với mọi } x(n_1, n_2) \quad (24)$$

$$\begin{aligned} R_x(n_1 - k_1, n_2 - k_2) &= R_x(n_1 - k_1, n_2 - k_2; 0, 0) \\ &= E[x(n_1 - k_1, n_2 - k_2)x^*(0, 0)] \end{aligned} \quad (25)$$

Viết lại (25) ta có:

$$R_x(n_1, n_2) = E[x(k_1, k_2)x^*(k_1 - n_1, k_2 - n_2)] \quad \text{với mọi } (k_1, k_2) \quad (26)$$

Lưu ý rằng các acgumen  $n_1$  và  $n_2$  trong  $R_x(n_1, n_2)$  ở công thức (26) là  $k_1 - n_1$  trừ đi  $k_1$  và  $k_2 - n_2$  trừ đi  $k_2$ . Một quá trình ngẫu nhiên thoả mãn (24) và (26) nhưng không nhất thiết thoả mãn (23) được gọi là dừng theo nghĩa rộng. Từ (26)

$$R_x(n_1, n_2) = R_x^*(-n_1, -n_2) \quad (27)$$

Từ (27) thấy rằng dãy tương quan có đối xứng liên hiệp phức.

Một quá trình ngẫu nhiên dừng  $x(n_1, n_2)$  được gọi là ecgodic nếu trung bình theo thời gian (hay không gian) bằng trung bình của tập hợp. Giả sử ta muốn ước lượng  $m_x = E[x(n_1, n_2)]$  từ các mẫu của một  $x(n_1, n_2)$  dừng. Vì  $m_x$  đại diện cho trung bình của tập hợp, ta cần có một tập hợp (tất cả các mẫu có thể có) của  $x(n_1, n_2)$  với mọi bộ  $(n_1, n_2)$ . Nếu  $x(n_1, n_2)$  ecgodic thì có thể từ một mẫu của  $x(n_1, n_2)$  tính ra  $m_x$ :

$$m_x = E[x(n_1, n_2)] = \lim_{N \rightarrow \infty} \frac{1}{(2N+1)^2} \sum_{n_1=-N}^N \sum_{n_2=-N}^N x(n_1, n_2). \quad (28)$$

Tương tự, với quá trình ecgodic:

$$\begin{aligned} R_x(n_1, n_2) &= E[x(k_1, k_2)x^*(k_1 - n_1, k_2 - n_2)] \\ &= \lim_{N \rightarrow \infty} \frac{1}{(2N+1)^2} \sum_{k_1=-N}^N \sum_{k_2=-N}^N x(k_1, k_2)x^*(k_1 - n_1, k_2 - n_2). \end{aligned} \quad (29)$$

Các phương trình (28) và (29) cho phép xác định ra  $m_x$  hay  $R_x(n_1, n_2)$  từ một mẫu của  $x(n_1, n_2)$ . Lưu ý rằng tính egodic kéo theo tính dừng (theo nghĩa rộng), nhưng tính dừng không kéo theo tính egodic.

Phổ công suất  $P_x(w_1, w_2)$  của một quá trình dừng  $x(n_1, n_2)$  được định nghĩa bởi:

$$P_x(w_1, w_2) = F[R_x(n_1, n_2)] = \lim_{N \rightarrow \infty} \frac{1}{(2N+1)^2} \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} R_x(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2}. \quad (30)$$

Từ (27) và (30)

$$P_x(w_1, w_2) = P_x^*(w_1, w_2) \quad (31)$$

Từ (31),  $P_x(w_1, w_2)$  luôn luôn là số thực. Hơn nữa, có thể chứng minh là  $P_x(w_1, w_2)$  luôn luôn không âm; nghĩa là:

$$P_x(w_1, w_2) \geq 0 \quad \text{cho tất cả } (w_1, w_2) \quad (32)$$

Từ (26) và (30)

$$\begin{aligned} R_x(0,0) &= \sigma_x^2 = E[(n_1, n_2)x^*(n_1, n_2)] \\ &= \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} P_x(\omega_1, \omega_2) d\omega_1 d\omega_2. \end{aligned} \quad (33)$$

Cũng có thể chứng minh được rằng:

$$R_x(0,0) \geq |R_x(n_1, n_2)| \quad \text{với mọi } (n_1, n_2) \quad (34)$$

Giá trị  $\sigma_x$  gọi là công suất bình quân của quá trình ngẫu nhiên  $x(n_1, n_2)$

Một quá trình ngẫu nhiên được gọi là quá trình nhiễu trắng nếu:

$$\begin{aligned} R_x(n_1, n_2; k_1, k_2) &= E[x(n_1, n_2)x^*(k_1, k_2)] \\ &= \begin{cases} \sigma_x^2(n_1, n_2), & n_1 = k_1, n_2 = k_2 \\ 0, & \text{các trường hợp khác.} \end{cases} \end{aligned} \quad (35)$$

Với một quá trình nhiễu trắng dừng thì:

$$\begin{aligned} R_x(n_1, n_2) &= E[x(k_1, k_2)x^*(k_1 - n_1, k_2 - n_2)] \\ &= \sigma_x^2 \delta(n_1, n_2). \end{aligned} \quad (36)$$

Từ (30) và (36) nhận được phổ công suất của quá trình nhiễu trắng dừng:

$$P_x(\omega_1, \omega_2) = \sigma_x^2 \quad \text{với mọi } (\omega_1, \omega_2) \quad (37)$$

Phổ công suất là một hằng số với mọi tần số, do đó gọi là nhiễu “trắng”.

Với một quá trình ngẫu nhiên  $x(n_1, n_2)$  thực, thì (19), (26), (27), (29) rút gọn lại là:

$$R_x(n_1, n_2; k_1, k_2) = E[x(n_1, n_2)x(k_1, k_2)] \quad (38)$$

$$R_x(n_1, n_2) = E[x(k_1, k_2)x(k_1 - n_1, k_2 - n_2)] \quad (39)$$

với mọi  $(k_1, k_2)$ , với một quá trình dừng

$$R_x(n_1, n_2) = R_X(-n_1, -n_2) \quad (40)$$

Và  $R_x(n_1, n_2) = \lim_{N \rightarrow \infty} \frac{1}{(2N+1)^2} \sum_{k_1=-N}^N \sum_{k_2=-N}^N x(k_1, k_2)x(k_1 - n_1, k_2 - n_2).$  (41)

với một quá trình ergodic.

Từ (40),

$$P_x(\omega_1, \omega_2) = P_x(-\omega_1, -\omega_2) \quad (42)$$

Vì  $P_x(\omega_1, \omega_2)$  là tuần hoàn với chu kỳ là  $2\pi$  cho cả các biến  $\omega_1$  và  $\omega_2$ , từ (42),  $P_x(\omega_1, \omega_2)$  cho một quá trình ngẫu nhiên thực hoàn toàn được xác định bởi  $P_x(\omega_1, \omega_2)$  khi  $-\pi \leq \omega_1 \leq \pi, 0 \leq \omega_2 \leq \pi$ . Vì thế với một quá trình ngẫu nhiên thực,  $P_x(\omega_1, \omega_2)$  chỉ cần biểu diễn trong khoảng  $-\pi \leq \omega_1 \leq \pi, 0 \leq \omega_2 \leq \pi$ .

Hai quá trình ngẫu nhiên  $x(n_1, n_2)$  và  $y(n_1, n_2)$  có thể được đặc trưng hoàn toàn bởi hàm mật độ xác suất đồng thời của các biến ngẫu nhiên trong  $x(n_1, n_2)$  và  $y(n_1, n_2)$ . Hàm tương quan chéo  $R_{xy}(n_1, n_2; k_1, k_2)$  của  $x(n_1, n_2)$  và  $y(n_1, n_2)$  định nghĩa theo:

$$R_{xy}(n_1, n_2; k_1, k_2) = E[x(n_1, n_2)y^*(k_1, k_2)]. \quad (43)$$

Hàm hiệp biến chéo  $\gamma_{xy}(n_1, n_2; k_1, k_2)$  của  $x(n_1, n_2)$  và  $y(n_1, n_2)$  định nghĩa theo:

$$\gamma_{xy}(n_1, n_2; k_1, k_2) = E[x(n_1, n_2) - E[x(n_1, n_2)]][y^*(k_1, k_2) - E[y^*(k_1, k_2)]] \quad (44)$$

Từ (43) và (44), với các quá trình trung vị bằng không  $x(n_1, n_2)$  và  $y(n_1, n_2)$ .

$$R_{xy}(n_1, n_2; k_1, k_2) = \gamma_{xy}(n_1, n_2; k_1, k_2). \quad (45)$$

Với các quá trình  $x(n_1, n_2)$  và  $y(n_1, n_2)$  dừng:

$$R_{xy}(n_1, n_2) = E[x(k_1, k_2)y^*(k_1 - n_1, k_2 - n_2)]. \quad \text{độc lập với } (k_1, k_2) \quad (46)$$

Với các quá trình  $x(n_1, n_2)$  và  $y(n_1, n_2)$  ergodic:

$$R_{xy}(n_1, n_2) = E[x(k_1, k_2)y^*(k_1 - n_1, k_2 - n_2)] \quad (47)$$

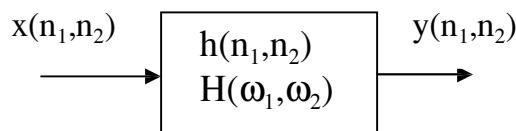
$$= \lim_{N \rightarrow \infty} \frac{1}{(2N+1)^2} \sum_{k_1=-N}^N \sum_{k_2=-N}^N x(k_1, k_2)y^*(k_1 - n_1, k_2 - n_2).$$

Phổ công suất chéo  $P_{xy}(\omega_1, \omega_2)$  của hai quá trình dừng đồng thời  $x(n_1, n_2)$  và  $y(n_1, n_2)$  được định nghĩa bởi:

$$P_{xy}(\omega_1, \omega_2) = F[R_{xy}(n_1, n_2)] = \lim_{N \rightarrow \infty} \frac{1}{(2N+1)^2} \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} R_{xy}(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2}. \quad (48)$$

### 3. TÍN HIỆU NGẪU NHIÊN LÀ ĐẦU VÀO CỦA HỆ TUYẾN TÍNH

Xét một quá trình ngẫu nhiên dừng  $x(n_1, n_2)$  với giá trị trung vị  $m_x$  và hàm tương quan  $R_x(n_1, n_2)$ . Giả sử nhận được một quá trình ngẫu nhiên mới  $y(n_1, n_2)$  khi cho  $x(n_1, n_2)$  đi qua một hệ tuyến tính bất biến (LSI) mà đáp ứng xung là  $h(n_1, n_2)$ , như biểu diễn trên hình 1.



**Hình 1:** Hệ tuyến tính bất biến khi dịch chuyển, với tín hiệu ngẫu nhiên  $x(n_1, n_2)$  ở đầu vào và tín hiệu ngẫu nhiên  $y(n_1, n_2)$  tương ứng ở đầu ra.

Quan hệ giữa  $x(n_1, n_2)$  và  $y(n_1, n_2)$  là:

$$y(n_1, n_2) = h(n_1, n_2) * x(n_1, n_2)$$

$$= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2). \quad (49)$$

Lưu ý rằng  $x(n_1, n_2)$  là một quá trình ngẫu nhiên còn  $h(n_1, n_2)$  là một tín hiệu xác định. Trong thực tế  $h(n_1, n_2)$  thường là một tín hiệu thực, nhưng ở đây ta sẽ suy diễn ra một kết quả tổng quát có thể áp dụng cho cả  $h(n_1, n_2)$  phức.

Ta muốn xác định  $E[y(n_1, n_2)]$ ,  $R_y(n_1, n_2; k_1, k_2)$ ,  $R_{xy}(n_1, n_2; k_1, k_2)$  và  $R_{yx}(n_1, n_2; k_1, k_2)$ . Ta sẽ thấy rằng  $y(n_1, n_2)$  là một quá trình ngẫu nhiên dừng. Để xác định  $E[y(n_1, n_2)]$ , ta áp dụng toán tử kỳ vọng cho (49), biết rằng  $h(n_1, n_2)$  là một tín hiệu xác định.

$$\begin{aligned} E[y(n_1, n_2)] &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} E[x(k_1, k_2)] h(n_1 - k_1, n_2 - k_2) \\ &= m_x H(0,0) \quad \text{với mọi } (n_1, n_2) \\ &= m_y. \end{aligned} \quad (50)$$

Nếu  $x(n_1, n_2)$  có trung vị bằng không thì  $y(n_1, n_2)$  cũng có trung vị bằng không

Để nhận được  $R_y(n_1, n_2; k_1, k_2)$  từ (49),

$$\begin{aligned} R_y(n_1, n_2; k_1, k_2) &= E[y(n_1, n_2) y^*(k_1, k_2)] \\ &= \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(l_1, l_2) h^*(m_1, m_2) E[x(n_1 - l_1, n_2 - l_2) x^*(k_1 - m_1, k_2 - m_2)] \\ &= \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(l_1, l_2) h^*(m_1, m_2) R_x(n_1 - k_1 - l_1 + m_1, n_2 - k_2 - l_2 + m_2). \end{aligned} \quad (51)$$

Từ (51),  $R_y(n_1, n_2; k_1, k_2)$  là một hàm của  $n_1 - k_1$  và  $n_2 - k_2$ , và như vậy khi ký hiệu  $R_y(n_1, n_2; 0, 0)$ , bằng  $R_y(n_1, n_2)$ , ta có thể viết lại (51) là:

$$\begin{aligned} R_y(n_1, n_2) &= \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(l_1, l_2) h^*(m_1, m_2) R_x(n_1 - l_1 + m_1, n_2 - l_2 + m_2) \\ &= R_x(n_1, n_2) * h(n_1, n_2) * h^*(-n_1, -n_2). \end{aligned} \quad (52)$$

Tương tự:

$$R_{xy}(n_1, n_2) = R_x(n_1, n_2) * h^*(-n_1, -n_2). \quad (53)$$

$$R_{yx}(n_1, n_2) = R_x(n_1, n_2) * h(n_1, n_2). \quad (54)$$

Từ (52), (53) và (54):

$$P_y(\omega_1, \omega_2) = P_x(\omega_1, \omega_2) |H(\omega_1, \omega_2)|^2 \quad (55)$$

$$P_{xy}(\omega_1, \omega_2) = P_x(\omega_1, \omega_2) H^*(\omega_1, \omega_2) \quad (56)$$

$$P_{yx}(\omega_1, \omega_2) = P_x(\omega_1, \omega_2) H(\omega_1, \omega_2) \quad (57)$$

Phương trình (55) đặc biệt quan trọng và được dùng nhiều trong các ứng dụng. Nó cũng có thể dùng để chứng minh rằng  $P_x(\omega_1, \omega_2)$  và  $P_y(\omega_1, \omega_2)$  là không âm và gọi chúng là phổ công suất hay mật độ phổ công suất là hợp lý.

#### 4. BỘ LỌC WIENER KHÔNG NHÂN QUẢ

Giả sử có tín hiệu  $s(n_1, n_2)$  và nhiễu  $w(n_1, n_2)$ , theo thứ tự là mẫu của các quá trình ngẫu nhiên dừng trung vị bằng không  $s(n_1, n_2)$  và  $w(n_1, n_2)$ . Tín hiệu bị nhiễu là  $x(n_1, n_2)$ :

$$x(n_1, n_2) = s(n_1, n_2) + w(n_1, n_2) \quad (58)$$

Ta muốn từ  $x(n_1, n_2)$  xác định ra  $s(n_1, n_2)$  bằng bộ ước lượng tuyến tính sau đây:

$$\hat{s}(n_1, n_2) = x(n_1, n_2) * h(n_1, n_2) \quad (59)$$

Ước lượng tuyến tính là một hệ tuyến tính bất biến (LSI) vì ta đang làm việc với nhữngh quá trình dừng. Tiêu chí sai số được sử dụng là:

$$\text{Error} = E[|e(n_1, n_2)|^2] \quad (60a)$$

$$e(n_1, n_2) = s(n_1, n_2) - \hat{s}(n_1, n_2) \quad (60b)$$

trong đó  $\hat{s}(n_1, n_2)$  là giá trị ước lượng của  $s(n_1, n_2)$ . Đây gọi là bài toán tuyến tính ước lượng sai số quan phương cực tiểu vì ta đang dùng bộ ước lượng tuyến tính và cực tiểu hoá khoảng cách quan phương giữa  $s(n_1, n_2)$  và  $\hat{s}(n_1, n_2)$ .

Bài toán ước lượng tín hiệu này có thể giải bằng cách dùng nguyên lý trực giao. Nguyên lý này nói rằng sai số trong (40) được cực tiểu hóa bằng cách yêu cầu  $e(n_1, n_2)$  không kết hợp với bất kỳ biến ngẫu nhiên nào trong  $x^*(n_1, n_2)$ . Từ nguyên lý trực giao:

$$E[e(n_1, n_2) x^*(m_1, m_2)] = 0 \quad \text{với mọi } (n_1, n_2) \text{ và } (m_1, m_2). \quad (61)$$

Từ (59), (60b) và (61):

$$\begin{aligned} E[s(n_1, n_2) x^*(m_1, m_2)] &= E[\hat{s}(n_1, n_2) x^*(m_1, m_2)] \\ &= E[h(n_1, n_2) * x(n_1, n_2) x^*(m_1, m_2)] \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) E[x(n_1 - k_1, n_2 - k_2) x^*(m_1, m_2)]. \end{aligned} \quad (62)$$

Viết lại (62) ta có:

$$R_{sx}(n_1 - m_1, n_2 - m_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h(k_1, k_2) R_x[x(n_1 - k_1, n_2 - k_2) x^*(m_1, m_2)] \quad (63)$$

Từ (63)

$$R_{sx}(n_1, n_2) = h(n_1, n_2) * R_x(n_1, n_2) \quad (64)$$

và do đó:

$$H(\omega_1, \omega_2) = \frac{P_{sx}(\omega_1, \omega_2)}{P_x(\omega_1, \omega_2)} \quad (65)$$

Bộ lọc  $H(w_1, w_2)$  trong (65) gọi là bộ lọc Wiener không nhân quả.

Giả thiết  $s(n_1, n_2)$  không kết hợp với  $w(n_1, n_2)$ . Thì ta có:

$$E[s(n_1, n_2) w^*(m_1, m_2)] = E[s(n_1, n_2)] E[w^*(m_1, m_2)] \quad (66)$$

Từ (66) và lưu ý rằng  $s(n_1, n_2)$  và  $w(n_1, n_2)$  là những quá trình trung vị bằng không, ta nhận được:

$$R_{sx}(n_1, n_2) = R_s(n_1, n_2) \quad (67a)$$

$$R_x(n_1, n_2) = R_s(n_1, n_2) + R_w(n_1, n_2) \quad (67b)$$

Từ (67)

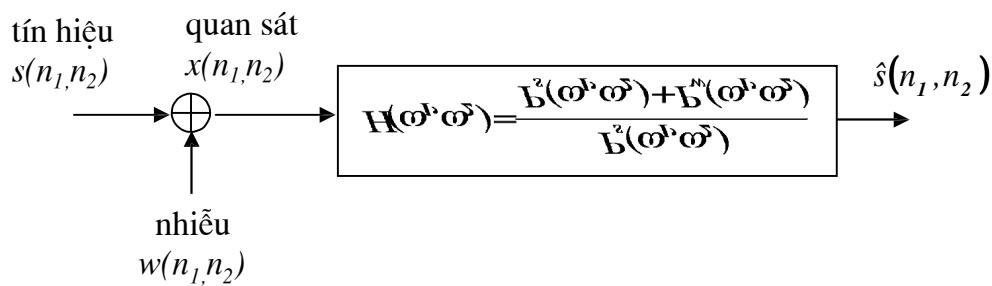
$$P_{sx}(\omega_1, \omega_2) = P_s(\omega_1, \omega_2). \quad (68a)$$

$$P_x(\omega_1, \omega_2) = P_s(\omega_1, \omega_2) + P_w(\omega_1, \omega_2) \quad (68b)$$

Từ (65) và (68), suy ra bộ lọc Wiener không nhân quả  $H(\omega_1, \omega_2)$ :

$$H(\omega_1, \omega_2) = \frac{P_s(\omega_1, \omega_2)}{P_s(\omega_1, \omega_2) + P_w(\omega_1, \omega_2)} \quad (69)$$

Ứng dụng của bộ lọc Wiener không nhân quả ở (69) vào bài toán ước lượng tín hiệu được biểu diễn ở hình 2. Bộ lọc Wiener không nhân quả là cơ sở của nhiều hệ đã được phát triển, như hệ cải thiện tiếng nói hoặc hệ phục hồi ảnh.



**Hình 2:** Ứng dụng của bộ lọc Wiener không nhân quả để ước lượng tín hiệu theo tiêu chí sai số quân phương cực tiểu tuyến tính.

Các kết quả này cũng có thể ứng dụng để giải bài toán sai số quân phương tuyến tính khi  $m_s$  và  $m_w$  là các giá trị trung vị của các quá trình tín hiệu và nhiễu.

Cách giải bài toán trong trường hợp này bao gồm:

- đếm tín hiệu bị nhiễu  $x(n_1, n_2)$  trừ  $E[x(n_1, n_2)] = m_s + m_w$ ,
- áp dụng bộ lọc Wiener không nhân quả khai triển cho các quá trình trung vị bằng không,
- cộng  $m_s$  vào kết quả nhận được.

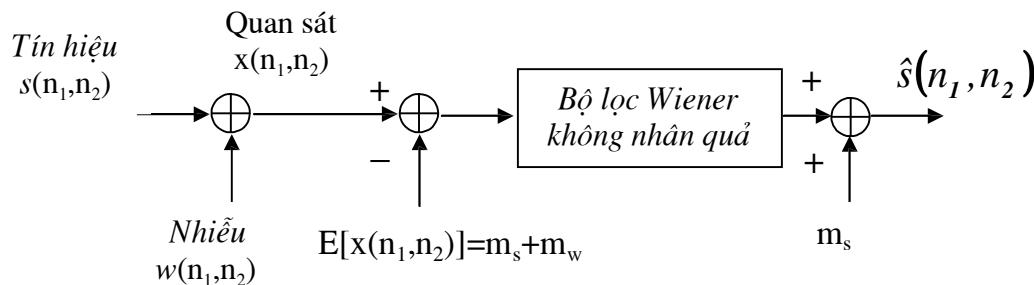
Hình 6.3 minh họa quy trình này.

## 5. UỐC LUỢNG THÔNG SỐ THỐNG KÊ

Giả sử  $\theta$  là vectơ cột của những thông số mà ta muốn đánh giá và  $x$  là vectơ cột của những lần quan sát. Ta giả sử có sự ánh xạ xác suất giữa  $\theta$  và  $x$ , với một điểm  $\theta$  trong không gian thông số ánh xạ vào một điểm  $x$  trong không gian quan sát. Bài toán ước

lượng thông số là ước lượng  $\theta$  từ quan sát  $x$  bằng một quy tắc ước lượng nào đó. Điều này khác một cách cơ bản với bài toán ước lượng hệ số bộ lọc. Trong bài toán thiết kế bộ lọc, đáp ứng xung  $h(n_1, n_2)$  là một hàm xác định của những thông số ước lượng và không có vấn đề ánh xạ xác suất giữa các thông số  $\theta$  và quan sát  $x$ .

Ba quy tắc ước lượng, là quy tắc gần đúng nhất (ML), quy tắc cực đại hậu nghiệm (MAP), và quy tắc sai số quân phương cực tiểu (MMSE) có nhiều thuộc tính thú vị và đã được nhiều người nghiên cứu. Với những thông số không ngẫu nhiên, thường dùng quy tắc ML. Khi ước lượng theo quy tắc ML giá trị thông số được chọn sao cho đó là những kết quả có vẻ phù hợp nhất trong lần quan sát  $x$ . Như vậy, giá trị  $\theta$  được chọn sao cho  $p_{x|\theta}(x_o|\theta_o)$  cực đại ở lần quan sát  $x$  với giá trị  $\theta$  đã chọn. Một thuộc tính quan trọng của ước lượng ML là ước lượng nhận được có phương sai cực tiểu ở một tập hợp điều kiện nào đó.



**Hình 3:** Ước lượng sai số quân phương cực tiểu tuyến tính của một tín hiệu khi các quá trình tín hiệu và nhiễu đều có trung vị bằng không.

Các quy tắc ước lượng MAP và MMSE thường dùng đối với những thông số có thể coi như biến ngẫu nhiên mà hàm mật độ tiên nghiệm đã biết. Trong quy tắc ước lượng MAP giá trị thông số được chọn sao cho mật độ tiên nghiệm  $p_{\theta|x}(\theta_o|x_o)$  là cực đại. Các quy tắc ước lượng ML và MMSE dẫn đến những ước lượng tương tự của giá trị thông số khi mật độ tiên nghiệm của các thông số trong quy tắc ước lượng MAP có thể coi là phẳng trong không gian thông số. Vì lý do đó, quy tắc ước lượng ML thường được coi như là trường hợp đặc biệt của quy tắc ước lượng MAP. Trong quy tắc ước lượng MMSE giá trị ước lượng  $\hat{\theta}$  nhận được bằng cách cực tiểu hóa sai số quân phương

$E[(\hat{\theta} - \theta)^T(\hat{\theta} - \theta)]$ . Ước lượng MMSE của  $\theta$  nhận được từ  $E[\theta | x]$ , trung vị hậu nghiệm của  $\theta$  khi cho  $x$ . Vậy là, khi cực đại của hàm mật độ hậu nghiệm  $p_{\theta|x}(\theta_o | x_o)$  trùng với giá trị trung vị của nó thì các quy tắc ước lượng MAP và MMSE đưa đến những giá trị ước lượng như nhau.

Để minh họa ba quy tắc ước lượng đó, ta hãy xét một ví dụ đơn giản. Giả sử ta gieo một đồng tiền mà xác suất sấp là  $\theta$ , ngửa là  $1 - \theta$ . Ta gieo N lần, và muốn ước lượng thông số  $\theta$  theo kết quả thực nghiệm. Ta ký hiệu kết quả thực nghiệm bằng biến ngẫu nhiên  $x$ , định nghĩa như sau:

$$x = \begin{cases} 1, & \text{khi kết quả là sấp} \\ 0, & \text{khi kết quả là ngửa.} \end{cases} \quad (70)$$

Kết quả của N lần gieo theo thứ tự được ký hiệu là  $x_1, x_2, \dots, x_N$ . Thoạt tiên ta giả thiết là  $\theta$  không phải ngẫu nhiên, và dùng quy tắc ML để ước lượng  $\theta$ .

Từ (70)  $p_{x|\theta}(x_o | \theta_o)$  tính ra

$$p_{x|\theta}(x_o | \theta_o) = \theta_o^M (1 - \theta_o)^{N-M} \quad (71)$$

trong đó

$$M = \sum_{i=1}^N x_i \quad (72)$$

và M đại biểu số lần sấp trong N lần thử. Để cực đại hóa  $p_{x|\theta}(x_o | \theta_o)$  trong (71) đối với  $\theta_o$ , ta lấy vi phân đối với  $\theta_o$  và cho kết quả bằng không, giải ra được  $\theta_o$ . Kết quả là:

$$\hat{\theta}_{ML} = \frac{M}{N} = \frac{\sum_{i=1}^N x_i}{N} \quad (73)$$

trong đó  $\hat{\theta}_{ML}$  là ước lượng của  $\theta$ . Kết quả trong (73) là hợp lý. Ta đếm số lần sấp và chia cho số lần thử.

Bây giờ ta giả thiết  $\theta$  là ngẫu nhiên và hàm mật độ xác suất tiên thiêng  $p_\theta(\theta_o)$  như sau:

$$p(\theta) = \begin{cases} 2\theta, & 0 \leq \theta \leq 1 \\ 0, & \text{các trường hợp khác.} \end{cases} \quad (74)$$

Để dùng quy tắc ước lượng ML, ta xác định  $p_{\theta|x}(\theta_o|x_o)$ . Từ (13) ta nhận được:

$$p_{\theta|x}(\theta_o/x_o) = \frac{p_{x/\theta}(x_o/\theta_o)p_\theta(\theta_o)}{p_x(x_o)}. \quad (75)$$

Vì  $p_x(x_o)$  trong (75) không phụ thuộc  $\theta$ , nên cực đại hoá  $p_{\theta|x}(\theta_o|x_o)$  đối với  $\theta_o$  tương đương với cực đại hoá  $p_{x|\theta}(x_o|\theta_o)p_\theta(\theta_o)$  đối với  $\theta_o$ . Từ (71), (74) và (75) cực đại hoá  $p_{\theta|x}(\theta_o|x_o)$  đối với  $\theta_o$  dẫn đến:

$$\hat{\theta}_{MAP} = \frac{M+1}{N+1} = \frac{\left[ \sum_{i=1}^N x_i \right] + 1}{N+1}. \quad (76)$$

Khi  $M = 1$  và  $N = 2$  (hai lần thử thì một lần sấp)  $\hat{\theta}_{ML}$  là  $1/2$  trong khi  $\hat{\theta}_{MAP}$  là  $2/3$ . Lý do  $\hat{\theta}_{MAP}$  lớn hơn là vì  $p_\theta(\theta_o)$  trong (74) đã giả định xác suất cao hơn cho những giá trị  $\theta$  lớn. Nếu  $p_\theta(\theta_o)$  là hằng số trong phạm vi  $0 \leq \theta_o \leq 1$  thì  $\hat{\theta}_{MAP}$  đồng nhất với  $\hat{\theta}_{ML}$ .

Ước lượng sai số quan phương cực tiểu  $\hat{\theta}_{MMSE}$  của  $\theta$  tính theo:

$$\hat{\theta}_{MMSE} = E[\theta|x] = \int_{\theta_o=0}^1 \theta_o p_{\theta|x}(\theta_o|x_o) d\theta_o. \quad (77)$$

trong đó  $p_{\theta|x}(\theta_o|x_o)$  tính theo (75). Tính (77) cho trường hợp  $M$  và  $N$  có giá trị bất kỳ có thể thực hiện được nhưng cũng hơi khó cho nên ta chỉ xét trường hợp đặc biệt  $M = 1$  và  $N = 2$ . Xác định  $\hat{\theta}_{MMSE}$  thường khó hơn là xác định  $\hat{\theta}_{ML}$  và  $\hat{\theta}_{MAP}$ . Từ (71), (74) và (75),

$$p_{\theta|x}(\theta_o/x_o) = \frac{\theta_o(1-\theta_o)2\theta_o}{p_x(x_o)}. \quad (78)$$

Để xác định  $p_x(x_o)$ , lưu ý rằng  $p_{\theta,x}(\theta_o, x_o)$  tính theo  $p_{x|\theta}(x_o|\theta_o)p_\theta(\theta_o)$  trong (13). Như vậy, ta có:

$$\begin{aligned} p_x(x_o) &= \int_{\theta_o=0}^1 p_{\theta,x}(\theta_o, x_o) d\theta_o = \int_{\theta_o=0}^1 p_{x|\theta}(x_o|\theta_o)p_\theta(\theta_o) d\theta_o \\ &= \int_{\theta_o=0}^1 \theta_o(1-\theta_o) 2\theta_o d\theta_o = \frac{1}{6}. \end{aligned} \quad (79)$$

Từ (78) và (79),

$$p_{\theta|x}(x_o|\theta_o) = 12\theta_o^2(1-\theta_o). \quad (80)$$

Từ (77) và (80)

$$\hat{\theta}_{MMSE} = \int_{\theta_o=0}^1 12\theta_o^3(1-\theta_o) d\theta_o = \frac{3}{5}. \quad (81)$$

Ước lượng  $\hat{\theta}_{MMSE}$  trong trường hợp này lớn hơn  $\hat{\theta}_{ML}$ , nhưng nhỏ hơn  $\hat{\theta}_{MAP}$ .