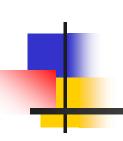
TRƯỜNG ĐẠI HỌC THỦY LỢI Khoa Công nghệ Thông tin Bộ môn KHMT



ĐỒ HỌA MÁY TÍNH (Computer Graphics)

Ngô Trường Giang

E-mail: giangnt@tlu.edu.vn

Nội dung

- Tổng quan đồ họa máy tính
- Màu và phối màu
- Thuật toán cơ sở vẽ đồ họa
- Các kỹ thuật trong đồ họa 2D
- Phép biến đổi đồ họa 2D
- Phép biến đổi đồ họa 3D
- Quan sát đồ họa 3D
- Mô hình hóa bề mặt



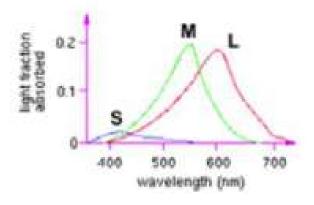
MÀU VÀ PHỐI MÀU

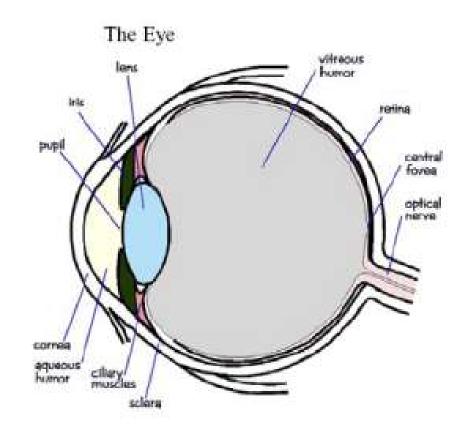
- Khái niệm màu
- Các mô hình màu
- Fading
- Hiệu chỉnh gamma
- Dithering và haftoning

Có sử dụng slides của PGS.TS Đặng Văn Đức – Viện CNTT, Viện HLKH&CN VN

Cảm nhận màu

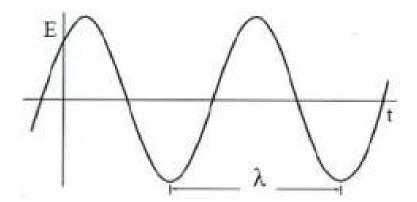
- Phần nhạy cảm với ảnh: võng mạc (retina)
 - retina bao gồm hai loại tế bào: rod và cone
 - cone có trách nhiệm nhận biết màu.
- Cones có ba loại: S, M, L tương ứng với cảm biến B (430 nm), G (560 nm), R (610 nm)





Màu là gì?

- Có nhiều định nghĩa về màu (không có định nghĩa hình thức)
 - Từ góc nhìn khoa học:
 - Màu là phân bổ các bước sóng λ (red: 700 nm, violet: 400 nm)
 - Và tần số f
 - Tốc độ ánh sáng: c=λf



- Từ góc nhìn về nghệ thuật và cuộc sống:
 - Màu là Hue, Brightness, Saturation của ánh sáng
 - Sắc, độ sáng và bão hòa của đối tượng

Mô hình màu

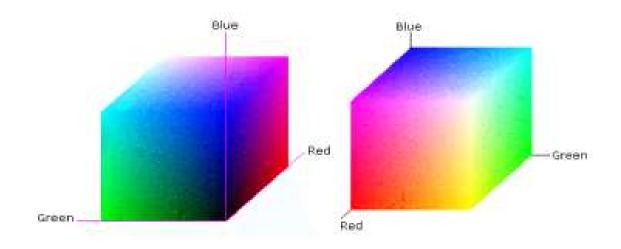
- Là phương pháp diễn giải các đặc tính và tác động của màu trong ngữ cảnh nhất định.
- Không có mô hình màu nào là đầy đủ cho mọi khía cạnh của màu
 - Sử dụng các mô hình màu khác nhau để mô tả các tính chất được nhận biết khác nhau của màu.

Thí dụ

- Mô hình màu RGB: ánh sáng Red, Green và Blue ứng dụng cho màn hình, TV.
- Mô hình HSV: Nhận thức của con người
- Mô hình CMYK: Máy in

Mô hình màu RGB

- Mô hình màu RGB được biểu diễn bởi lập phương với các trục R, G, B
 - Gốc biểu diễn màu đen
 - Tọa độ (1, 1, 1) biểu diễn màu trắng.
 - Tọa độ trên các cạnh trục biểu diễn các màu cơ sở.
 - Các cạnh còn lại biểu diễn màu bù cho mỗi màu cơ sở



Mô hình màu RGB

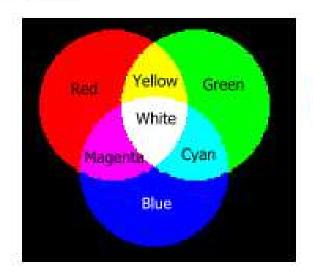
- Biểu đồ RGB thuộc mô hình cộng:
 - Phát sinh màu mới bằng cách cộng cường độ màu cơ sở
- Gán giá trị từ 0 đến 1 cho R, G, B
 - Red+Blue -> Magenta (1, 0, 1)
 - Đường chéo từ (0, 0, 0) đến (1, 1, 1) biểu diễn màu xám
- Nhận xét
 - Mô hình này không thể biểu diễn mọi màu trong phổ nhìn thấy
 - Đủ cho các ứng dụng máy tính
 - Màn hình máy tính và TV sử dụng mô hình này
 - Được sử dụng rộng rãi nhất
 - Đơn giản

Mô hình màu CMYK

- Với màn hình: màu là tổ hợp các ánh sáng phản xạ từ phosphor.
- Với giấy: phủ chất màu lên giấy, mắt ta nhận biết ánh sáng phản xạ sau khi chất màu đã hấp thụ.
 - Mực viết chữ màu đen có nghĩa rằng mực đã hấp thụ toàn bộ ánh sáng nhìn thấy trên nó
 - Những dòng chữ này có màu green vì mực hấp thụ toàn bộ bước sóng trừ bước sóng tương ứng với màu green. Ánh sáng còn lại phản xạ vào mắt ta.

Mô hình màu CMYK

- Mô hình màu xác định bởi các màu cơ sở cyan, magenta và yellow dành cho máy in màu. Mô hình CMY (bên phải) là bù của mô hình RGB (bên trái)
- Biểu đô CMY thuộc loại mô hình trừ.
- CMY CMYK





Chuyển đổi giữa CIE và RGB

```
void CIE2RGB(float x,float y,float z,float &R,float &G,float &B)
 float x1,z1;
 x1 = x*z/y;
 z1 = (1 - x - y)*z/y;
 R = 2.739*x1 - 1.145*z - 0.424*z1;
 G = -1.119*x1 + 2.029*z + 0.033*z1;
 B = 0.138*x1 - 0.333*z + 1.105*z1;
void RGB2CIE(float R,float G,float B,float &x,float &y,float &z)
 float x1,z1,sum;
 x1 = 0.478*R + 0.299*G + 0.175*B;
 z = 0.263*R + 0.655*G + 0.081*B;
 z1 = 0.020*R + 0.160*G + 0.908*B;
 sum = x1 + z + z1;
 x = x1/sum;
 y = z/sum;
```

Chuyển đổi giữa RGB và CMY

RGB -> CMY

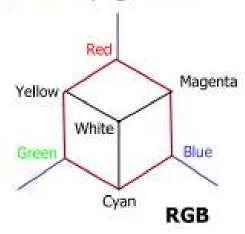
CMYK -> RGB

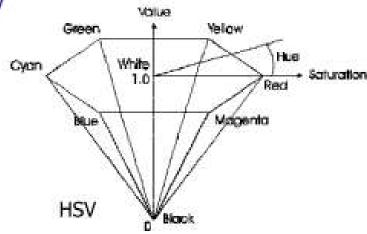
```
void RGB2CMYK(float R,float G,float B,float &C,float &M,float &Y,float &K)
{
   RGB2CMY(R, G, B, C, M, Y);
   K = min3(C, M, Y); // Cho lại giá trị min từ ba đổi số
   C = C - K;
   M = M - K;
   Y = Y - K;
}
```

Mô hình màu HSV

- Thay vì chọn các phần tử RGB để có màu mong muốn, người ta chọn các tham số màu: Hue, Saturation và Value (HSV)
- Mô hình HSV suy diễn từ mô hình RGB
 - Hãy quan sát hình hộp RGB theo đường chéo từ White đến Black (gốc) -> ta có hình lục giác

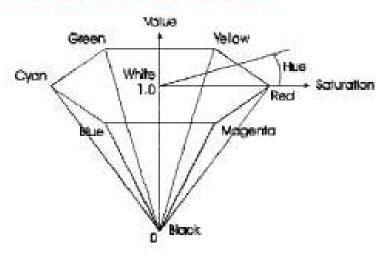
Sử dụng làm đỉnh hình nón HSV





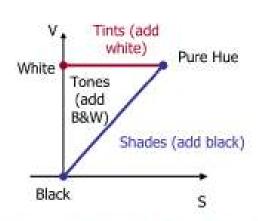
Mô hình màu HSV

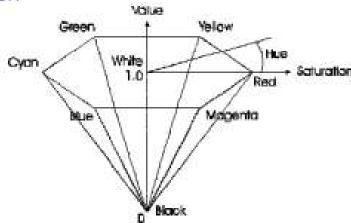
- Hue: Bước sóng gốc của ánh sáng
 - Trong mô hình Hue được biểu diễn bằng góc từ 0° đến 360°
- Value: Cường độ hay độ chói ánh sáng
 - Value có giá trị [0, 1]
 - V=0 -> màu đen. Đỉnh lục giác có cường độ màu cực đại
- Saturation: Thước đo độ tinh khiết ánh sáng gốc
 - S trong khoảng [0, 1]
 - Biểu diễn tỷ lệ độ tinh khiết của màu sẽ chọn với độ tinh khiết cực đại



Mô hình màu HSV

- Mô hình HSV trực giác hơn mô hình RGB
 - Bắt đầu từ Hue (H cho trước và V=1, S=1)
 - Thay đổi S: Bổ sung hay bớt trắng
 - Thay đổi V: Bổ sung hay bớt đen
 - Cho đến khi có màu mong muốn





 Mắt người có thể phân biệt 128 Hues, 130 tints và cực đại 30 shades (Yellow):

128 x 130 x 30 = 382 720 màu khác nhau

Mô hình màu YIQ

- Mô hình YIQ được áp dụng cho National Television
 Standards Committee (NTSC) để phát sóng TV
 - Måt người nhạy cảm với độ sáng (Brightness Luminance) của ảnh hơn là thay đổi Hue và Saturation (Chrominance)
 - Tách phần tử Y để chứa thông tin độ chói (Luminance)
 - Phần tử I (in-phase) và Q (quadrature) chứa thông tin về màu (Chromaticity)
- YIQ là nền tảng của nén ảnh JPEG
- Chuyển đổi từ RGB sang YIQ

$$Y = 0.299R + 0.587G + 0.114B$$

 $I = 0.596R - 0.275G - 0.321B$
 $Q = 0.212R - 0.523G + 0.311B$

Mô hình màu YIQ

Chuyển đổi qua lại RGB <-> YIQ

```
void RGB2YIQ (float R, float G, float B, float &Y, float &I, float &Q)
    1
         Y=0.2999*R+0.587*G+0.114*B;
         I=0.596*R-0.274*G-0.322*B;
         Q=0.211*R-0.522*G+0.311*B;
    }
    void YIQ2RGB (float Y, float I, float Q, float &R, float &G, float &B)
    1
         R=Y+0.956*I+0.623*Q;
         G=Y-0.272*I-0.648*Q;
         B=Y-1.105*I+1.705*Q;
}
```

Mô hình màu YIQ

- YUV được sử dụng vào xây dựng chuẩn Video số
- Sử dụng quan niệm tương tự YIQ
- Chuyển đổi từ RGB sang YCbCr

$$Y = 0.299R + 0.587G + 0.114B$$

 $Cb = -0.147R - 0.289G + 0.436B$
 $Cr = 0.615R - 0.515G - 0.1B$

Fading

- Fading: Kỹ thuật làm màu tan dần
 - Trong các trò chơi điện tử
- Fading có nghĩa làm giảm độ sáng của ảnh màn hình một các trơn tru.
- Giải pháp
 - Các giá trị Palette lưu trữ trong CRTC theo mô hình RGB
 - Chuyển đổi sang giá trị YIQ
 - Giảm dân Q về 0
 - Mỗi lần giảm xong ta chuyển ngược lại RGB để thay thế palette trong CRTC

Dithering và Haftoning

Cơ sở

- Mắt người nhìn xa ~20 cm thì phân biệt được 0.003 inch
- Nếu ta quan sát một vùng nhỏ có các chẩm đen trắng ở khoảng cách xa thì mắt người coi vùng nhỏ đó là màu đồng nhất.
- Ånh B&W analog có hàng triệu mức xám
 - Khi in: sử dụng các pattern đen trắng để thay thế
 - Độ xám thấp thay thể bằng mẫu ít điểm đen hơn
- Khả năng sử dụng độ phân giải không gian để biểu diễn màu khác nhau.
- Nhóm các điểm ảnh thành cell bao gồm nhiều giá trị màu khác nhau
- Dithering là kỹ thuật làm tăng khả năng hỗ trợ màu của thiết bị
- Là phương pháp tổ hợp có hệ thống hay ngẫu nhiên các điểm ảnh có màu cơ sở để phát sinh bóng màu (color shade) hay độ xám (gray)

Dithering



Ånh gốc

Ảnh 4 mức xám



Ảnh 32 mức xám



Dithering

Mục tiêu

 Hiển thị ảnh màu hay đa mức xám trên thiết bị đen/trắng hay ít màu hơn

Giải pháp

- Tạo lập pattern thay thế các pixel ảnh
 - Lượng tử các pixel ảnh vào các mẫu
 - Thuật toán thay thế pixel bởi các pattern
- Hoặc phân bổ lỗi ngẫu nhiên trên các điểm ảnh lân cận

Dithering



Original Full-color photograph



Error diffusion



Ordered dither

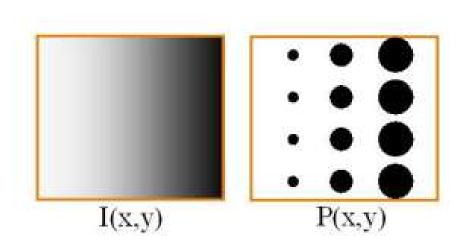
Haftoning

Là một loại của dithering:

- Haftoning xấp xỉ ảnh màu hay ảnh xám liên tục bằng ma trận phối màu các trùm dot.
- Haftoning và Dithering là đồng nghĩa
 - Khái niệm "haftoning" thường sử dụng trong công nghiệp in
 - Khái niệm "dithering" sử dụng trong đồ họa máy tính
 - Cả hai đều sử dụng trùm pixel (tế bào haftoning hay mẫu dithering) để biểu diễn cường độ của một pixel.

Haftoning

- Sử dụng các dot có kích thước khác nhau để biểu diễn cường độ
 - Diện tích của dot tỷ lệ với cường độ ảnh





Newspaper Image

Haftoning

- Phụ thuộc vào kinh nghiệm và thực tiễn
- Sử dụng phối màu 4x4 cho các ảnh đơn giản như đô thị, biểu đồ
- Sử dụng halftoning để in ảnh từ máy scanner sẽ tự nhiên hơn
- Với cả hai halftoning và dithering: sử dụng ma trận lớn hơn 5x5 sẽ không cho kết quả tốt
 - Làm giảm mật độ màu, làm tăng mật độ không gian
- Không nên sử dụng ma trận nhỏ hơn 3x2, nó không có ý nghĩa
- Khi in ảnh ở mật độ cao: nên sử dụng halftoning thay cho dithering.
- Thuật toán Floyd-Steinberg cho mật độ ảnh tốt nhất. Kết quả là ảnh in đôi khi có chất lượng cao hơn ảnh chụp

Bài tập

- Viết chương trình thực hiện fading.
- Viết chương trình cho phép người sử dụng chọn các tham số màu HSV từ thực đơn, chuyển đổi màu chọn sang RGB để hiển thị.
- Có thể hiển thị bao nhiều mức xám bằng halftoning với lưới nxn pixel, mỗi pixel hiển thị m mức khác nhau.
- Tìm ma trận dither 4x4, viết chương trình hiển thị ảnh bằng ma trận vừa tìm ra.
- Giả sử thiết bị có khả năng hiển thị 8 màu cơ bản. Hãy viết chương trình sử dụng bộ lọc Floyd-Steinberg để hiển thị ảnh màu thực