XỬ LÝ ẢNH SỐ:
4. Hệ màu và kỹ thuật in ấn

NỘI DUNG

- Các phương pháp biểu diễn ảnh
- Các định dạng ảnh cơ bản
- Các mô hình màu
- Kỹ thuật in ảnh

CÁC PHƯƠNG PHÁP BIỂU DIỄN ẢNH: **BITMAP** VÀ **VECTOR**

Giới thiệu

- Cấu trúc dữ liệu để lưu trữ thông tin ảnh trong bộ nhớ có ảnh hưởng rất lớn đến việc hiển thị, in ấn và xử lý.
- Tiêu chí biểu diễn ảnh:
  - Tiết kiệm bộ nhớ
  - Giảm thời gian xử lý

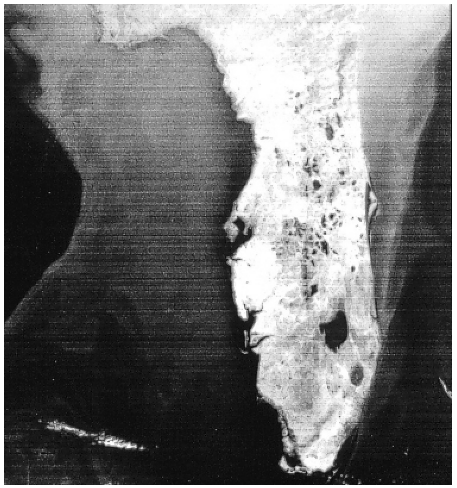
Giới thiệu

- Dữ liệu không gian có thể biểu diễn:

(1) các *đặc điểm của thế giới thực có đường biên rời rạc* (như đường xá, tòa nhà, hồ, sông, ranh giới hành chính)

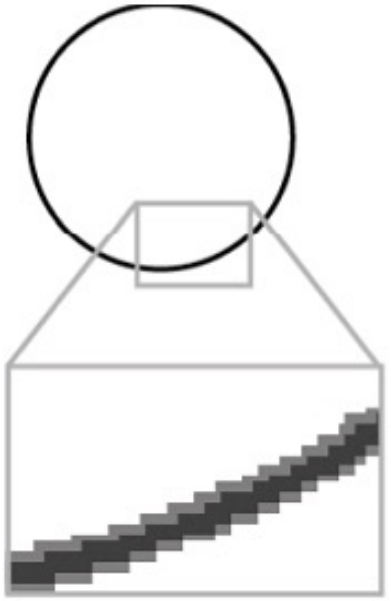
(2) các *hiện tượng của thế giới thực có sự thay đổi liên tục* (chẳng hạn như lượng mưa, nhiệt độ và mức độ dinh dưỡng, địa hình).

- Nói chung, *mô hình dữ liệu kiểu vector* được sử dụng để biểu diễn loại thứ nhất và *mô hình dữ liệu kiểu raster* để biểu diễn loại thứ 2.

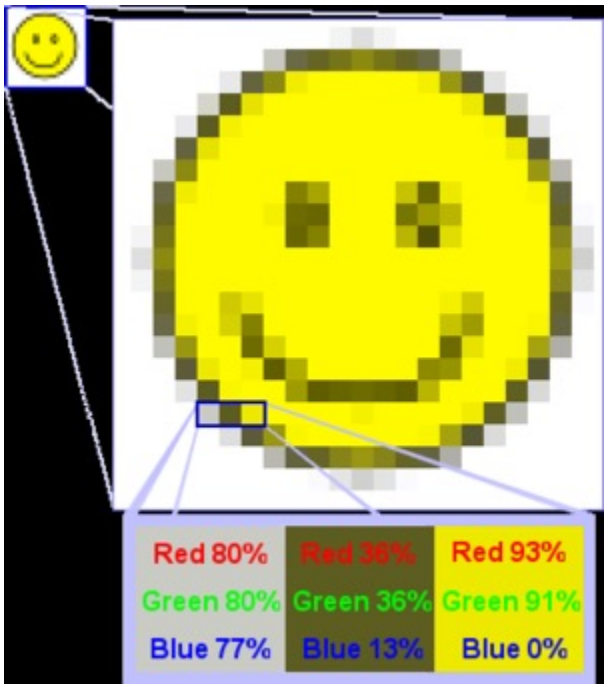


Ảnh Bitmap (1)

- Trong một ảnh bitmap (hay raster), dữ liệu hình ảnh là một ma trận các điểm màu hay pixel.
- Khi phóng đại trên một màn hình máy tính, hình ảnh bitmap trông giống như khảm.
- Càng nhiều các điểm ảnh, chi tiết hình ảnh sẽ càng sắc nét.



Enlarged bitmap image



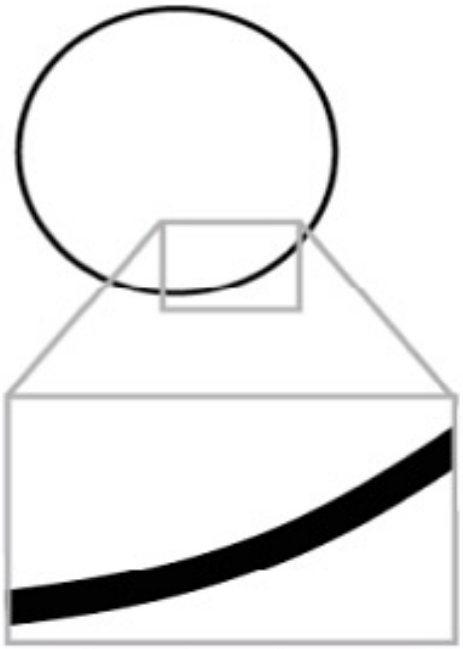
Ảnh Bitmap (2)

- Bởi vì hình ảnh có mức độ chi tiết cao với một loạt các tông và màu sắc nên biểu diễn dưới dạng bitmap là phù hợp nhất.
- Scanner và máy ảnh kỹ thuật số tạo ra hình ảnh bitmap và Photoshop là một chương trình bitmap.
- **Các định dạng bitmap bao gồm:**

  PSD, TIFF, JPEG, PNG, GIF, PICT, BMP

Vector graphics (1)

- Đồ họa vector tạo ra hình ảnh như một tập hợp của đường nét, hình elip, hình tam giác, đa giác...
- Đồ họa vector sử dụng các công thức toán học để mô tả các phác thảo và điền đầy các đối tượng ảnh.
- Đồ họa vector có thể phóng to hoặc thu nhỏ mà không bị mất mát dữ liệu và không làm thay đổi chất lượng ảnh.

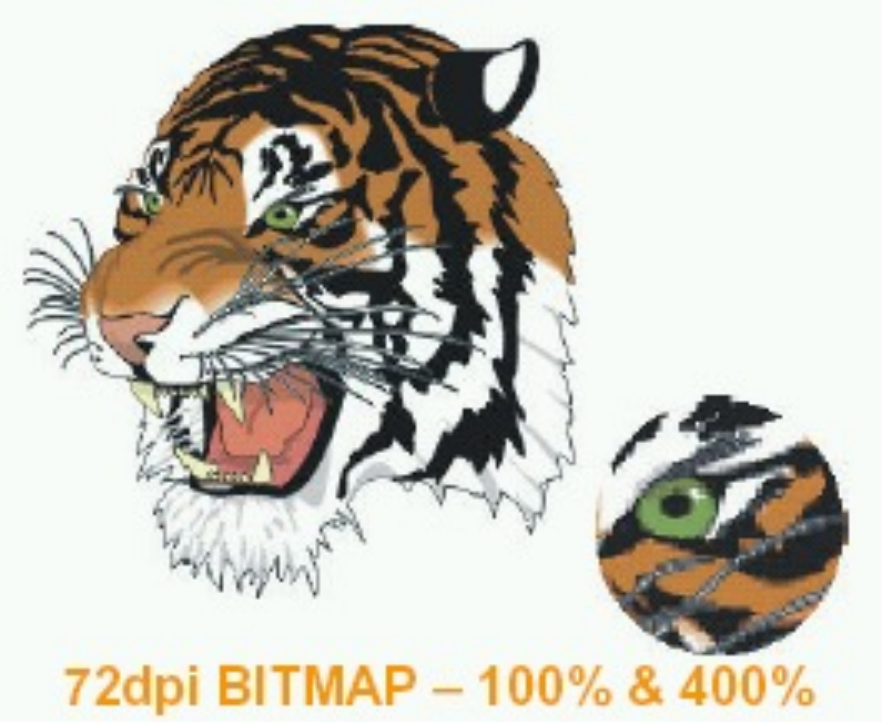


Enlarged vector graphic

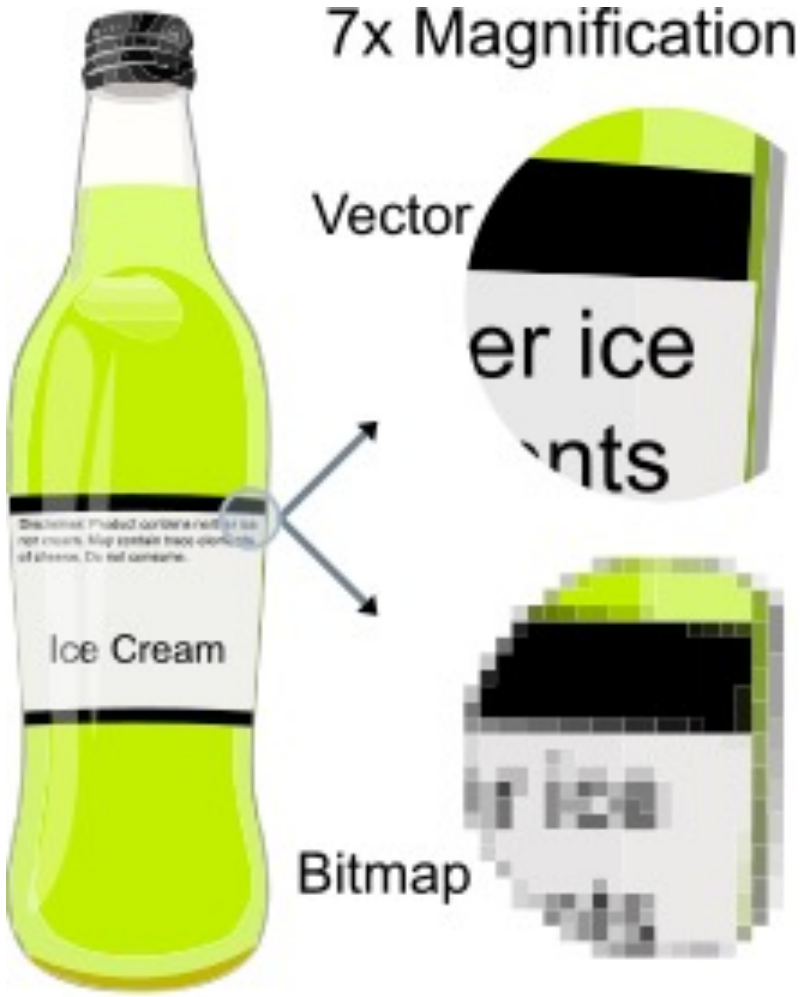
Vector graphics (2)

- CorelDraw, Illustrator, Freehand, AutoCAD & Flash tạo ra vector graphics
- Đồ họa vector là sự lựa chọn tốt nhất khi vẽ các bản vẽ, logo, kiểu chữ... và các hình ảnh không có mô hình phức tạp hay một dải rộng các tông màu thay đổi liên tục.
- **Các định dạng file kiểu vector gồm: EPS, AI, SVG, DXF, DWG**

Bitmap và Vector



Bitmap và Vector



Bitmap vs Vector

- File dạng vector phù hợp với các mô tả yêu cầu các phép đo chính xác.

- File dạng bitmap phù hợp với các ảnh photo-realistic (ảnh thực tế), loại ảnh có sự thay đổi màu sắc phức tạp.

CÁC ĐỊNH DẠNG ẢNH CƠ BẢN

GIỚI THIỆU

Các định dạng file ảnh thông dụng nhất, quan trọng nhất đối với camera, in ấn, scanning và internet là:

- JPG
- TIFF
- PNG
- GIF

Nén không tổn thất và có tổn thất

- Nén không tổn thất nghĩa là không bị mất mát thông tin.
  - Ví dụ: thuật toán nén không tổn thất tìm ra các mẫu lặp đi lặp lại trong file và thay thế các mẫu đó bằng các ký hiệu ngắn hơn và do đó giảm được kích thước file.
- Nén có tổn thất chấp nhận sự giảm cấp của ảnh để đạt được kích thước ảnh nhỏ hơn
  - Thuật toán nén có tổn thất có thể lưu giữ thông tin màu ở độ phân giải nhỏ hơn bản thân của bức ảnh và vì vậy mắt không thể nhận ra sự thay đổi của màu sắc ở khoảng cách nhỏ.

Số màu

- Ảnh đơn giản nhất có thể chỉ chứa hai màu (ví dụ trắng và đen) và sẽ chỉ cần 1 bit để biểu diễn cho mỗi pixel.
- Ảnh phức tạp hơn có thể sử dụng 24 bit để biểu diễn cho 1 pixel và vì vậy có khả năng hiển thị 2^{24} hay 16 triệu màu.
- Vì mắt khó có thể phân biệt các màu sắc tương tự nhau nên 24 bit/ 1 pixel hay 16 triệu màu thường được gọi là màu thực (truecolor).

TIFF

- TIFF là định dạng ảnh không tổn thất. Thực tế TIFF thường được sử dụng như định dạng ảnh không nén.
- Hầu hết các chương trình đồ họa sử dụng TIFF không nén. Do đó kích thước file khá lớn (đôi khi TIFF sử dụng thuật toán nén không tổn thất LZW nhưng nó không được hỗ trợ rộng rãi).

JPG

- Máy ảnh số và các trang web thông thường sử dụng file JPG vì JPG nén có tỉ lệ nén rất cao (kích thước file nhỏ hơn rất nhiều). Để đạt được điều này, JPG sử dụng các thuật toán nén có tổn thất.
- Kích thước file càng nhỏ chất lượng ảnh càng giảm và ngược lại. JPG cho phép lựa chọn giữa chất lượng cao và kích thước file lớn với chất lượng kém và kích thước file nhỏ.

GIF

- GIF tạo một bảng tối đa 256 màu trong 16 triệu màu.
- Nếu một bức ảnh có ít hơn 256 màu, GIF có thể tạo ra bức ảnh một cách chính xác. Nếu bức ảnh có nhiều màu, phần mềm có thể tạo ra ảnh GIF sử dụng các thuật toán để xấp xỉ số màu trong ảnh với bảng màu giới hạn 256 màu.

GIF

- GIF đạt được khả năng nén theo 2 cách:
  - Giảm số màu trong bức ảnh nhiều màu và vì thế giảm số bit cần cho mỗi pixel.
  - Nó thay thế các mẫu lặp lại thường xuyên (đặc biệt là một vùng lớn đồng màu) bằng cách viết tắt: “white white white white white ”→ “5 white”.
- GIF là không tổn thất đối với các ảnh có ít hơn 256 màu, còn đối với các bức ảnh màu chân thực nó có thể mất 99,998% màu.

PNG

- PNG cũng là một định dạng không tổn thất.
- Nó tìm kiếm các mẫu trong ảnh để có thể sử dụng để nén file.
- Qúa trình nén là thuận nghịch, do đó có thể khôi phục bức ảnh một cách chính xác.
- So với TIFF, PNG có dung lượng nhỏ hơn một chút nhưng thời gian đọc và ghi chậm hơn.

Kết luận

- GIF và JPG là các định dạng được sử dụng cho hầu hết các ảnh web.
- PNG được hỗ trợ bởi hầu hết các trình duyệt thế hệ mới nhất.
- TIFF không được sự hỗ trợ rộng rãi của các trình duyệt web và vì vậy ít được sử dụng cho web.
- PNG có thể làm được mọi điều như GIF và thậm chí tốt hơn, vì vậy trong tương lai PNG có thể thay thế cho GIF.
- PNG không thể thay thế cho JPG vì JPG có khả năng nén ảnh tốt hơn nhiều.

CÁC CÁCH SỬ DỤNG

	Photographic Images	Graphics, including Logos or Line art
Đặc điểm	Các bức ảnh có tông màu liên tục, 24 bit màu hoặc 8bit đối với ảnh xám, không có chữ viết, ít đường nét và biên.	Đồ họa thường là các màu đơn với số lượng màu ít, tối đa là 256 màu, thường có chữ viết hoặc đường nét, biên sắc nét.
Chất lượng tốt nhất	TIFF hoặc PNG (nén không tổn thất và không có nhiễu JPG)	TIFF hoặc PNG (nén không tổn thất và không có nhiễu JPG)
Kích thước file nhỏ nhất	JPG với hệ số chất lượng phù hợp.	TIFF LZW or GIF or PNG (đồ họa/logos không có gradient thường sử dụng 2-16 màu để đạt được kích thước file nhỏ nhất)
Độ tương thích tối đa. (PC, Mac, Unix)	TIFF or JPG	TIFF or GIF

CÁC MÔ HÌNH MÀU

Cảm nhận màu sắc

- Mắt người cảm nhận màu sắc dựa vào tế bào hình nón (cones) và tế bào hình que (rods) trên hoàng điểm.
- Tế bào hình que chủ yếu cảm nhận về độ chói (luminance).
- Tế bào hình nón chịu trách nhiệm chính về màu sắc. Có 3 loại tế bào hình nón cảm biến chính tương ứng với màu đỏ (red), xanh lá cây (green) và xanh da trời (blue) với tỷ lệ 65%, 33% và 2% (nhưng tế bào nón xanh da trời là loại nhạy cảm nhất).

- Do đặc tính hấp thụ của mắt người, màu được coi như sự kết hợp của các màu sắc cơ bản là Red (R), Green (G) và Blue (B).
- Ba màu cơ bản này có thể cộng với nhau để tạo ra các màu thứ cấp của ánh sáng là:

Magenta (đỏ thẫm) = red + blue

Cyan (xanh nhạt hay lục lam) = green + blue

Yellow (vàng) = red + green

RGB values of the same three colors

MÔ HÌNH MÀU (1)

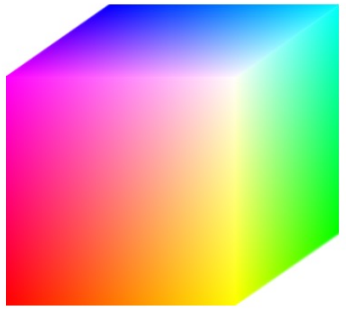
- Một mô hình màu (hay hệ màu, không gian màu) là một hệ thống có quy tắc để tạo ra tất cả các màu sắc từ một nhóm nhỏ các màu cơ bản.
- Các mô hình màu hiện nay tập trung hướng vào phần cứng (ví dụ phần cứng có chức năng thu nhận và hiển thị hình ảnh) hoặc các ứng dụng ở đó mục đích là thao tác trên màu.

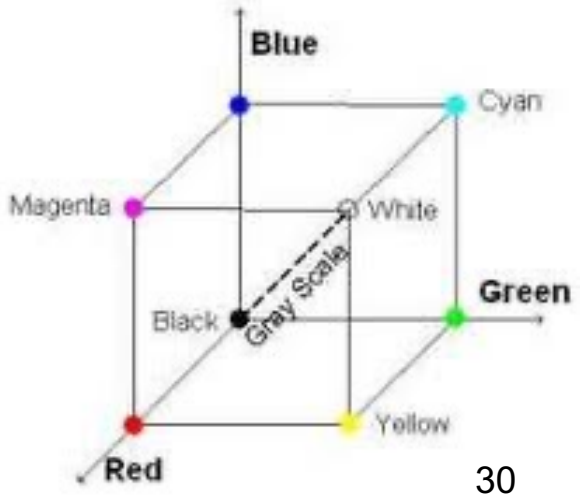
MÔ HÌNH MÀU (2)

- Mô hình theo hướng phần cứng được sử dụng nhiều nhất trong thực tế là mô hình RGB (red, green, blue) cho các màn hình màu và camera video màu.
- CMY (cyan, magenta, yellow) và CMYK (Cyan, magenta, yellow, black) cho in màu.
- HSV (Hue, saturation, value) và HSL (hue, saturation, lightness) gần tương ứng với cách con người mô tả và biên dịch màu sắc nên phù hợp với các hệ thống phần mềm hỗ trợ biên soạn ảnh.

RGB

- Trong hệ màu RGB, các màu sắc đều được tạo ra từ 3 màu cơ bản là red, green và blue.
- 3 màu RGB nằm ở 3 góc, 3 màu thứ cấp CMY nằm ở 3 góc đối diện. Màu đen ở gốc tọa độ và màu trắng ở góc xa gốc nhất.
- Các màu khác nhau trong mô hình này là các điểm nằm trên hoặc trong hình lập phương và được xác định bởi vector tính từ gốc tọa độ.



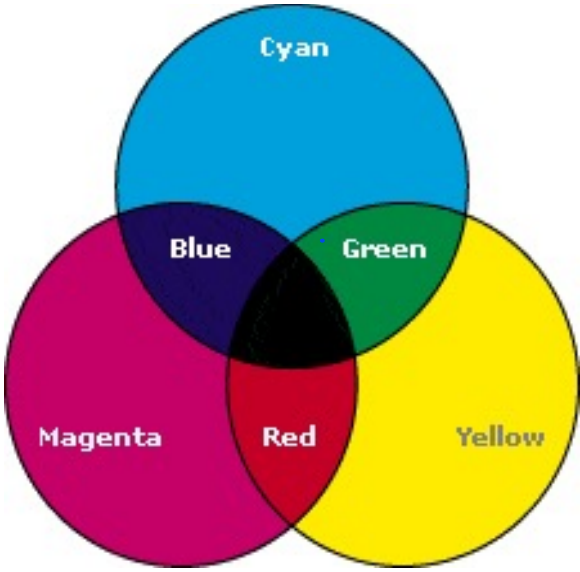


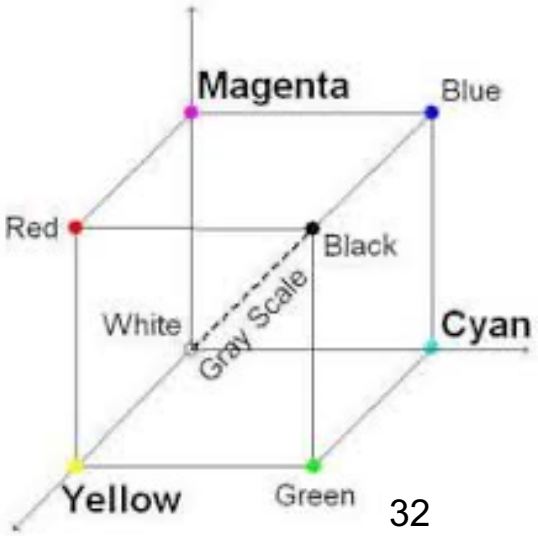
Đặc điểm của hệ màu RGB

- Mô hình màu cộng
- Dùng cho hiển thị máy tính, TV, camera số ..
- Sử dụng ánh sáng để hiển thị màu.
- Màu sắc là kết quả của ánh sáng truyền qua.
- Red + Green + Blue = White.
- Bởi vì hệ màu cộng hiển thị màu như là kết quả của ánh sáng truyền qua (cộng vào) nên khi không có ánh sáng sẽ được coi là màu đen.
- Các giá trị màu là một vector gồm 3 thành phần R, G, B có giá trị nằm trong khoảng từ 0 đến 1.

CMY

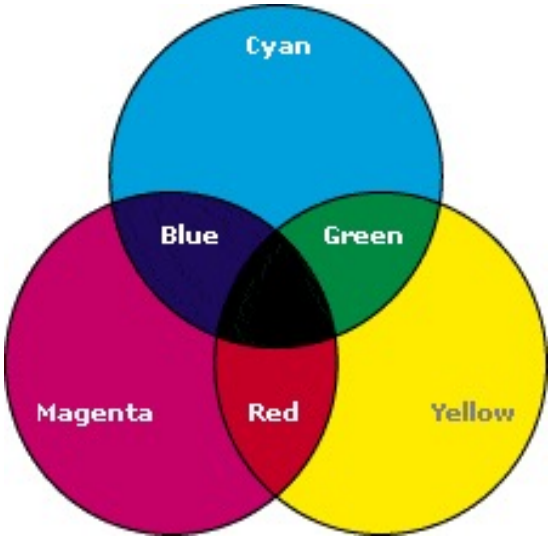
- Hệ màu trừ
- Sử dụng trong in ấn.
- Sử dụng mực để hiển thị màu
- Màu là kết quả của ánh sáng bị phản xạ (không bị hấp thụ)
- Cyan + Magenta + Yellow = Black
- Hệ màu trừ hiển thị màu như là kết quả của ánh sáng bị hấp thụ (bị trừ) bởi mực in nên khi thêm vào càng nhiều mực in thì ánh sáng phản xạ càng ít. Khi không có mực in thì ánh sáng bị phản xạ (từ bề mặt trắng) sẽ được coi là màu trắng.

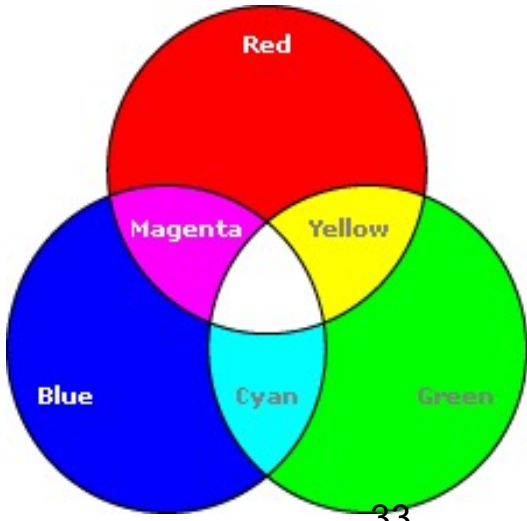




CMY

- Khi một bề mặt phủ màu cyan được rọi bởi ánh sáng trắng, sẽ không có ánh sáng màu đỏ phản chiếu từ bề mặt. Đó là bởi vì cyan đã triệt tiêu phần màu đỏ phản xạ khi có tia sáng trắng mà bản chất là tổng của 3 màu RGB.
- Cyan= 1 – red = blue + green





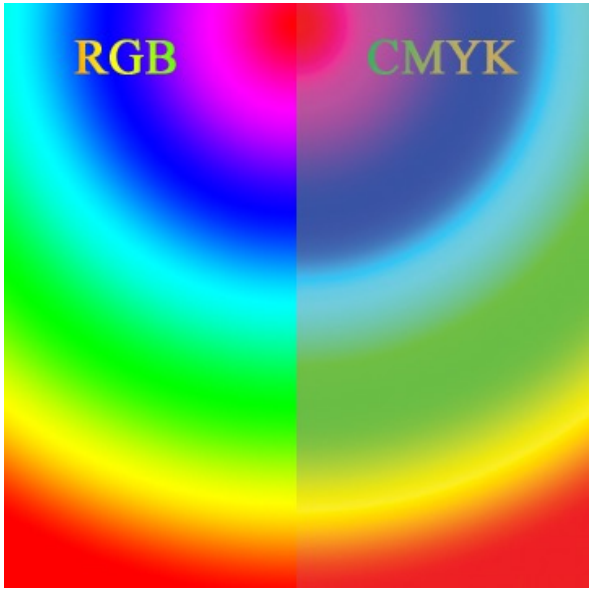
RGB và CMY

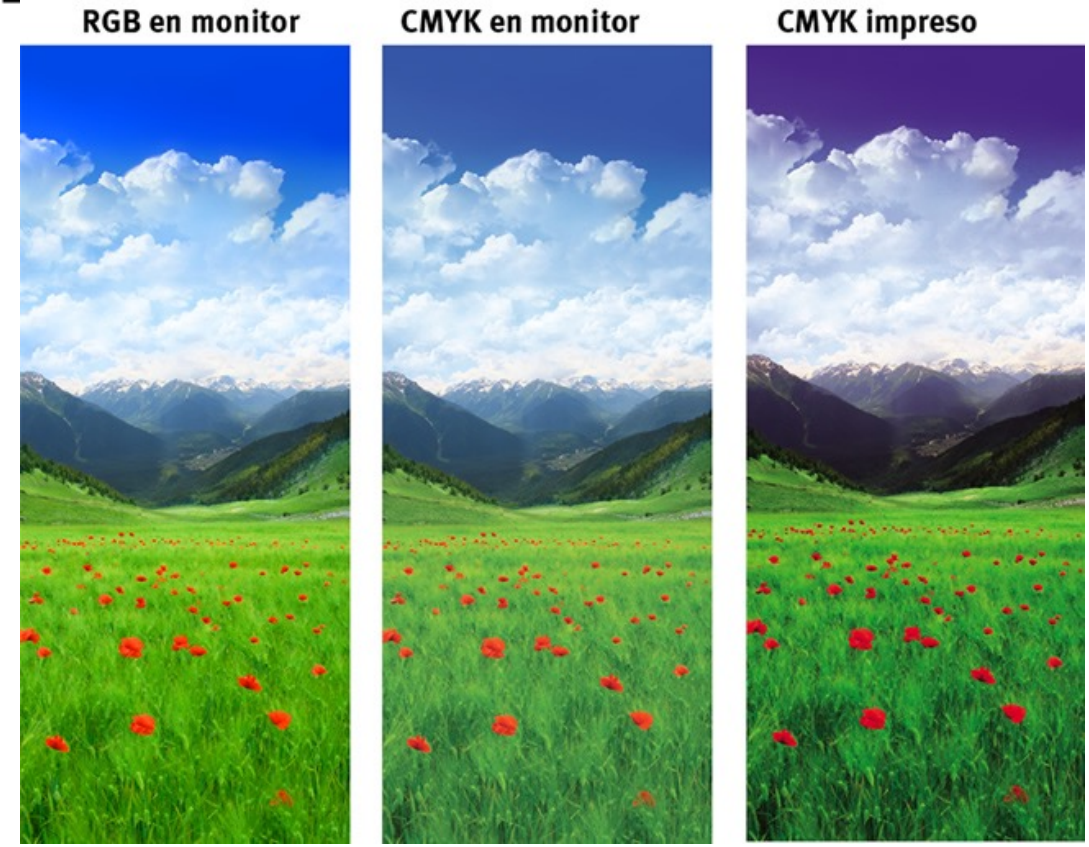
- Sự khác biệt nằm ở mục đích sử dụng.
- Với các ứng dụng liên quan đến trình chiếu trên các màn hình thì lựa chọn RGB. RGB sẽ làm việc tốt với các thiết bị phát quang sử dụng ánh sáng trắng làm cơ sở.
- Để phục vụ cho in ấn thì CMY (CMYK) là lựa chọn tối ưu.

RGB và CMY

- Chuyển đổi giữa hệ màu RGB và CMY:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$





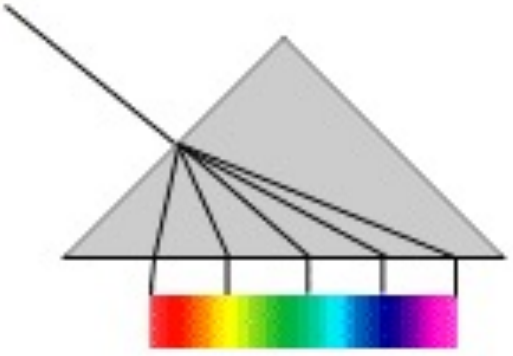
HSV (Hue, Saturation, Value)

- Ngoài RGB, CMY còn có thể mô tả màu bằng các thuộc tính khác:
  - Hue (độ màu)
  - Saturation (độ bão hòa)
  - Value (giá trị độ sáng)

HUE (Độ màu)

- Định nghĩa –một màu cụ thể trong dải màu của mô hình màu, mô tả độ chân thực của màu sắc.
- Được xác định bởi độ dài bước sóng xác định.
- Là cái mà hầu hết chúng ta nghĩ đến khi nói “màu sắc”.

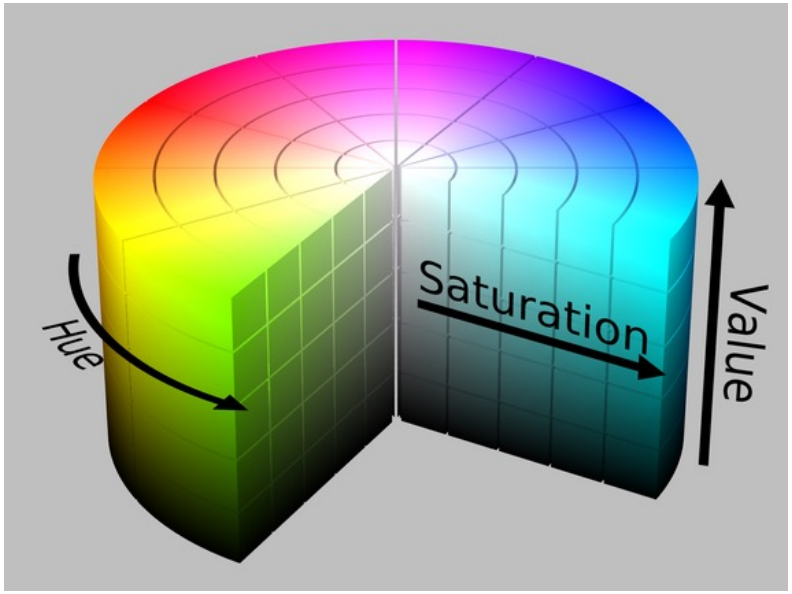
- ROY G. BIV =
  - Red
  - Orange
  - Yellow
  - Green
  - Blue
  - Indigo
  - Violet



Saturation (Độ bão hòa)

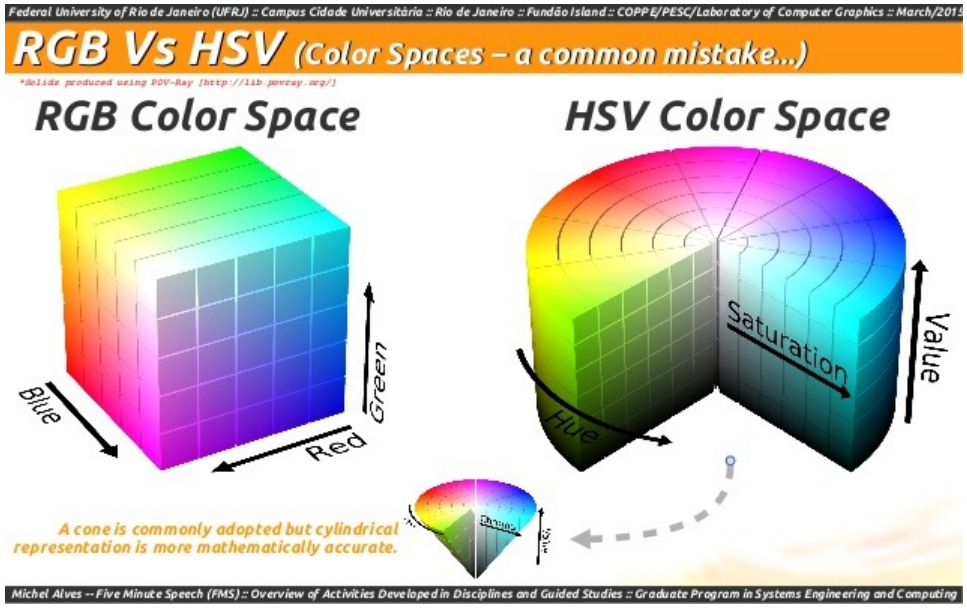
- Định nghĩa:
  - "là lượng ánh sáng trắng được trộn với hue (độ màu)"
  - đại diện cho lượng màu; mức độ trội của độ màu.

- Độ bão hòa là thước đo độ chân thực bị làm loãng đi hay bị giảm bớt đi bởi sự tác động của ánh sáng trắng.



Value (Độ sáng)

- Định nghĩa: biểu diễn mức độ sáng tối của màu.
- Cũng thường gọi là độ sáng “brightness” hay cường độ “intensity”.
- Đây là khái niệm chủ quan khó có thể đo đạc được, dùng để mô tả cường độ màu sắc cảm nhận bởi mắt người.



HSV

- Mô hình này tách biệt giá trị cường độ ra khỏi các thông tin chứa màu sắc(độ màu và độ bão hòa) trong ảnh màu.

- HSV trở thành công cụ lý tưởng cho xử lý ảnh dựa trên sự mô tả màu sắc một cách tự nhiên và trực quan nhất với con người.

- HSV thường được sử dụng trong các phần mềm biên soạn ảnh, hoặc các công cụ trình diễn dữ liệu như bản đồ hoặc ảnh y tế.

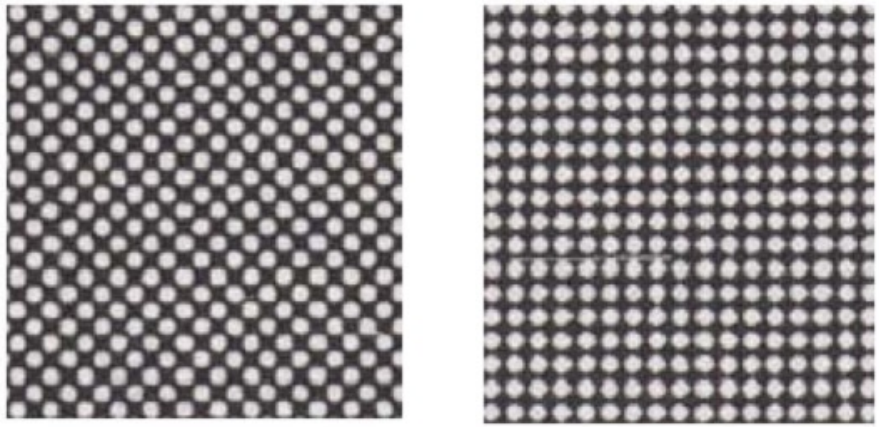
KỸ THUẬT IN ẢNH
Kỹ thuật nửa cường độ số
(Digital Halftoning)

Digital Halftoning: Giới thiệu

- Kỹ thuật nửa cường độ là một quá trình mô phỏng các sắc thái của màu xám bằng cách thay đổi kích thước của chấm đen nhỏ sắp xếp theo một mô hình chung.
- Kỹ thuật này được sử dụng trong máy in, cũng như các ngành công nghiệp xuất bản.
- Điều này thực hiện được do sự tổng hợp không gian được thực hiện bởi đôi mắt. Đôi mắt sẽ pha trộn các chi tiết sắc nét và ghi lại cường độ tổng thể
- Cũng được áp dụng cho ảnh màu.

Các tham số của halftoning

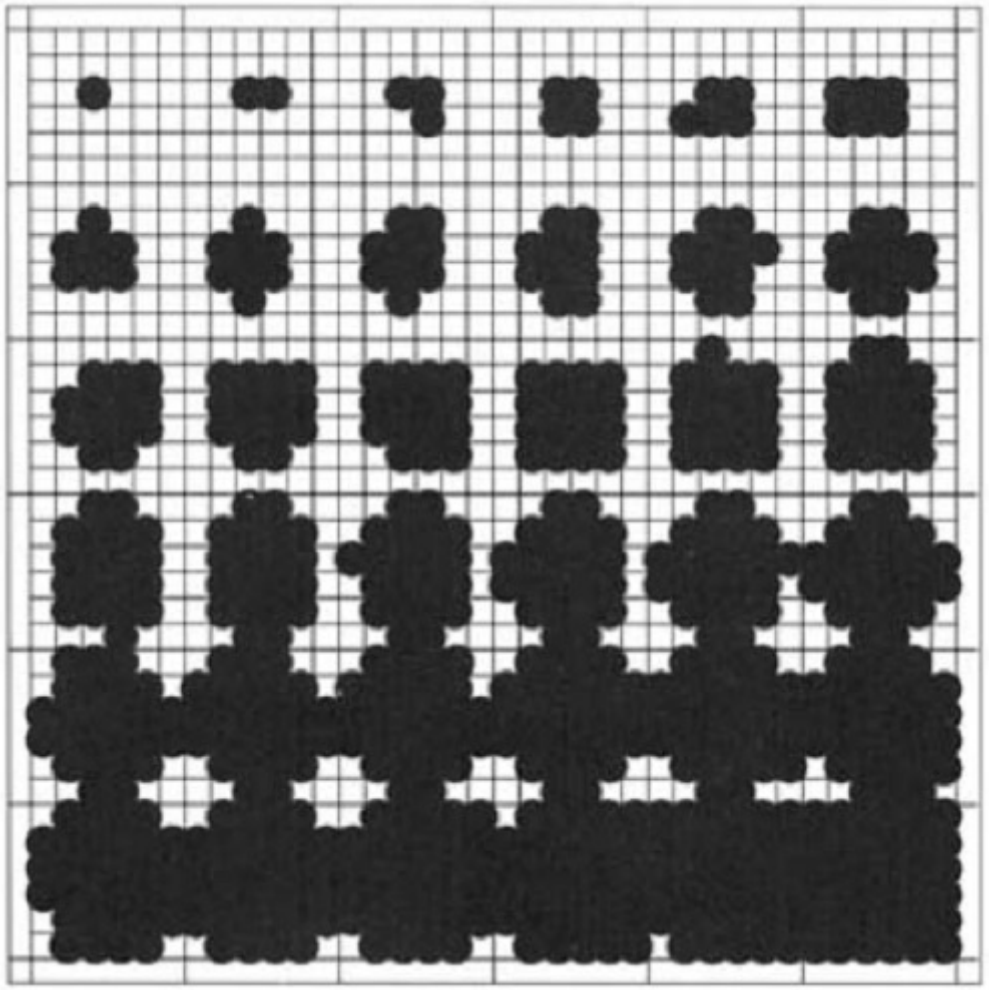
- Hình dạng của thành phần lưới:
  - Hình vuông, hình tròn, hình elip, dấu thập, dấu chéo, tam giác, đường thẳng.
- Mật độ đường thẳng trên lưới, được xác định bằng lines per inch (lpi).
  - Báo giấy: 85 lpi, tạp chí: 200 – 300 lpi
- Góc màn hình: góc phổ biến nhất là 45^0



45° 90°

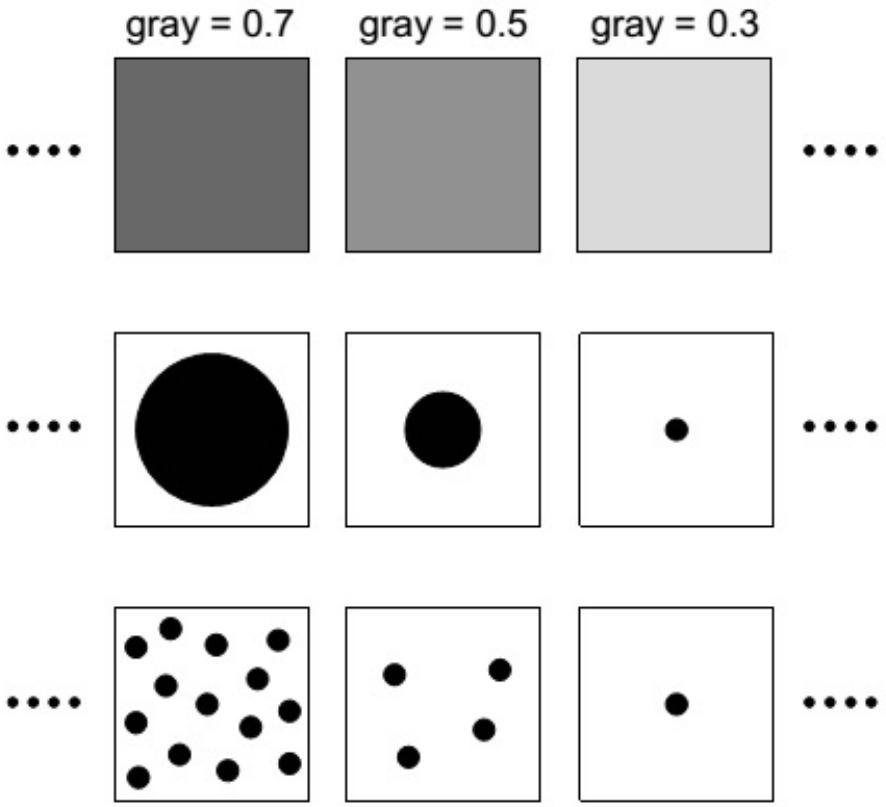
Ví dụ halftoning

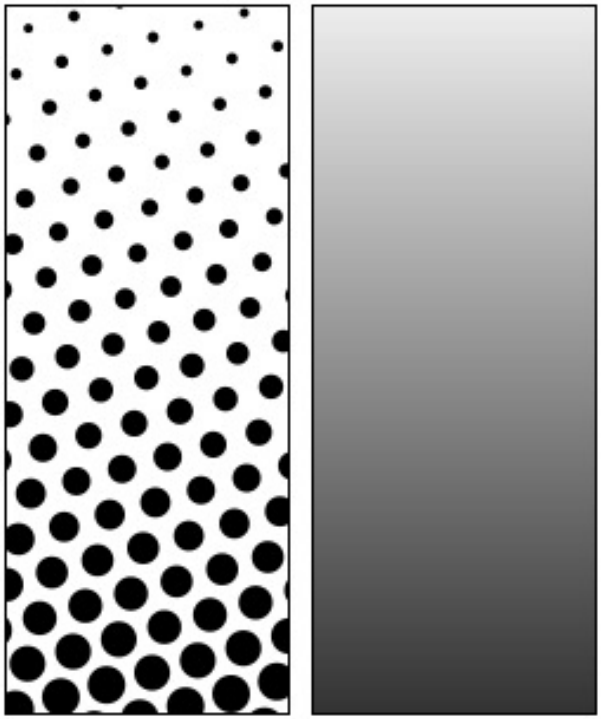
- Trường hợp lưới 6 x 6.
- Có thể hiện thị 37 mức sáng.



Ví dụ halftoning

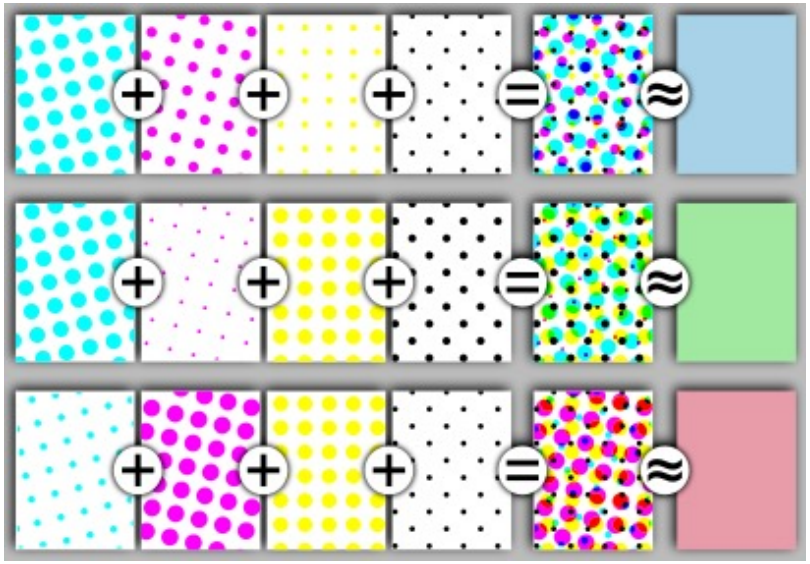
- Mức độ bao phủ của mực trong vùng sẽ xác định mức xám.

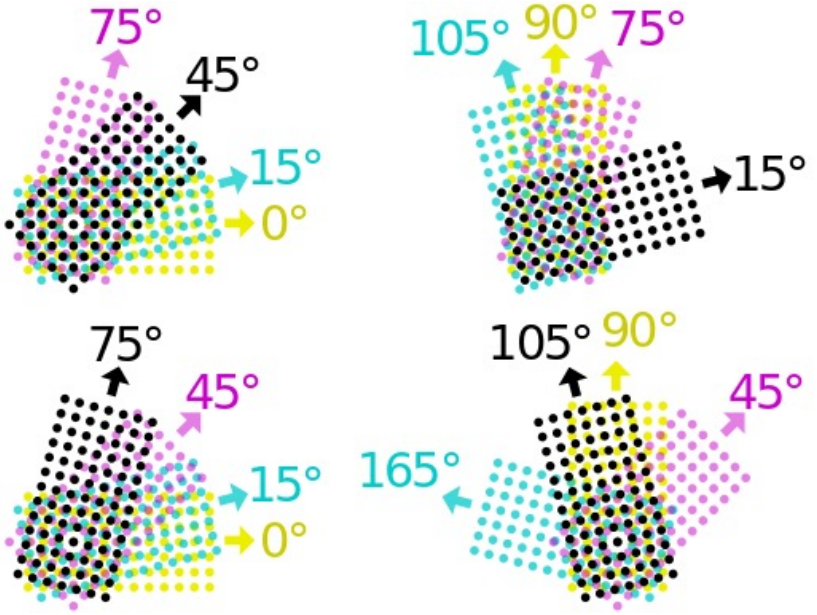




Ví dụ

- Dưới đây là các ví dụ về kỹ thuật nửa cường độ đối với ảnh màu với sự phân tách CMYK.





Các kỹ thuật nửa cường độ số

- Thresholding (phân ngưỡng)
- Patterning (chọn mẫu)
- Dithering (phối màu)
- Error diffusion (Khuếch tán lỗi)

Kỹ thuật phân ngưỡng (Thresholding)

- Kỹ thuật đơn giản nhất để cải thiện độ phân giải hình ảnh là sử dụng một mẫu ngưỡng.
- Ta tạo ra mẫu này và so sánh với ảnh gốc. Nếu pixel trong ảnh gốc lớn hơn pixel tương ứng trong mẫu ngưỡng thì pixel này được thay bằng 1, ngược lại thay bằng 0.

GrayScale

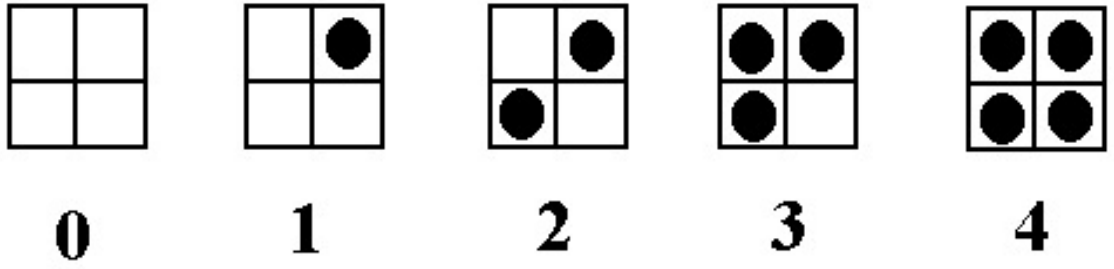
Threshold

Kỹ thuật chọn theo mẫu (Patterning)

- Được sử dụng khi thiết bị đầu ra có độ phân giải không gian cao hơn ảnh gốc.
- Độ phân giải hình ảnh của các ảnh được tạo ra bằng máy tính có thể tăng lên nhờ sử dụng kỹ thuật này.
- Trong kỹ thuật này, thực hiện chọn một nhóm các mẫu (patterns) để tạo ra nhiều mức cường độ xám hơn.

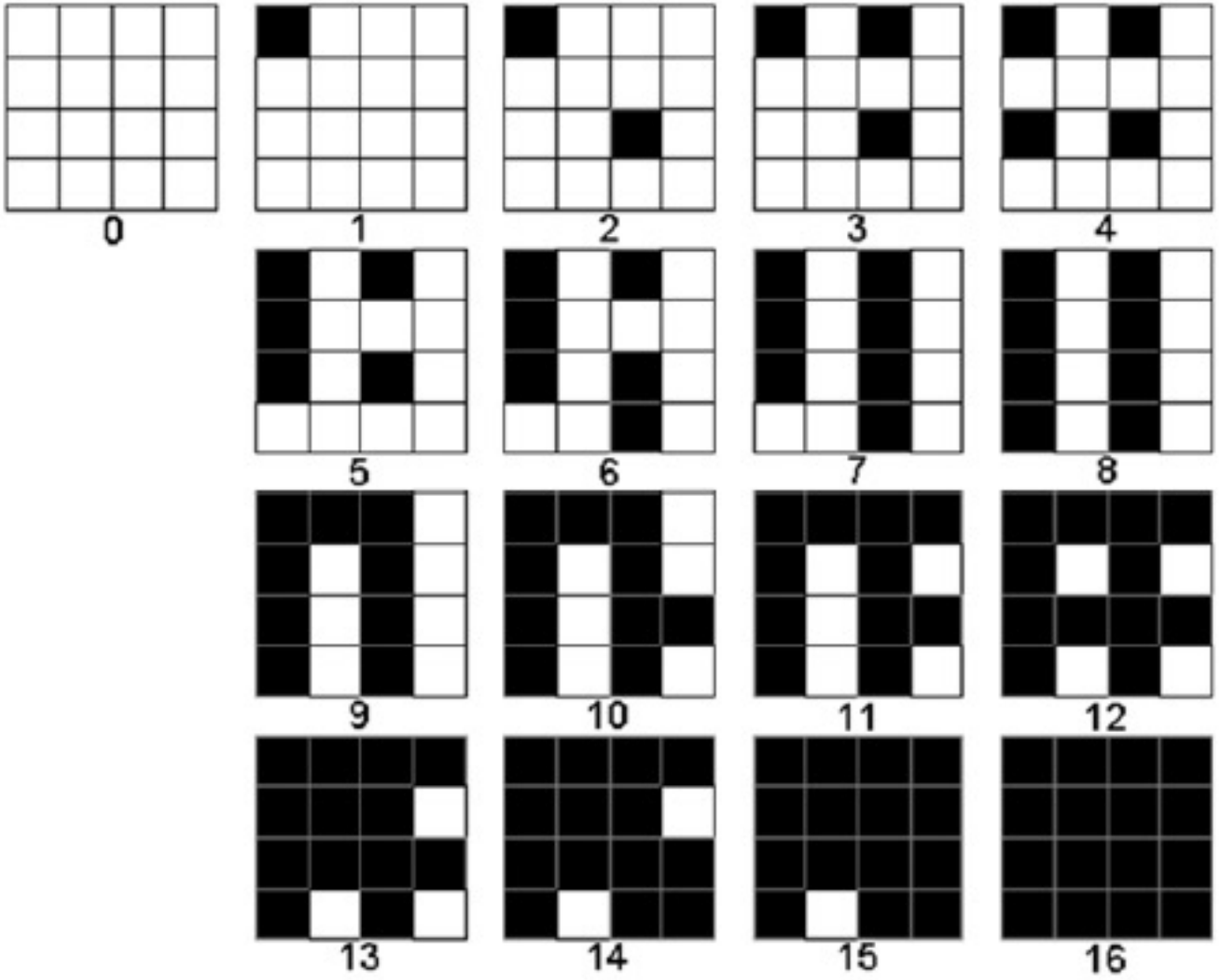
Patterning

- Với mỗi cell gồm 4 pixel, cách sử dụng này tạo ra 5 mức xám.

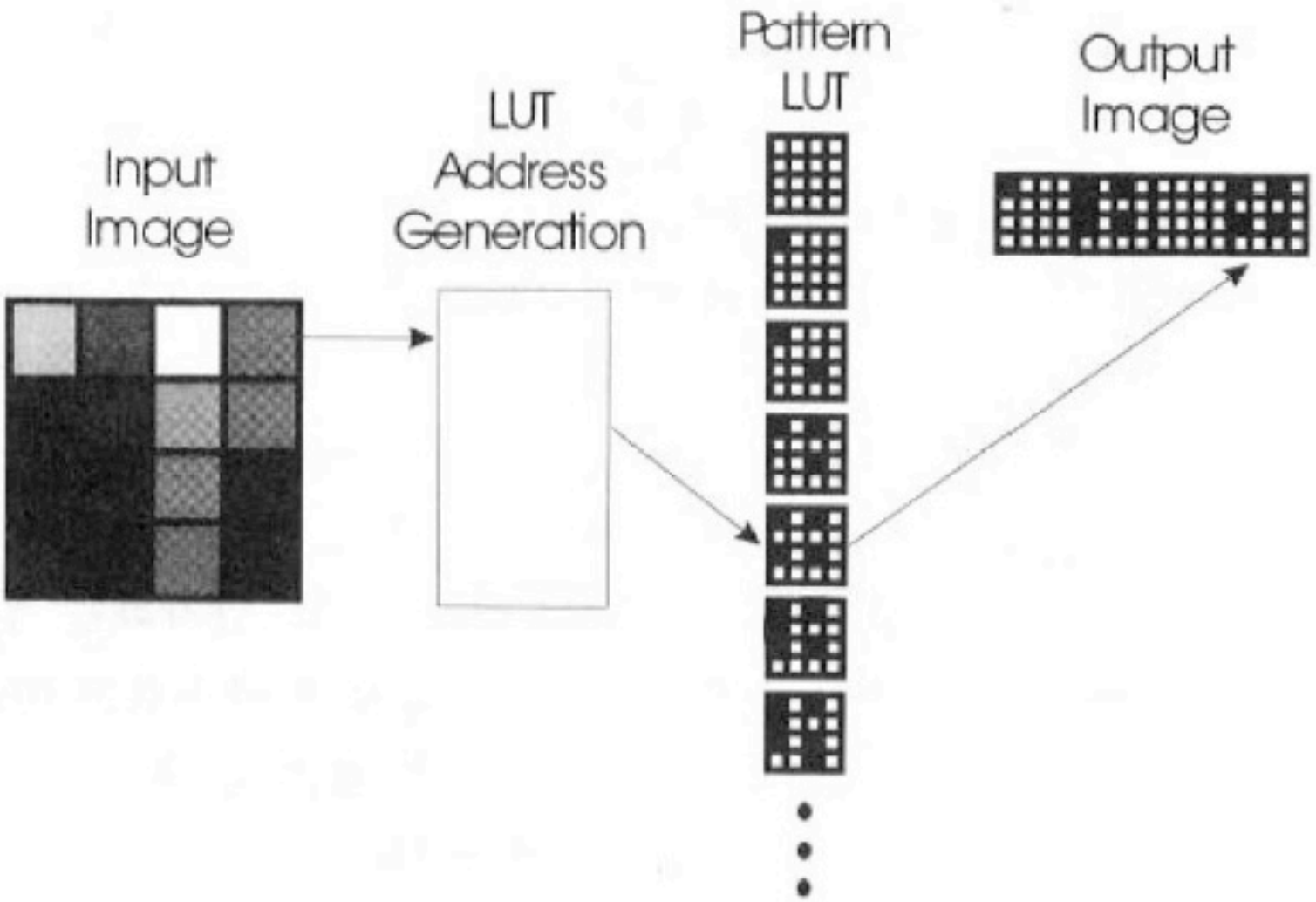


- Thông thường sử dụng ma trận chọn mẫu hồi quy của Rylander:
  - Khối n x n tạo ra n^2+1 mức xám.
  - Yêu cầu lượng tử hóa pixel đầu vào.
  - Ví dụ khối 4 x 4 tạo ra 17 mức cường độ. Do đó phải chia pixel đầu vào cho 15 (15 x 17 = 255). Khi đó sẽ tạo ra các pixel có giá trị từ 0 đến 17.

Ma trận chọn mẫu hồi quy của Rylander

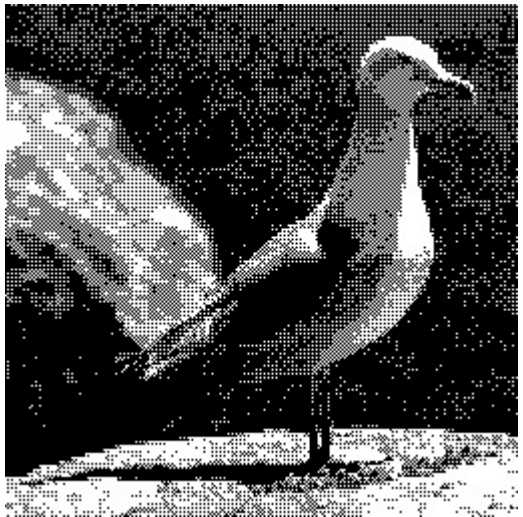


Qúa trình chọn mẫu



Nhận xét

- Việc chọn kích thước của nhóm như vậy sẽ làm giảm độ mịn của ảnh.
- Vì vậy kỹ thuật này chỉ áp dụng trong trường hợp mà độ phân giải của thiết bị ra lớn hơn độ phân giải của ảnh nguồn.
- Ví dụ thiết bị ra có độ phân giải 256x256 và sử dụng ma trận 2 x 2, ta phải sử dụng ảnh gốc 128x128.



Kỹ thuật phối màu (Dithering)

- Không giống như pattern, dithering tạo ra *hình ảnh đầu ra với cùng số lượng điểm ảnh trong hình ảnh đầu vào*.
- Dithering có thể được coi là phân ngưỡng ảnh với một ma trận phối màu.
- Kỹ thuật này tạo ra ảnh đa cấp sáng khi độ phân giải nguồn và đích là như nhau.
- Để tạo ảnh, mỗi phần tử của ảnh gốc sẽ được so sánh với phần tử tương ứng của ma trận Dither. Nếu lớn hơn, phần tử ở đầu ra sẽ sáng và ngược lại.

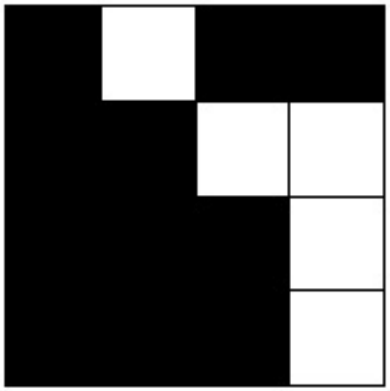
Ví dụ

12	51	34	121
78	254	10	97
45	113	110	16
90	200	206	34

input image

0	60	0	60
45	110	45	110
0	60	0	60
45	110	45	110

repeated dither matrix



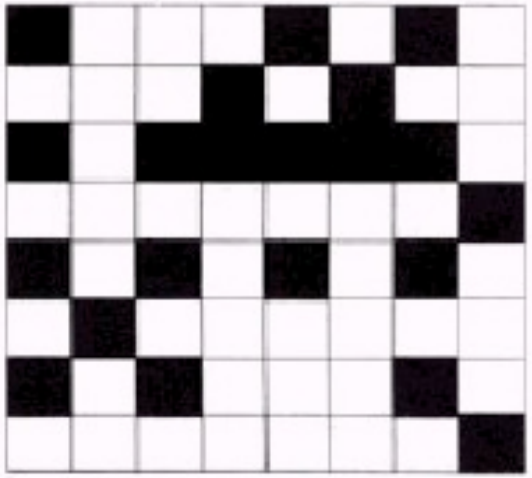
output image

12	51	14	31	16	50	60	70
30	65	23	100	13	99	79	83
77	65	199	203	202	200	85	99
66	58	43	11	15	65	89	91
87	81	64	24	98	56	80	19
41	98	31	30	18	16	9	0
78	43	51	61	45	59	87	15
64	53	41	43	61	13	71	115

Input Image Segment

0	128	32	160	0	128	32	160
192	64	224	96	192	64	224	96
48	176	16	144	48	176	16	144
240	112	208	80	240	112	208	80
0	128	32	160	0	128	32	160
192	64	224	96	192	64	224	96
48	176	16	144	48	176	16	144
240	112	208	80	240	112	208	80

Tiled Dither Matrices

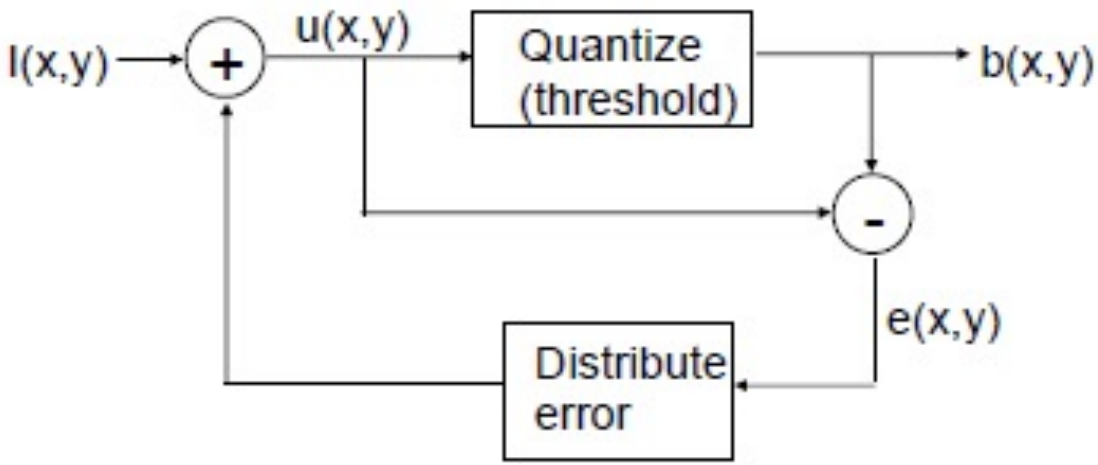


Bi-level Output Image Segment

Khuếch tán lỗi (error diffusion)

- ✓ Khuếch tán lỗi cho phép giảm thiểu mức độ mất chi tiết của ảnh khi tách ngưỡng bằng cách phân tán lỗi do lượng tử hóa ra các điểm ảnh xung quanh (bên phải và bên dưới pixel hiện thời).
- ✓ Bằng cách này tổng giá trị điểm ảnh của một vùng nhỏ được giữ tương đối gần với giá trị trên ảnh gốc.
- ✓ Lưu ý: Khuếch tán lỗi là toán tử láng giềng trong khi chọn mẫu và phối màu là toán tử điểm (pixel).

Mô tả quá trình khuếch tán lỗi



Khuếch tán lỗi một chiều (1D)

- Ảnh được duyệt từ trái qua phải, từ trên xuống dưới.
- Tại mỗi điểm ảnh, giá trị điểm ảnh được tách theo ngưỡng có sẵn.
- Phần dư do lượng tử hóa được được chuyển sang điểm ảnh tiếp theo trên cùng dòng.
- Các bước được lặp lại cho đến hết dòng, phần dư của điểm ảnh cuối cùng sẽ được loại bỏ.

Ví dụ

I =	0.7	0.7	0.3	0.5	0.1	0.1	0.1

I(1) = u(1) = 0.7 → threshold at 0.5 → b(1) = 1

e(1) = b(1) - u(1) = 0.3

Since pixel I(1) was over represented, compensate by subtracting error from next pixel I(2)

u(2) = I(2) - e(1) = 0.4

u(2) = 0.4 → threshold at 0.5 → b(1) = 0

e(2) = b(2) - u(2) = -0.4

u(3) = I(3) - e(2) = 0.7

and so on....

b =	1	0	1	0	0	0	1

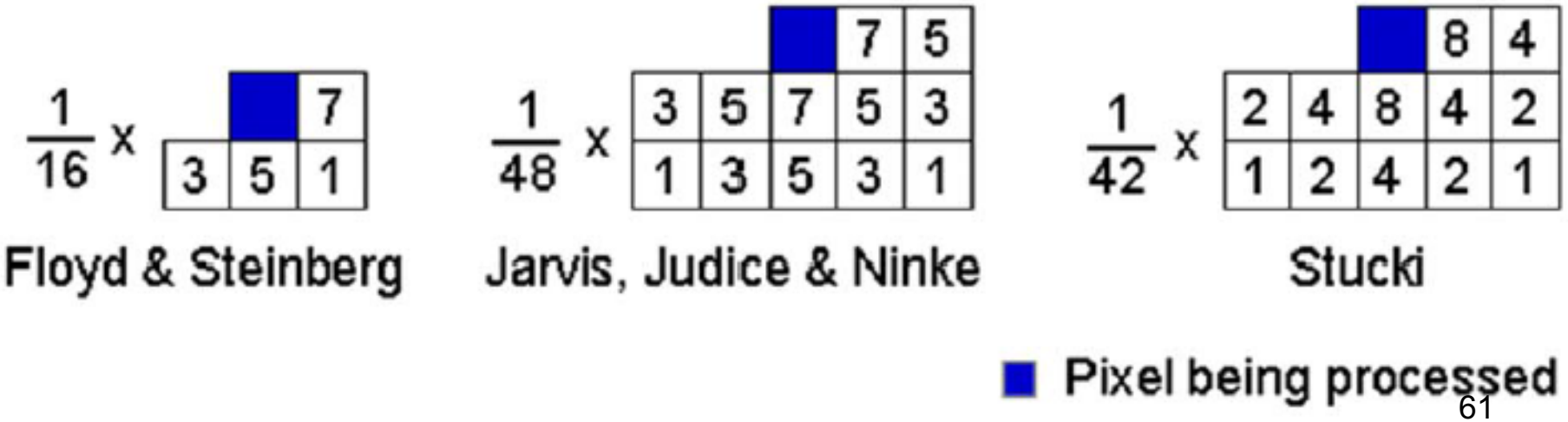
Bài tập

- Thực hiện khuếch tán lỗi một chiều với ảnh I sử dụng ngưỡng 127, được biết ảnh này là ảnh 256 mức xám với mức nhỏ nhất là 0 và lớn nhất là 255.
-

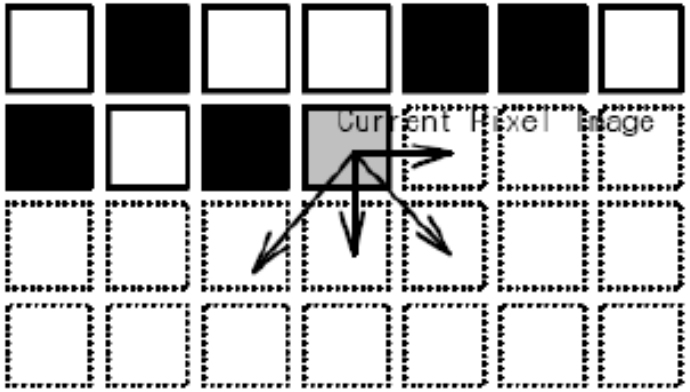
$$I = \begin{bmatrix} 1 & 13 & 156 & 22 & 45 \\ 133 & 13 & 12 & 12 & 232 \\ 12 & 222 & 127 & 32 & 21 \end{bmatrix}$$

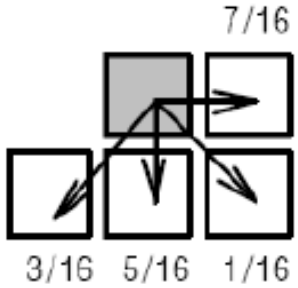
Khuếch tán lỗi hai chiều (2D)

- Các bước được thực hiện như khuếch tán lỗi một chiều, tuy nhiên lỗi do lượng tử hóa sẽ đươc phân tán ra các điểm xung quanh (tối thiểu là 4 pixel) theo tỷ lệ xác định bởi ma trận khuếch tán.
- Một số ma trận khuếch tán phổ biến:



Floyd & Steinberg






```
FOR Y=0 TO HEIGHT
    FOR X=0 TO WIDTH
        IF INPUT_IMAGE[X][Y] < THRESHOLD
            OUTPUT_IMAGE[X][Y]=0
            ERROR = INPUT_IMAGE[X][Y]
        ELSE
            OUTPUT_IMAGE[X][Y]=1
            ERROR = INPUT_IMAGE[X][Y]-MAX_PIXEL
        INPUT_IMAGE[X+1][Y]=INPUT_IMAGE[X+1][Y]+ERROR*7/16
        INPUT_IMAGE[X-1][Y+1]=INPUT_IMAGE[X+1][Y]+ERROR*3/16
        INPUT_IMAGE[X][Y+1]=INPUT_IMAGE[X+1][Y]+ERROR*5/16
        INPUT_IMAGE[X+1][Y+1]=INPUT_IMAGE[X+1][Y]+ERROR*1/16
```

NHẬN XÉT

- Các kỹ thuật phân ngưỡng, chọn mẫu hay phối màu đều là các toán tử pixel nên tính toán nhanh nhưng cho kết quả chưa được tốt.
- Kỹ thuật khuếch tán lỗi cho kết quả tốt nhất nhưng là toán tử láng giềng nên cần các tính toán chuyên sâu hơn.

NỘI DUNG ĐÃ HỌC

- Các phương pháp biểu diễn ảnh
- Các định dạng ảnh cơ bản
- Các mô hình màu
- Kỹ thuật in ảnh.