

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP

TÀI LIỆU HỌC TẬP
KỸ THUẬT ĐỒ HỌA MÁY TÍNH

Nam Định – 2019

LỜI GIỚI THIỆU

Ngày nay, hình ảnh được tạo ra từ máy tính có mặt ở khắp mọi nơi. Các sản phẩm đồ họa tràn ngập cuộc sống của chúng ta và đạt được đến kỹ thuật mà đôi khi chúng ta không thể nhận ra đó là một hình ảnh nhân tạo.

Các chương trình máy tính thường hiển thị kết quả bằng đồ họa. Các chương trình truyền hình và quảng cáo sử dụng hình ảnh từ đồ họa mang lại hiệu quả rõ rệt về tính thẩm mỹ. Nhiều chương trình truyền hình và điện ảnh gần đây pha trộn các diễn viên thực và hình ảnh nhân tạo đến mức người xem có thể rất khó để phân biệt một nhân vật hoặc cảnh thật với hình ảnh do máy tính tạo ra. Ngày càng nhiều máy ảnh kỹ thuật số, thiết bị điện tử sử dụng input là màn hình cảm ứng thay vì bàn phím. Do vậy, kỹ thuật đồ họa máy tính trở thành một lĩnh vực lý thú và có nhiều ứng dụng trong thực tế.

Kỹ thuật đồ họa liên quan đến tin học và toán học vì hầu hết các giải thuật vẽ, tô màu cùng các phép biến hình đều được xây dựng dựa trên nền tảng của hình học không gian hai chiều và ba chiều.

Hiện nay, Kỹ thuật đồ họa máy tính là một môn học được giảng dạy cho sinh viên chuyên ngành công nghệ thông tin học với 30 tiết lý thuyết. Bài giảng được biên soạn theo đúng chương trình đào tạo của Nhà trường. Nội dung bài giảng tập trung vào các vấn đề chính trong 5 chương như sau:

Chương 1: Tổng quan về đồ họa máy tính, giới thiệu các khái niệm và cái nhìn tổng quan về đồ họa máy tính.

Chương 2: Đồ họa 2 chiều, trình bày các thuật toán vẽ và tô các đường cơ bản như đường thẳng, đường tròn, các phép biến đổi trên đối tượng đồ họa 2D và các thuật toán xén đoạn thẳng.

Chương 3: Đồ họa 3 chiều, Giới thiệu tổng quan về đồ họa 3 chiều, các phép biến đổi và các phép chiếu, một số cách biểu diễn đối tượng đồ họa 3 chiều từ các đối tượng đơn giản như các hình khối, các đa diện ...

Chương 4: Các hệ màu, trình bày về các hệ màu cơ bản cùng các qui tắc phối màu cơ bản trong đồ họa máy tính.

Chương 5: Các kỹ thuật đồ họa trong bộ Công cụ Adobe After Effect

1. Mục tiêu môn học:

Với các nội dung như trên, sau khi học xong môn này, sinh viên cần đạt được các yêu cầu sau:

- Hiểu thế nào là đồ họa trên máy tính.
- Thiết kế và cài đặt được các thuật toán vẽ các đường cơ bản như đường thẳng, đường tròn, ...
- Thiết kế và cài đặt được thuật toán tô màu.

- Sử dụng được các phép biến hình trong không gian 2 chiều, 3 chiều để làm thay đổi một hình ảnh đã có sẵn.
- Có thể tạo một cửa sổ để cắt - dán một hình.
- Hiểu được cách biểu diễn của một số đối tượng 3 chiều và các phép chiếu cơ bản.

2. Đối tượng sử dụng

Môn Đồ họa máy tính được giảng dạy cho sinh viên năm thứ ba của khoa Công nghệ thông tin – Trường Đại học Kinh tế kỹ thuật công nghiệp.

3. Phương pháp nghiên cứu:

Sinh viên nghe đầy đủ các tiết học lý thuyết trên lớp, kết hợp với các ngôn ngữ lập trình đã được học để thực hiện cài đặt các thuật toán trong nội dung môn học.

Hình thức đánh giá kết quả:

- Điểm quá trình: Kiểm tra lý thuyết kết hợp bài tập cài đặt chương trình
- Thi hết học phần: Thi trắc nghiệm.

MỤC LỤC

LỜI GIỚI THIỆU	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ HỌA MÁY TÍNH	9
1.1. Giới thiệu về đồ họa máy tính.....	9
1.1.1. Mở đầu	9
1.1.2. Lịch sử phát triển	10
1.1.3. Một số ứng dụng	11
1.1.4 Phân loại các lĩnh vực của kỹ thuật đồ họa máy tính	13
1.2. Tổng quan về một hệ đồ họa.....	14
1.2.1. Phần cứng đồ họa.....	15
1.2.2. Phần mềm đồ họa.....	15
1.3 Các kỹ thuật đồ họa	15
1.3.1 Kỹ thuật đồ họa điểm	16
1.3.2 Kỹ thuật đồ họa vector	16
CHƯƠNG 2: ĐỒ HỌA 2 CHIỀU	19
2.1. Giới thiệu về các đối tượng đồ họa cơ sở.....	19
2.1.1 Mở đầu.....	19
2.1.2 Các hệ tọa độ	20
2.1.3 Các đối tượng đồ họa cơ sở	20
2.2. Các thuật toán vẽ đường thẳng	22
2.2.1 Mở đầu.....	22
2.2.2 Thuật toán DDA (Digital Differential Analyzer)	23
2.2.3 Thuật toán Bresenham	25
2.2.4 Thuật toán Midpoint vẽ đường thẳng	27
2.3. Thuật toán Midpoint vẽ đường tròn	29
2.4 Các phép biến đổi cơ sở của đối tượng 2 chiều.....	31
2.4.1 Mở đầu.....	31
2.4.2 Phép tịnh tiến	32
2.4.3 Phép biến đổi tỷ lệ tâm $O(0,0)$	33
2.4.4 Phép quay tâm $O(0,0)$	34

2.4.5. Kết hợp các phép biến đổi.	35
2.5. Một số tính chất của phép biến đổi affine.	37
2.5.1. Bảo toàn đường thẳng:	37
2.5.2 Bảo toàn tính song song	38
2.5.3 Bảo toàn tính tỷ lệ.....	38
2.6 Một số phép biến đổi khác.....	38
2.6.1 Phép biến đổi tỷ lệ tâm bất kỳ	38
2.6.2 Phép quay có tâm quay bất kỳ.....	39
2.6.3 Phép đối xứng.....	40
2.6.4 Phép biến đổi ngược	40
2.7 Các thuật toán xén đoạn thẳng.....	40
2.7.1 Thuật toán Cohen-Sutherland.....	42
2.7.2 Thuật toán Liang Barsky	45
CHƯƠNG 3: ĐỒ HỌA 3 CHIỀU	54
3.1. Tổng quan về đồ họa ba chiều	54
3.2. Biểu diễn các đối tượng ba chiều	54
3.2.1. Biểu diễn mặt đa giác	55
3.2.2. Lưới đa giác (polygon meshes)	58
3.2.3. Các đường cong và mặt cong	59
3.3. Các phép biến đổi hình học ba chiều.....	59
3.3.1 Các phép biến đổi cơ bản	60
3.3.2 Các phép đối xứng.....	63
3.3.3 Phép biến đổi tỷ lệ tâm bất kỳ	64
3.4. Các phép chiếu.....	64
3.4.1 Phép chiếu song song	65
3.4.2 Phép chiếu phối cảnh.....	67
CHƯƠNG 4: CÁC HỆ MÀU	73
4.1 Ánh sáng và sự cảm nhận màu sắc của mắt	73
4.2 Biểu diễn màu trên màn hình đồ họa.....	74
4.2.1 Chế độ hiển thị màu RGBA.....	74

4.2.2. Chế độ hiển thị màu chỉ mục.....	76
4.3. Hệ RGB (Red – Green – Blue):.....	77
4.4. Hệ CMY và CMYK (Cyan - Magenta - Yellow):.....	79
4.4.1 Hệ màu CMY:	79
4.4.2 Hệ màu CMYK:	80
4.5 Một số hệ màu khác.....	81
4.5.1. Hệ màu HSV (Hue - Saturation - Value):	81
4.5.2. Hệ màu HSI (Hue - Saturation - Intensity):	82
CHƯƠNG 5: CÁC KỸ THUẬT ĐỒ HỌA TRONG BỘ CÔNG CỤ ADOBE AFTER EFFECT	90
5.1 Giới thiệu về After Effect.....	90
5.2. Giao diện.....	91
5.3. Thực hiện một Project	95
5.4. Kỹ thuật Storyboard	98
5.5. Tạo hình đồ họa nhân vật: gắn xương, các chuyển động cơ bản	100
5.6. Kỹ thuật tạo bối cảnh đồ họa	103
TÀI LIỆU THAM KHẢO	107

DANH MỤC HÌNH VẼ

Hình 2.1: Các điểm lân cận của (x_i, y_i)	22
Hình 2.2: Đường thẳng có hệ số góc $0 < m < 1$	23
Hình 2.3: Minh họa thuật toán DDA trường hợp $0 < m < 1$	23
Hình 2.4: Lưu đồ thuật toán vẽ đường thẳng DDA.....	24
Hình 2.5: Minh họa thuật toán Bresenham, $0 < m < 1$	25
Hình 2.6: Lưu đồ thuật toán vẽ đường thẳng Bresenham	26
Hình 2.7: Minh họa thuật toán Midpoint vẽ đường thẳng.....	28
Hình 2.8: Các vị trí đối xứng của đường tròn	29
Hình 2.9: Minh họa thuật toán Midpoint vẽ đường tròn	29
Hình 2.10: Lưu đồ giải thuật Midpoint vẽ đường tròn.....	30
Hình 2.11: Minh họa phép tịnh tiến.....	32
Hình 2.12: Minh họa phép biến đổi tỷ lệ.....	33
Hình 2.13: Minh họa phép quay tâm O	34
Hình 2.14: Minh họa phép quay có tâm quay bất kỳ	39
Hình 2.15: (a)Trước khi thực hiện xén đoạn thẳng, (b) sau khi thực hiện	41
Hình 2.16: Cách đánh mã vùng – thuật toán Cohen-Sutherland	42
Hình 2.17: Minh họa thuật toán Cohen - Sutherland	43
Hình 2.18: Giải thuật xén đoạn thẳng Cohen-Sutherland	44
Hình 2.19: Minh họa thuật toán Liang Barsky	46
Hình 3.1: Mô hình wireframe của một hình trụ	55
Hình 3.2: Minh họa mô hình khung kết nối	56
Hình 3.3: Vector pháp tuyến của mặt phẳng	58
Hình 3.4: Triangle strip và quadrilateral mesh.....	59
Hình 3.5: Minh họa phép tịnh tiến.....	60
Hình 3.6: Minh họa phép biến đổi tỷ lệ.....	61
Hình 3.7: Phép quay, (a) quay quanh trục z, (b) quanh trục x, (c) quanh trục y	62
Hình 3.8: Phép chiếu trục giao, (a) phép chiếu 6 mặt, (b) phép chiếu 3 mặt.....	65
Hình 3.9: Minh họa phép chiếu trục lượng	66
Hình 3.10: Phép chiếu Cavalier, (a) góc $\phi=45^0$, (b) $\phi=30^0$	67

Hình 3.11: Phép chiếu Cabinet, (a) góc $\phi=450$, (b) $\phi=300$	67
Hình 3.12: Minh họa phép chiếu phối cảnh	68
Hình 3.13: (a) Một tâm chiếu, (b) hai tâm chiếu, (c) ba tâm chiếu	68
Hình 3.14: Phép chiếu phối cảnh tâm chiếu I(0,0,d)	69
Hình 4.1: Minh họa phối màu.....	75
Hình 4.2: Ánh xạ màu lên pixel.....	76
Hình 4.3: Minh họa sử dụng bản đồ màu để vẽ.....	76
Hình 4.4: Hệ màu RGB	77
Hình 4.5: Hệ màu CMY	80
Hình 4.6: Hệ màu HSV	81
Hình 4.7: Hệ màu HSI	82
Hình 4.8: Màu đơn sắc (Monochromatic)	83
Hình 4.9: Kết hợp màu kiểu bổ sung.....	83
Hình 4.9: Kết hợp màu kiểu bổ sung.....	84
Hình 4.10: Kết hợp màu kiểu bộ ba.....	84
Hình 4.11: Kết hợp màu kiểu hình chữ nhật	85
Hình 4.12: Kết hợp màu kiểu hình vuông	85
Hình 5.1: Giao diện After Effect	91
Hình 5.2: Mở workspace của After Effect	92
Hình 5.3: Panel trong After Effect	92
Hình 5.4: Các thao tác với Panel trong After Effect	93
Hình 5.5: Các icon Panel trong After Effect	94
Hình 5.6: Vùng an toàn trong After Effect.....	94
Hình 5.7: Toolbar trong After Effect.....	95
Hình 5.8: Chức năng New Project trong After Effect.....	95
Hình 5.9: Hiệu chỉnh Preferences trong After Effect	96
Hình 5.10: Hiệu chỉnh Preferences trong After Effect	96
Hình 5.11: Import File trong After Effect	97
Hình 5.12: Hiệu chỉnh thông số Import File trong After Effect.....	98
Hình 5.13: Minh họa Storyboard trong After Effect	100

Hình 5.14: File PSD nhân vật.....	101
Hình 5.15: Import File PSD nhân vật.....	101
Hình 5.16: Mô tả Import File PSD nhân vật	102
Hình 5.17: Layer nhân vật	102
Hình 5.18: Hiệu chỉnh các điểm neo nhân vật	103
Hình 5.19: Tạo mô tả chuyển động nhân vật	103
Hình 5.20: Bối cảnh trong After Effect	104
Hình 5.21a: Công cụ Roto Brush	104
Hình 5.21b: Công cụ Roto Brush	105

CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ HỌA MÁY TÍNH

Mục tiêu của chương: Sinh viên nắm được về lịch sử ra đời và một số các cột mốc phát triển của đồ họa máy tính, nhận thức được vai trò, các ứng dụng và các thành tựu của đồ họa máy tính trong các lĩnh vực như giáo dục, y tế, giải trí, nghệ thuật, khoa học, kỹ thuật..

Sinh viên cần nắm được tổng quan về một hệ đồ họa máy tính, phân loại được các lĩnh vực của đồ họa máy tính, nắm được một số chuẩn của đồ họa và cần phân biệt, so sánh được giữa kỹ thuật đồ họa điểm và đồ họa vector.

1.1. Giới thiệu về đồ họa máy tính

1.1.1. Mở đầu

Đồ họa máy tính là một lĩnh vực của công nghệ thông tin, liên quan đến việc nghiên cứu, xây dựng và tập hợp các công cụ (mô hình lý thuyết và phần mềm) khác nhau để kiến tạo, xây dựng lưu trữ và xử lý các mô hình và hình ảnh của đối tượng, sự vật hiện tượng khác nhau trong đời sống, sản xuất và nghiên cứu. Các mô hình và hình ảnh này có thể là các kết quả thu được từ những lĩnh vực khác nhau của rất nhiều ngành khoa học (vật lý, toán học, ...)

Đồ họa máy tính cũng là lĩnh vực liên quan đến việc thiết kế, chế tạo phần cứng như: các thiết bị hiển thị, các thiết bị đầu vào như bàn phím, chuột, bút quang ... các thuật toán cần thiết để phát sinh hình ảnh trên các thiết bị này, các phần mềm được sử dụng cho cả người lập trình hệ thống và người lập trình ứng dụng đồ họa.

Đồ họa máy tính tương tác là một trong những phương tiện mang lại thêm nhiều sự thuận lợi cho người dùng trong việc phát sinh hình ảnh. Chúng ta có thể tạo các hình ảnh không chỉ của các đối tượng cụ thể, thực tế, mà còn của các đối tượng trừu tượng, nhân tạo; biểu diễn dữ liệu mà không có tính kế thừa về mặt hình học như kết quả điều tra, khảo sát..

Hơn nữa, với đồ họa máy tính chúng ta không bị giới hạn trong các ảnh tĩnh. Các ảnh động thông thường mang lại nhiều hiệu quả hơn so với ảnh tĩnh, đặc biệt là với các hiện tượng biến đổi theo thời gian, cả thực tế (như sự đổi hướng của cánh máy bay siêu âm, hay sự phát triển của khuôn mặt người từ lúc trẻ tới lúc già) và trừu tượng (như là xu hướng phát triển của việc sử dụng năng lượng, sự gia tăng dân số, ...).

Có nhiều cách tiếp cận trong việc học môn đồ họa, trải rộng từ việc nghiên cứu phần cứng, nghiên cứu các kỹ thuật để hiển thị các đối tượng đồ họa... tới việc học để biết cách sử dụng một ứng dụng đồ họa. Ở đây chúng ta tiếp cận môn học từ góc độ của người lập trình ứng dụng, sử dụng tất cả các hỗ trợ của phần cứng, các công cụ phần mềm để xây dựng nên các ứng dụng.

Tuy nhiên để có thể thiết kế và cài đặt các chương trình ứng dụng đồ họa được tốt, ngoài việc tìm hiểu các khả năng của công cụ lập trình, chúng ta cũng cần phải nắm vững các khái niệm về phần cứng; các vấn đề, các nguyên lý liên quan đến cài đặt phần mềm, các thuật toán, các ứng dụng, ...

1.1.2. Lịch sử phát triển

Năm 1955, hệ thống phòng không SAGE (Môi trường mặt đất bán tự động) bắt đầu hoạt động, sử dụng màn hình quét vector làm đầu ra chính và Light pen là thiết bị đầu vào và được sử dụng rộng rãi vào năm 1958.

Tập đoàn thiết bị kỹ thuật số DEC được thành lập vào năm 1957, bắt đầu chế tạo minicomputers mà sau này được sử dụng phát triển đồ họa máy tính.

Năm 1960, nhà khoa học người Mỹ, William Fette nghiên cứu, xây dựng mô hình buồng lái máy bay cho hãng Boeing của Mỹ. Ông đã dựa trên hình ảnh 3 chiều của mô hình người phi công trong buồng lái của máy bay để xây dựng nên một mô hình tối ưu cho buồng lái máy bay. Phương pháp này cho phép các nhà thiết kế quan sát một cách trực quan vị trí của người lái trong khoang. Ông đặt tên cho phương pháp này là đồ họa máy tính (Computer Graphics). Màn hình là thiết bị thông dụng nhất trong hệ đồ họa, các thao tác của hầu hết các màn hình đều dựa trên thiết kế ống tia âm cực CRT (Cathode ray tube). Khi đó máy tính xử lý chậm, đắt và không chắc chắn (không đáng tin cậy).

Vào những năm 1960, lĩnh vực này đã có những phát triển đầu tiên. Năm 1961, Ivan Sutherland phát triển dự án Sketchpad tại MIT. Bảng vẽ được sử dụng và một thiết bị light pen làm thiết bị đầu vào chính, một máy hiện sóng là thiết bị đầu ra. Phiên bản đầu tiên chỉ xử lý các số liệu hai chiều và sau đó được phát triển để vẽ, biến đổi và hiển thị các đối tượng 3 chiều dưới dạng hình chiếu xuống mặt phẳng 2 chiều.

Kỹ thuật đồ họa liên tục được phát triển vào những năm 1960 và 1970 kéo theo sự ra đời của các chuẩn đồ họa, làm tăng khả năng giao tiếp và tái sử dụng của phần mềm cũng như các thư viện đồ họa.

Sự phát triển đột phá của công nghệ điện tử máy tính vào thập niên 1980 làm xuất hiện một loạt các vi mạch hỗ trợ cho việc truy xuất đồ họa, làm giảm giá thành của máy tính cá nhân. Vào cuối những năm 1980, các máy tính đồ họa SGI (Silicon Graphics Inc.) đã tạo ra một số phim hoạt hình ngắn đầu tiên tại Pixar.

Hệ điều hành Microsoft Windows 3.0 được xuất xưởng lần đầu tiên vào năm 1990 thúc đẩy to lớn cho khái niệm GUI (Graphical User Interface).

Ngày càng nhiều ứng dụng được phát triển chạy trên MS Windows. OpenGL (Thư viện đồ họa mở) được SGI đưa ra năm 1992. Những năm 1990 cũng chứng kiến sự phát triển nhanh chóng trong đồ họa ba chiều, đặc biệt là trong phát triển game, đa phương tiện và hoạt hình. Quake, một trong những game 3D hoàn chỉnh đầu tiên, đã được phát hành vào năm 1996.

Ngày nay, đồ họa có những tác động to lớn đến ngành công nghiệp giải trí, các lĩnh vực khoa học, kỹ thuật, y học, thiết kế trang web, truyền thông, xử lý thông tin thiết lập các tương tác mới từ lớn đến nhỏ các yếu tố thực tế ảo, màn hình cỡ lớn, màn hình thực tế ảo, thiết bị cảm ứng đa điểm và điện thoại thông minh...[1][5]

Nếu nhìn lịch sử đồ họa theo góc độ phát triển của kỹ thuật hiển thị ta sẽ thấy các kỹ thuật hiển thị phát triển theo những giai đoạn sau:

- *Kỹ thuật hiển thị bằng ký tự:*

Kỹ thuật này chỉ cho phép hiển thị text và các hình ảnh đồ họa đơn giản. Giao tiếp với người sử dụng thông qua các lệnh dưới dạng text. Có thể sử dụng những ký tự mã hóa đặc biệt để hiển thị đối tượng. Tất cả các chương trình và phần mềm được thực hiện đều là đơn nhiệm.

Ví dụ:

- Hệ điều hành MS_DOS. Hệ điều hành tiêu biểu của giai đoạn 80, đầu những năm 90 của Microsoft.
- Phần mềm soạn thảo văn bản BKED chạy trên môi trường MS_DOS.

- *Kỹ thuật hiển thị vector:*

Kỹ thuật này phát triển từ năm 1963 đến năm 1980, cho phép hiển thị text, các đường thẳng, các hình đơn giản. Giao tiếp với người sử dụng được thực hiện thông qua các dòng lệnh, các phím “nóng” và menu chọn.

- *Kỹ thuật ảnh 2 chiều:*

Kỹ thuật này cho phép hiển thị các cửa sổ (window), các biểu tượng (icon), và các dòng văn bản (text). Trong giao tiếp với người sử dụng đã hạn chế việc gõ lệnh, thay vào đó người sử dụng có thể thực hiện thông qua các giao diện đồ họa (Graphical User Interface)

- *Trạm làm việc đồ họa (3D Graphics Workstation)*

Công nghệ này được phát triển bởi công ty Silicon Graphics, một trong những công ty phát triển đồ họa hàng đầu của Mỹ. Công nghệ này cho phép hiển thị các hình ảnh thực và hình ảnh thời gian thực 3 chiều của hình ảnh và sự vật dựa trên các cảnh 3 chiều.

Môi trường giao tiếp với người sử dụng được xây dựng trên cơ sở các hình ảnh 2 chiều, 3 chiều và mô phỏng thế giới thực. [4]

1.1.3. Một số ứng dụng

Như đã nói ở trên, đồ họa máy tính không còn là những gì mới mẻ, xa lạ nữa mà nó trở nên gần gũi và cần thiết trên hầu hết các thiết bị chúng ta sử dụng hàng ngày. Dưới đây là một số các ứng dụng cơ bản của đồ họa máy tính.

a, Hỗ trợ thiết kế

Một trong những ứng dụng lớn nhất của đồ họa máy tính là hỗ trợ thiết kế (CAD – computer-aided design). Ngày nay CAD đã được sử dụng hầu hết trong việc thiết kế các

cao ốc, ô tô, máy bay, tàu thủy, tàu vũ trụ, máy tính, trang trí mẫu vải, và rất nhiều sản phẩm khác.

Người ta có thể quan sát các đối tượng được thiết kế qua mô tả trên máy tính theo bất kỳ hướng nào. Những thay đổi trên đối tượng được cập nhật nhanh chóng, được thể hiện thử nghiệm lại ngay.

b, Biểu diễn thông tin

Các công cụ đồ họa trong các ứng dụng có thể phát sinh các biểu đồ, đồ thị, ... thuận tiện cho việc minh họa mối quan hệ giữa nhiều đối tượng với nhau. Các biểu đồ có thể dưới dạng 2 chiều, 3 chiều, được biểu diễn trực quan. Trong đó, các ứng dụng thường được dùng để tóm lược các dữ liệu về tài chính, thống kê, kinh tế, khoa học, toán học, ... giúp cho việc nghiên cứu, quản lí, ... một cách có hiệu quả.

c, Lĩnh vực giải trí, nghệ thuật

Trong lĩnh vực nghệ thuật, các chương trình máy tính như Paint Shop Pro, Adobe Photoshop, 3D Studio, ... hỗ trợ rất đắc lực cho các họa sĩ, các nhà tạo mẫu trong việc thiết kế các hình ảnh sống động, và rất thực. Với các chương trình này, người họa sĩ được máy tính tạo cho cảm giác y như đang làm việc ngoài đời thực bằng cách cung cấp các công cụ như khung vẽ, giá vẽ, bảng pha màu, các hiệu ứng ba chiều, ... làm cho họ cảm thấy rất thoải mái và tiện lợi.

Ngoài ra đồ họa máy tính còn giúp tạo ra các chương trình trò chơi, giải trí; hỗ trợ cho các kĩ xảo điện ảnh, cho các nhà làm phim. Có nhiều bộ phim rất nổi tiếng nhờ vào kĩ xảo điện ảnh như : Công viên Khủng long kỉ Jura (Jurassic Park), Titanic, Thế giới nước (Water World), ...

d, Giáo dục và đào tạo

Hiện nay các chương trình mô phỏng cấu trúc của các vật thể, tiến trình của các phản ứng hóa học, hoạt động của các gói tin trên mạng máy tính, ... được dùng rất nhiều trong việc hỗ trợ giảng dạy.

Trong đào tạo, các ứng dụng mô phỏng được dùng để kiểm tra trình độ người lái, huấn luyện phi công, điều khiển giao thông, ...

e, Giao tiếp giữa máy tính và người dùng

Mọi ứng dụng đều phải có giao diện giao tiếp với người dùng. Giao diện đồ họa thực sự là một cuộc cách mạng mang lại sự thuận tiện và thoải mái cho người dùng ứng dụng. Các ứng dụng dựa trên hệ điều hành MS Windows là một minh họa rất trực quan của giao diện đồ họa. Các chức năng của các ứng dụng này được thiết kế cho người dùng làm việc thông qua các biểu tượng mô tả chức năng đó.

Ví dụ: chức năng lưu tập tin được hiểu thông qua biểu tượng đĩa mềm, chức năng in ấn được hiểu thông qua biểu tượng máy in, ... Để chọn các chức năng, người dùng sử dụng chuột trỏ đến và nhấn vào các biểu tượng tương ứng. Điểm thuận lợi chính khi dùng

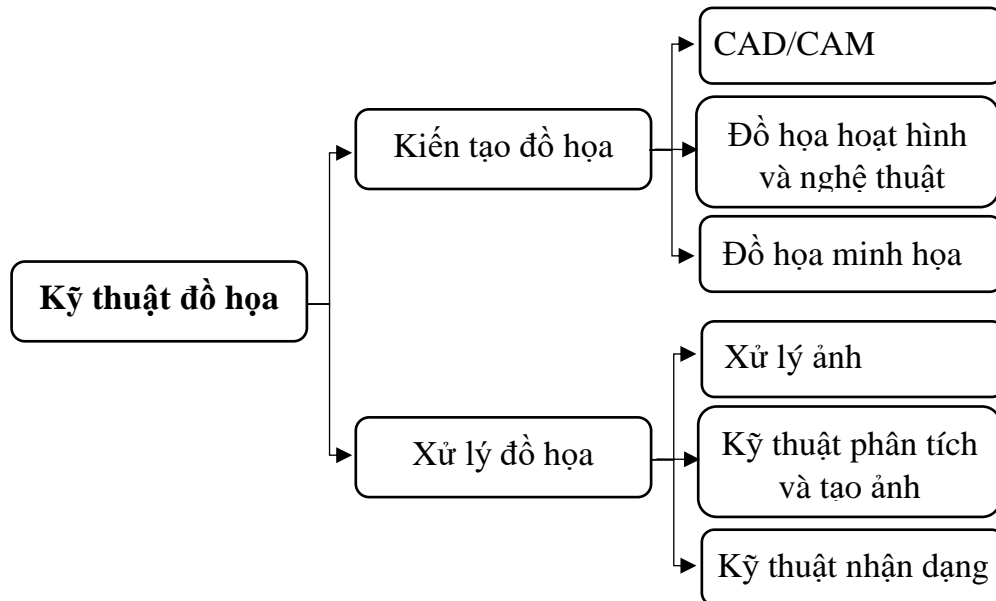
biểu tượng là kích thước không gian mà nó chiếm ít hơn nhiều so với dùng văn bản để mô tả cho cùng một chức năng, ngoài ra việc nắm bắt các chức năng qua các biểu tượng sẽ dễ dàng hơn rất nhiều khi người dùng gặp trở ngại về mặt ngôn ngữ.

Các ứng dụng có giao diện đồ họa còn cho phép người dùng khả năng làm việc dễ dàng với nhiều cửa sổ với nhiều dạng tài liệu khác nhau cùng một lúc.

1.1.4 Phân loại các lĩnh vực của kỹ thuật đồ họa máy tính

Có nhiều phương pháp để phân loại các lĩnh vực của đồ họa máy tính.

a, *Phân loại theo tiêu chí là mục đích xử lý dữ liệu trong các lĩnh vực:*



- *Các hệ CAD/CAM* (Computer Aided Design/Computer Aided Manufacture System): kỹ thuật đồ họa tập hợp các công cụ, các kỹ thuật trợ giúp cho thiết kế các chi tiết và các hệ thống khác nhau: hệ thống cơ, hệ thống điện, hệ thống điện tử...

- *Đồ họa hoạt hình và nghệ thuật*: bao gồm các công cụ giúp cho các họa sĩ, các nhà thiết kế phim hoạt hình chuyên nghiệp làm các kỹ xảo hoạt hình, vẽ tranh...

Ví dụ: phần mềm 3D Studio, 3D Animation, 3D Studio Max

- *Đồ họa minh họa* (Presentation Graphics): gồm các công cụ giúp hiển thị các số liệu thí nghiệm một cách trực quan, dựa trên các mẫu đồ thị hoặc các thuật toán có sẵn.

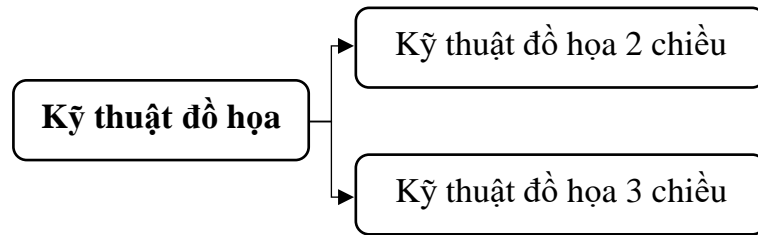
- *Kỹ thuật xử lý ảnh* (Computer Imaging): sau quá trình xử lý ảnh cho ta ảnh số của đối tượng. Trong quá trình xử lý ảnh sử dụng rất nhiều các kỹ thuật phức tạp: kỹ thuật khôi phục ảnh, kỹ thuật làm nổi ảnh, kỹ thuật xác định biên ảnh.

- *Kỹ thuật phân tích và tạo ảnh* (Image Synthesis): là lĩnh vực xây dựng mô hình và hình ảnh của các vật thể dựa trên các đối tượng và mối quan hệ giữa chúng.

- *Kỹ thuật nhận dạng* (Pattern Recognition): từ những ảnh mẫu có sẵn ta phân loại theo cấu trúc, hoặc theo các tiêu chí được xác định từ trước và bằng các thuật toán chọn lọc để có thể phân tích hay tổng hợp ảnh đã cho thành một tập hợp các ảnh gốc, các ảnh

gốc này được lưu trong một thư viện và căn cứ vào thư viện này ta xây dựng được các thuật giải phân tích và tổ hợp ảnh.

b, Phân loại theo hệ toạ độ dùng trong thuật đồ họa:



- *Kỹ thuật đồ họa hai chiều*: là kỹ thuật đồ họa máy tính sử dụng hệ toạ độ hai chiều (hệ toạ độ phẳng), sử dụng rất nhiều trong kỹ thuật xử lý bản đồ, đồ thị.

- *Kỹ thuật đồ họa ba chiều*: là kỹ thuật đồ họa máy tính sử dụng hệ toạ độ ba chiều, đòi hỏi rất nhiều tính toán và phức tạp hơn nhiều so với kỹ thuật đồ họa hai chiều.[1]

1.2. Tổng quan về một hệ đồ họa

Một trong các yêu cầu chính của một hệ thống đồ họa là các ứng dụng có thể thực thi trên nhiều hệ thiết bị, được phát triển không phụ thuộc vào phần cứng. Để có được điều đó, phải có tiêu chuẩn hóa cho môi trường đồ họa ở mức chức năng, bằng việc cung cấp sự độc lập thiết bị và ngôn ngữ lập trình.

Sự độc lập với thiết bị cho phép các chương trình ứng dụng đồ họa chạy trên các hệ thống phần cứng khác nhau. Điều này được thực hiện thông qua thiết bị nhập xuất logic của thư viện đồ họa ánh xạ thiết bị cụ thể. Cho tới nay, có những tiêu chuẩn đồ họa đã được phát triển trong nhiều năm, bao gồm:

- GKS(Graphics Kernel System - 1985), phát triển riêng cho các thiết bị nhập xuất 2 chiều.
- GKS-3D bổ sung thêm khả năng lập trình 3 chiều.
- CGI (Computer Graphics Interface System): Hệ chuẩn cho các phương pháp giao tiếp với các thiết bị ngoại vi.
- CGM (Computer Graphics Metafile): Xác định các chuẩn cho việc lưu trữ và chuyển đổi hình ảnh.
- VRML (Virtual Reality Modeling Language): Ngôn ngữ thực tại ảo, một hướng phát triển trong công nghệ hiển thị được đề xuất bởi công ty Silicon Graphics, sau đó được chuẩn hóa như một chuẩn công nghiệp.
- PHIGS (Programmer's Hierarchical Interactive Graphics Standar): Xác định các phương pháp chuẩn cho mô hình thời gian thực và lập trình hướng đối tượng.
- PHIGS+ bao gồm khả năng lập trình không gian, tạo thành thao tác dữ liệu đồ họa phức tạp ...

- OpenGL: Thư viện đồ họa của hãng Silicon Graphics, được xây dựng theo đúng chuẩn của một hệ đồ họa – năm 1993.
- DirectX: Thư viện đồ họa của hãng Microsoft.

Các tiêu chuẩn đồ họa thực tế là kết quả của việc chấp nhận trong công nghiệp thiết kế các giao diện đặc trưng, được đề xuất bởi nhiều công ty.

Để đảm bảo sự linh hoạt, các tiêu chuẩn đồ họa thiết lập cho các thay đổi tối thiểu, cho phép định địa chỉ các thiết bị nhập xuất khác nhau. Khởi đầu, người lập trình tạo ra một hệ thống tọa độ mô hình, mô tả đối tượng gọi là hệ thống tọa độ thực. Tiếp theo, là hệ tọa độ tiêu chuẩn và hệ tọa độ thiết bị. Chương trình ứng dụng sẽ giao tiếp với hệ tọa độ chuẩn theo cách thức phù hợp, không quan tâm đến thiết bị xuất được dùng. Do đó, tạo ra sự độc lập với thiết bị trong việc tạo ra hình ảnh của đối tượng

1.2.1. Phần cứng đồ họa

Phần cứng đồ họa bao gồm các thành phần:

- ✓ CPU: Thực hiện các chương trình ứng dụng.
- ✓ Bộ xử lý hiển thị (Display Processor): Thực hiện công việc hiển thị dữ liệu đồ họa.
- ✓ Bộ nhớ hệ thống (System Memory): Chứa các chương trình và dữ liệu đang thực hiện.
- ✓ Bộ đệm (Frame buffer): Chứa các hình ảnh hiển thị.
- ✓ Bộ điều khiển màn hình (Video Controller): Điều khiển màn hình, chuyển dữ liệu dạng số ở frame buffer thành các điểm sáng trên màn hình.

1.2.2. Phần mềm đồ họa

Phần mềm đồ họa bao gồm các công cụ lập trình cung cấp một tập các hàm đồ họa có thể được dùng trong các ngôn ngữ lập trình cấp cao. Các hàm cơ sở của nó bao gồm việc tạo các đối tượng cơ sở của hình ảnh như đoạn thẳng, đa giác, đường tròn,... thay đổi màu sắc, chọn khung nhìn, áp dụng các phép biến đổi.

Các ứng dụng đồ họa được thiết kế cho những người dùng không phải là lập trình viên, cho phép người dùng tạo các đối tượng, hình ảnh, ... mà không cần quan tâm tới việc chúng được tạo ra như thế nào. Ví dụ như là Photoshop, AutoCAD, ...

Khi các ứng dụng được thiết kế với các hàm đồ họa chuẩn, nó có thể thực thi được trên nhiều hệ thống khác nhau và được tái sử dụng trong nhiều ứng dụng khác nhau.

1.3 Các kỹ thuật đồ họa

Ngày nay, số lượng các hệ thống sử dụng kỹ thuật đồ họa tương tác đã trở nên rất lớn, ngày càng nhiều và càng trở nên đa dạng hơn, phong phú hơn. Tuy vậy, căn cứ vào phương pháp xử lý các dữ liệu trong hệ thống mà người ta phân biệt ra hai hệ thống đồ họa: *Kỹ thuật đồ họa điểm* (Sample based – Graphics) và *kỹ thuật đồ họa vector* (Geometry based – Graphics).

1.3.1 Kỹ thuật đồ họa điểm

Các mô hình, hình ảnh của các đối tượng được hiển thị thông qua từng pixel (từng mẫu rời rạc) trong kỹ thuật này ta có thể:

- Tạo ra, thay đổi các thuộc tính, xóa đi từng pixel của mô hình và hình ảnh các đối tượng.
- Các mô hình hình ảnh được hiển thị như một lưới điểm (grid) các pixel rời rạc.
- Từng pixel đều có vị trí xác định, được hiển thị với một giá trị rời rạc (số nguyên) các thông số hiển thị (màu sắc hoặc độ sáng)

Tập hợp tất cả các pixel cho chúng ta mô hình, hình ảnh đối tượng mà chúng ta muốn hiển thị. Có 2 phương pháp để tạo ra các pixel:

- Phương pháp dùng phần mềm để vẽ trực tiếp từng pixel một. Dựa trên các lý thuyết mô phỏng để xây dựng nên hình ảnh mô phỏng của sự vật.
- Phương pháp rời rạc hoá (số hoá) hình ảnh thực của đối tượng. Có thể sửa đổi (image editing) hoặc xử lý (image processing) mảng các pixel thu được theo những phương pháp khác nhau để thu được hình ảnh đặc trưng của đối tượng.

1.3.2 Kỹ thuật đồ họa vector

Nguyên lý xây dựng các mô hình, hình ảnh trong kỹ thuật đồ họa vector như sau:

- Xây dựng mô hình hình học (geometrical model) cho mô hình hoặc hình ảnh của đối tượng.
- Xác định các thuộc tính của mô hình hình học này.
- Quá trình tô trát (rendering) để hiển thị từng điểm của mô hình, hình ảnh thực của đối tượng.

Trong kỹ thuật đồ họa vector, chỉ lưu trữ mô tả toán học của các thành phần trong mô hình hình học cùng với các thuộc tính tương ứng của nó mà không lưu lại toàn bộ các pixel của mô hình tô trát (rendering) được.

Các thành phần được mô tả trong mô hình hình học của đối tượng được gọi là các thực thể hình học cơ sở, các đối tượng hình ảnh sẽ được xây dựng từ những thực thể cơ sở này. Hình ảnh được tô trát từ nhiều góc nhìn và điểm nhìn khác nhau.

So sánh giữa 2 kỹ thuật đồ họa điểm và đồ họa vector:

<i>Đồ họa điểm</i>	<i>Đồ họa vector</i>
- Hình ảnh và mô hình của các đối tượng được biểu diễn bởi tập hợp các điểm của lưới (grid)	- Hình ảnh và mô hình của các đối tượng được biểu diễn bởi tập hợp các thực thể hình học cơ sở
- Thay đổi thuộc tính của các pixel dẫn đến thay đổi từng phần và từng vùng của đối tượng.	- Xử lý với từng thành phần hình học cơ sở của đối tượng và thực hiện quá trình tô trát, hiển thị.

Đồ họa điểm

- Copy được các pixel từ một hình ảnh này sang hình ảnh khác.

Đồ họa vector

- Quan sát hình ảnh và mô hình của hình ảnh và sự vật ở nhiều góc độ khác nhau bằng cách thay đổi điểm nhìn và góc nhìn.[1]

Câu hỏi ôn tập:

1. Trình bày một số mốc lịch sử phát triển của đồ họa máy tính
2. Trình bày một số ứng dụng của đồ họa máy tính trong các lĩnh vực: giáo dục, y tế, giải trí, nghệ thuật, kỹ thuật...
3. Tại sao phải sử dụng các chuẩn đồ họa, nêu một số chuẩn đồ họa?
4. Trình bày tổng quan về phần cứng, phần mềm đồ họa.
5. Trình bày về kỹ thuật đồ họa điểm và kỹ thuật đồ họa vector.
6. Phân biệt giữa đồ họa điểm và đồ họa vector.

CHƯƠNG 2: ĐỒ HỌA 2 CHIỀU

Mục tiêu của chương: Sinh viên cần nắm được các thuật toán DDA, Bresenham, Midpoint vẽ các đường cơ bản như đường thẳng, đường tròn, có khả năng cài đặt các thuật toán này bằng ngôn ngữ C, C++ ..

Có thể thực hiện tính toán được ảnh của các điểm 2D sau các phép biến đổi: Tỷ lệ tâm $O(0,0)$, tâm bất kỳ, phép tịnh tiến, phép quay tâm $O(0,0,0)$, quay tâm bất kỳ, và các phép lấy đối xứng qua trục tọa độ, gốc tọa độ.

Sinh viên cần nắm được các tính chất của các phép biến đổi Affine trên đối tượng đồ họa 2D, hiểu và cài đặt được các thuật toán xén đoạn thẳng: Cohen-Sutherland và Liang-Barsky.

2.1. Giới thiệu về các đối tượng đồ họa cơ sở

2.1.1 Mở đầu

Bất kỳ một ảnh mô tả thế giới thực nào bao giờ cũng được cấu trúc từ tập các đối tượng đơn giản hơn. Với các ảnh đồ họa phát sinh bằng máy tính, hình dạng và màu sắc của mỗi đối tượng có thể được mô tả riêng biệt bằng hai cách:

- Bằng dãy các pixel tương ứng
- Bằng tập các đối tượng hình học cơ sở như: đoạn thẳng, vùng tô đa giác, ...

Sau đó, các ảnh sẽ được hiển thị bằng cách nạp các pixel vào vùng đệm khung. Với các ảnh được mô tả bằng các đối tượng hình học cơ sở, cần phải có một quá trình chuyển các đối tượng này về dạng ma trận các pixel trước.

Quá trình này còn được gọi là quá trình chuyển đổi bằng dòng quét (scan-converting). Bất kỳ công cụ lập trình đồ họa nào cũng cung cấp các hàm để mô tả một ảnh dưới dạng các đối tượng hình học cơ sở hay còn gọi là các đối tượng đồ họa cơ sở (output primitives) và các hàm cho phép kết hợp tập các đối tượng cơ sở để tạo thành đối tượng có cấu trúc phức tạp hơn. Mỗi đối tượng đồ họa cơ sở được mô tả thông qua dữ liệu về tọa độ và các thuộc tính của nó, đây chính là thông tin cho biết kiểu cách mà đối tượng được hiển thị.

Đối tượng đồ họa cơ sở đơn giản nhất là điểm và đoạn thẳng, ngoài ra còn có đường tròn, và các đường conics, mặt bậc hai, các mặt và đường cong, các vùng tô đa giác, chuỗi kí tự, ... cũng được xem là các đối tượng đồ họa cơ sở để giúp xây dựng các ảnh phức tạp. Chương này sẽ khảo sát các thuật toán hiển thị các đối tượng đồ họa cơ sở cho các thiết bị hiển thị dạng điểm.

Xét về mặt bản chất, các thuật toán này thực hiện quá trình chuyển đổi các đối tượng đồ họa cơ sở được mô tả trong hệ tọa độ thực về dãy các pixel có tọa độ nguyên của thiết bị hiển thị. Có hai yêu cầu đặt ra cho các thuật toán đó là:

Đối tượng được mô tả trong hệ tọa độ thực là đối tượng liên tục, còn đối tượng trong hệ tọa độ thiết bị là đối tượng rời rạc, do đó bản chất của quá trình chuyển đổi này chính là sự rời rạc hóa và nguyên hóa các đối tượng sao cho có thể *xác định các điểm nguyên xấp xỉ đối tượng một cách tốt nhất, thực nhất*. Nghĩa là đối tượng hiển thị bằng lưới nguyên trên thiết bị hiển thị phải có hình dạng tương tự như đối tượng trong lưới tọa độ thực và “có vẻ” liên tục, liền nét. Sự liên tục trên lưới nguyên của thiết bị hiển thị có được do mắt người không thể phân biệt được hai điểm quá gần nhau.

Do các đối tượng đồ họa cơ sở là thành phần chính cấu trúc các đối tượng phức tạp nên các thuật toán hiển thị chúng cần phải được *tối ưu hóa về mặt tốc độ*, đây chính là điểm mấu chốt cho việc ra đời các thuật toán khác nhau.

2.1.2 Các hệ tọa độ

a, Hệ tọa độ thế giới thực

Hệ tọa độ thế giới thực (hay hệ tọa độ thực) là hệ tọa độ được dùng mô tả các đối tượng thế giới thực. Một trong các hệ tọa độ thực thường được dùng nhất đó là hệ tọa độ Descartes. Với hệ tọa độ này, bất kỳ một điểm nào trong mặt phẳng cũng được mô tả bằng một cặp tọa độ (x, y) trong đó $x, y \in \mathbb{R}$.

Các tọa độ thế giới thực cho phép người dùng sử dụng bất kỳ một thứ nguyên (dimension) quy ước như foot, cm, mm, km, inch, ... nào và có thể lớn nhỏ tùy ý.

b, Hệ tọa độ thiết bị

Hệ tọa độ thiết bị là hệ tọa độ được dùng bởi một thiết bị xuất cụ thể nào đó như máy in, màn hình, ... Đặc điểm chung của các hệ tọa độ thiết bị đó là:

- Các tọa độ x, y ($x, y \in \mathbb{N}$) của hệ tọa độ thiết bị đều bị giới hạn trong một khoảng nào đó (do kích thước của thiết bị).
- Khoảng giới hạn các tọa độ x, y là khác nhau đối với từng loại thiết bị khác nhau.

c. Hệ tọa độ thiết bị chuẩn

Do cách định nghĩa các hệ tọa độ thiết bị khác nhau nên một hình ảnh được hiển thị trên thiết bị này là chính xác nhưng có thể không được hiển thị chính xác ở thiết bị khác. Người ta xây dựng một hệ tọa độ thiết bị chuẩn (NDCS – Normalized Device Coordinate System) đại diện chung cho tất cả các thiết bị để có thể mô tả các hình ảnh mà không phụ thuộc vào bất kỳ thiết bị nào.

Trong hệ tọa độ thiết bị chuẩn, các tọa độ x, y sẽ được gán các giá trị trong đoạn $[0,1]$. Như vậy, vùng không gian của hệ tọa độ chuẩn chính là hình vuông đơn vị có góc trái dưới $(0,0)$ và góc phải trên $(1,1)$

2.1.3 Các đối tượng đồ họa cơ sở

a. Điểm:

Điểm là thành phần cơ sở được định nghĩa trong một hệ tọa độ. Đối với hệ tọa độ hai chiều mỗi điểm được xác định bởi cặp tọa độ (x, y) .

Ngoài thông tin về tọa độ, điểm còn có thuộc tính màu sắc.

b. Đoạn thẳng:

Một đường thẳng có thể xác định nếu biết hai điểm thuộc nó. Phương trình đường thẳng đi qua hai điểm (x_1, y_1) và (x_2, y_2) có dạng $y = mx + b$, trong đó:

$m = \Delta y / \Delta x$ là hệ số góc của đường thẳng,

với $\Delta y = y_2 - y_1$, $\Delta x = x_2 - x_1$

Dưới dạng phương trình tham số, ta có:

$$\begin{cases} x = x_1 + (x_2 - x_1)t \\ y = y_1 + (y_2 - y_1)t \end{cases}$$

- Nếu $t \in [0, 1]$, ta có các điểm (x, y) thuộc về đoạn thẳng giới hạn bởi hai điểm (x_1, y_1) và (x_2, y_2) ,
- Nếu $t \in [-\infty, +\infty]$, ta sẽ có đường thẳng.

Các thuộc tính của đoạn thẳng:

- Màu sắc
- Độ rộng của nét vẽ
- Kiểu nét vẽ của đoạn thẳng: nét liền, nét gạch, chấm gạch,...

Hầu hết các công cụ đồ họa đều định nghĩa tập các kiểu nét vẽ đoạn thẳng và cho phép người dùng định nghĩa thông qua một mẫu (pattern).

c. Vùng tô

Một vùng tô bao gồm đường biên và vùng bên trong. Đường biên là một đường khép kín.

Các thuộc tính của vùng tô bao gồm:

- Thuộc tính của đường biên: như thuộc tính của đoạn thẳng.
- Thuộc tính của vùng bên trong: màu tô và mẫu tô

d. Ký tự, chuỗi ký tự

Các chuỗi ký tự giúp hiển thị nội dung các thông điệp theo một ngôn ngữ nào đó.

Các thuộc tính của ký tự bao gồm :

- Màu sắc của các ký tự.
- Font chữ: bộ ký tự dùng hiển thị; Nó định nghĩa kiểu, kích thước của ký tự hiển thị. Hình dạng của mỗi ký tự có thể được xác định bởi một tập các đường gấp khúc (trường hợp font vector) hay là mẫu các pixel (font bitmap).

Có nhiều loại font khác nhau như: font bitmap, font truetype, font CHR, ...

- Kích thước: Chiều cao và chiều rộng của ký tự. Các ký tự định nghĩa bằng đường gấp khúc có thể dễ dàng thay đổi kích thước hơn là các ký tự định nghĩa bằng mẫu các

pixel.

- Khoảng cách giữa các kí tự.
- Sự canh lề: canh trái (left text), canh phải (right text), canh giữa (center text), canh hai lề (justify text).
- Cách hiển thị tuần tự của các kí tự: có thể là phải sang trái, từ trên xuống dưới, từ trái sang phải, từ dưới lên trên.
- Hướng của kí tự [2]

2.2. Các thuật toán vẽ đường thẳng

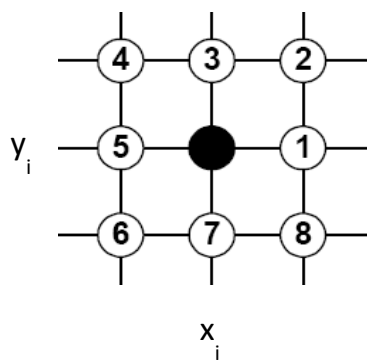
2.2.1 Mở đầu

Ta cần vẽ đoạn thẳng có hai điểm đầu cuối là $P1(x_1, y_1)$, $P2(x_2, y_2)$ lên trên một thiết bị hiển thị (màn hình), có nghĩa là ta cần phải tính toán xem pixel nào sẽ lần lượt được hiển thị sao cho đoạn thẳng hiển thị “giống” với đoạn thẳng thực muốn vẽ nhất.

Tọa độ các pixel được hiển thị là các điểm nguyên sau khi xấp xỉ đối tượng thực lần lượt là (x_i, y_i) , $i = 0, 1 \dots$

Vậy, nếu biết được (x_i, y_i) là tọa độ nguyên xác định ở bước thứ i , điểm nguyên tiếp theo (x_{i+1}, y_{i+1}) sẽ được xác định như thế nào?

Nhận xét rằng: để đối tượng hiển thị trên lưới nguyên được liền nét, các điểm mà (x_{i+1}, y_{i+1}) có thể chọn chỉ là một trong 8 điểm được đánh số từ 1 đến 8 trong hình 1.



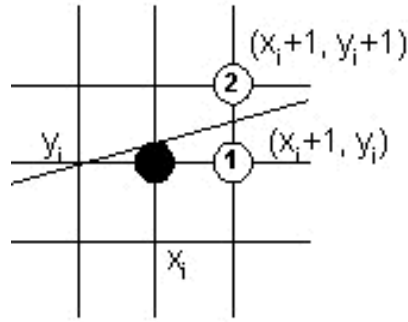
Hình 2.1: Các điểm lân cận của (x_i, y_i)

Ta có tọa độ các điểm là:

$1(x_i + 1, y_i)$	$3(x_i, y_i + 1)$	$5(x_i - 1, y_i)$	$7(x_i, y_i - 1)$
$2(x_i + 1, y_i + 1)$	$4(x_i - 1, y_i + 1)$	$6(x_i - 1, y_i - 1)$	$8(x_i + 1, y_i - 1)$

Phương của đường thẳng sẽ cho ta gợi ý khi chọn một trong 8 điểm trên. Cách chọn các điểm như thế nào sẽ tùy thuộc vào từng thuật toán trên cơ sở xem xét tới vấn đề tối ưu tốc độ.

Xét đường thẳng có hệ số góc $0 < m < 1$ và $Dx > 0$.



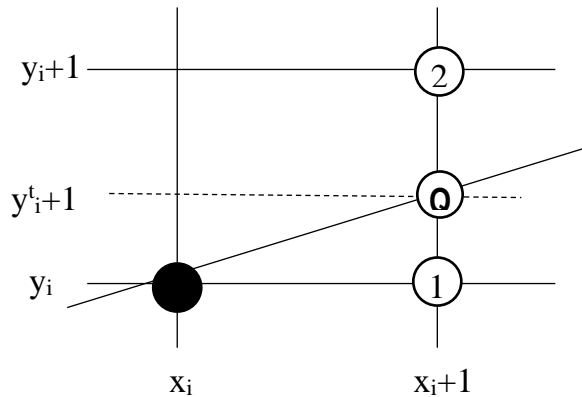
Hình 2.2: Đường thẳng có hệ số góc $0 < m < 1$

Với các đường thẳng dạng này, nếu (x_i, y_i) là điểm đã xác định được ở bước thứ i (điểm màu đen) thì điểm cần chọn (x_{i+1}, y_{i+1}) ở bước thứ $(i+1)$ sẽ là một trong hai điểm $1(x_i + 1, y_i)$ hoặc $2(x_i + 1, y_i + 1)$ như hình 2 trên.

Như vậy, trong trường hợp $0 < m < 1$, ta có: $x_{i+1} = x_i + 1, y_{i+1} \in \{y_i, y_i + 1\}$

2.2.2 Thuật toán DDA (Digital Differential Analyzer)

Với thuật toán DDA, việc quyết định chọn y_{i+1} là y_i hay $y_i + 1$, dựa vào phương trình $y = mx + b$ của đoạn thẳng.



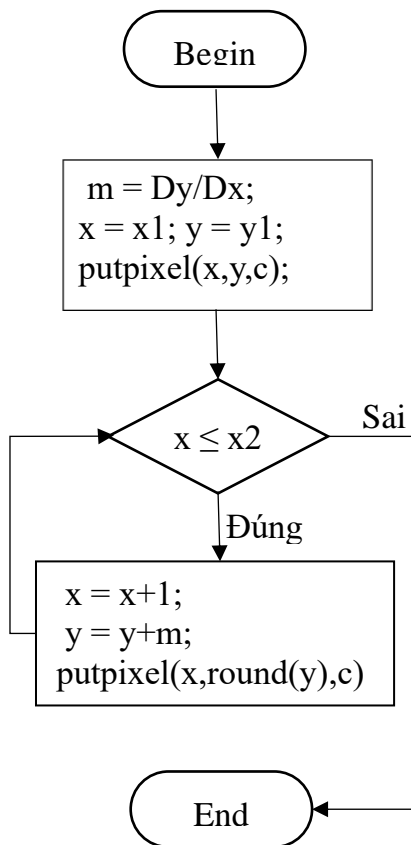
Hình 2.3: Minh họa thuật toán DDA trường hợp $0 < m < 1$

$Q(x_{i+1}, y_{i+1}^t)$ thuộc đường thẳng thực, ta có:

$$\begin{aligned} y_{i+1}^t &= mx_{i+1} + b \\ &= m(x_i + 1) + b \\ &= mx_i + b + m \\ \Rightarrow y_{i+1}^t &= y_i^t + m \end{aligned}$$

y_{i+1} là giá trị sau khi làm tròn y_{i+1}^t , vậy $y_{i+1} = \text{round}(y_{i+1}^t)$

Do vậy: $y_{i+1} = \text{round}(y_i^t + m)$



Hình 2.3: Lưu đồ thuật toán vẽ đường thẳng DDA

Cài đặt minh họa:

```

void DDA_line (int x1, int y1, int x2, int y2, int c)
{
    float m = (float)(y2-y1)/(x2-x1);
    int x=x1;
    float y=y1;
    putpixel(x,round(y),c);
    for(int i=x1; i<=x2; i++)
    {
        x++;
        y=y+m;
        putpixel(x,round(y),c);
    }
}
  
```

Nhận xét:

- Việc sử dụng công thức $y_{i+1}^t = y_i^t + m$ để tính giá trị y tại mỗi bước đã giúp cho thuật toán DDA cải thiện tốc độ so với cách tính y từ phương trình $y = mx + b$ do khử được phép nhân trên số thực.

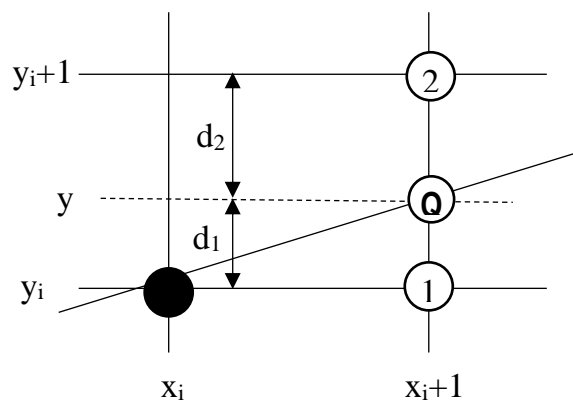
- Tuy nhiên vẫn còn tồn tại phép toán cộng số thực và phép làm tròn.

Để cải thiện tốc độ hơn nữa, ta phải thực hiện loại bỏ được các phép toán trên tập số thực như thuật toán Bresenham sau đây.[1][3]

2.2.3 Thuật toán Bresenham

Thuật toán Bresenham đưa ra cách chọn y_{i+1} là y_i hay $y_i + 1$ một hướng khác, sao cho có thể tối ưu hóa về mặt tốc độ so với thuật toán DDA.

Q thuộc đường thẳng thực, ta có: $y = mx_{i+1} + b \rightarrow y = m(x_i + 1) + b$



Hình 2.4: Minh họa thuật toán Bresenham, $0 < m < 1$

Đặt: $d1 = y - y_i$

$d2 = y_{i+1} - y$

$P_i = Dx(d1 - d2)$

$= Dx(2y - 2y_i - 1)$

Do $m = Dy/Dx$, ta có:

$P_i = 2x_iDy - 2y_iDx + c$ với $c = 2Dy + (2b - 1)Dx$

Nhận xét rằng do $Dx > 0$ nên dấu của biểu thức $d1 - d2$ cũng chính là dấu của P_i . Hay nói một cách khác, nếu tại bước thứ i ta xác định được dấu của P_i thì xem như ta xác định được điểm cần chọn ở bước $(i+1)$. Vấn đề còn lại là làm thế nào để tính được P_i tại mỗi bước thật nhanh.

Ta có: $P_{i+1} = 2x_{i+1}Dy - 2y_{i+1}Dx + c$

Tính: $P_{i+1} - P_i = 2Dy(x_{i+1} - x_i) - 2Dx(y_{i+1} - y_i)$ do $x_{i+1} = x_i + 1$, suy ra:

$P_{i+1} - P_i = 2Dy - 2Dx(y_{i+1} - y_i)$

- Nếu $P_i < 0 \Leftrightarrow d1 < d2 \rightarrow$ điểm Q “gần” điểm 1 hơn nên bước thứ $i+1$ vẽ điểm 1,
 Vậy $y_{i+1}=y_i \rightarrow P_{i+1} - P_i = 2Dy$

- Nếu $P_i \geq 0 \Leftrightarrow d1 \geq d2 \rightarrow$ điểm Q “gần” điểm 2 hơn nên bước thứ $i+1$ vẽ điểm 2,
 Vậy $y_{i+1}=y_i+1 \rightarrow P_{i+1} - P_i = 2Dy - 2Dx$

Giá trị P_1 được tính từ điểm vẽ đầu tiên (x_1, y_1) theo công thức :

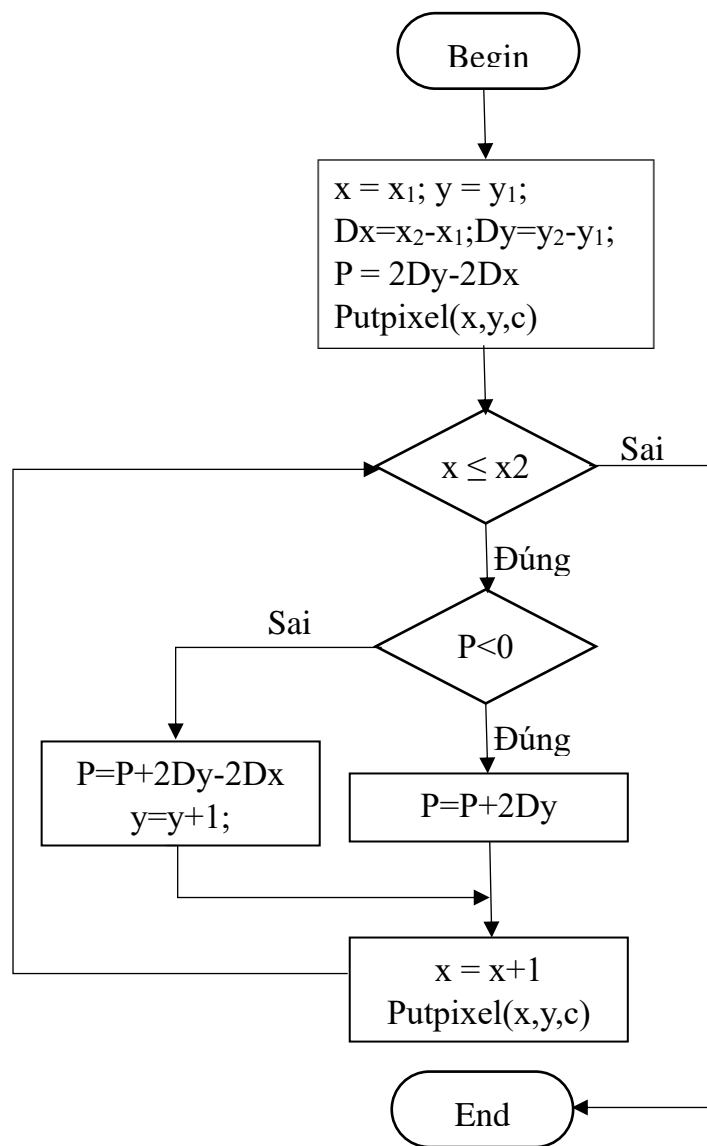
$$P_1 = 2Dyx_1 - 2Dxy_1 + c = 2Dyx_1 - 2Dxy_1 + 2Dy + (2b - 1)Dx$$

$$\Rightarrow P_1 = 2Dy - Dx + 2Dyx_1 - 2Dxy_1 + 2b$$

Do (x_1, y_1) là điểm nguyên thuộc đoạn thẳng, ta có $y_1 = mx_1 + b$, $m=Dy/Dx$, Thế vào phương trình trên, suy ra :

$$P_1 = 2Dy - Dx.$$

Lưu đồ giải thuật Bresenham:



Hình 2.5: Lưu đồ thuật toán vẽ đường thẳng Bresenham

Cài đặt minh họa:

```
void Bre_line(int x1, int y1, int x2, int y2, int c)
{
    int x, y, dx, dy, P, C1, C2;
    y = y1;
    Dx = x2 - x1;
    Dy = y2 - y1;
    P = 2*Dy - Dx;
    C1 = 2*Dy;
    C2 = 2*(Dy - Dx);
    for (x=x1; x<=x2; x++)
    {
        putpixel(x, y, c);
        if (P < 0)
            P += C1;
        else
        {
            P += C2;
            y++;
        }
    }
}
```

Nhận xét:

- Thuật toán Bresenham đã khử được các phép toán trên tập số thực
- Các thao tác trên tập số thực là phép cộng.

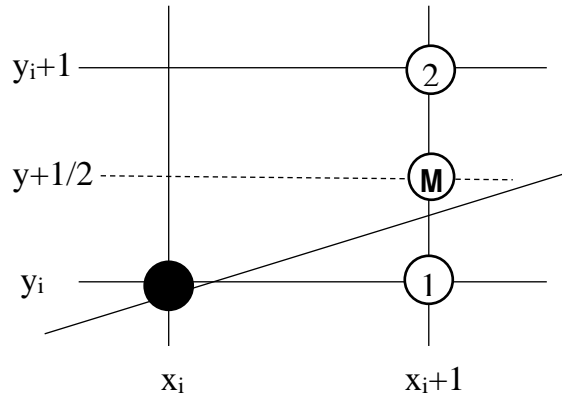
Do đó, thuật toán Bresenham đã cải thiện tốc độ đáng kể so với thuật toán DDA.[1]

2.2.4 Thuật toán Midpoint vẽ đường thẳng

Thuật toán đưa ra cách chọn y_{i+1} là y_i+1 hay y_i bằng cách so sánh vị trí tương đối giữa trung điểm $M(x_i+1, y_i + 1/2)$ của điểm 1 và 2 với đường thẳng thực.

- Nếu điểm M nằm phía trên đường thẳng thực, ta chọn vẽ điểm 1 hay $y_{i+1}=y_i$ (1)

- Ngược lại, nếu điểm M thuộc hoặc nằm phía dưới đường thẳng thực, ta chọn vẽ điểm 2 hay $y_{i+1}=y_i+1$ (2)



Hình 2.6: Minh họa thuật toán Midpoint vẽ đường thẳng

Từ phương trình của đường thẳng $y = mx + b$, $m = Dx/Dy$, suy ra: $x Dy - y Dx + c = 0$

Đặt $F(x,y) = x Dy - y Dx + c$. Ta có nhận xét sau:

- Nếu M thuộc đường thẳng $\Rightarrow F(M) = 0$ (3)

- Nếu M nằm phía trên đường thẳng $\Rightarrow F(M) < 0$ (4)

- Nếu M nằm phía dưới đường thẳng $\Rightarrow F(M) > 0$ (5)

Đặt $P_i = F(M) = F(x_{i+1}, y_i + 1/2)$,

$$\Rightarrow P_i = 2(x_{i+1})Dy - 2(y_i + 1/2)Dx + 2c$$

Tính: $P_{i+1} - P_i = 2Dy(x_{i+1} - x_i) - 2Dx(y_{i+1} - y_i)$ do $x_{i+1} = x_i + 1$, ta có:

$$P_{i+1} - P_i = 2Dy - 2Dx(y_{i+1} - y_i)$$

- Từ (1) và (4) ta có: Nếu $P_i < 0$ bước thứ $i+1$ vẽ điểm 1,
Vậy $y_{i+1}=y_i \rightarrow P_{i+1} - P_i = 2Dy$ (a)

- Từ (2), (3), (5) ta có: Nếu $P_i \geq 0$ bước thứ $i+1$ vẽ điểm 2,
Vậy $y_{i+1}=y_i+1 \rightarrow P_{i+1} - P_i = 2Dy - 2Dx$ (b)

Giá trị P_1 được tính từ điểm vẽ đầu tiên (x_1, y_1) theo công thức :

$$P_1 = F(x_1 + 1, y_1 + 1/2) = 2(x_1 + 1)Dy - 2(y_1 + 1/2)Dx + 2c$$

$$P_1 = 2(x_1 Dy - y_1 Dx + c) + 2Dy - Dx,$$

do (x_1, y_1) thuộc đường thẳng nên $x_1 Dy - y_1 Dx + c = 0$

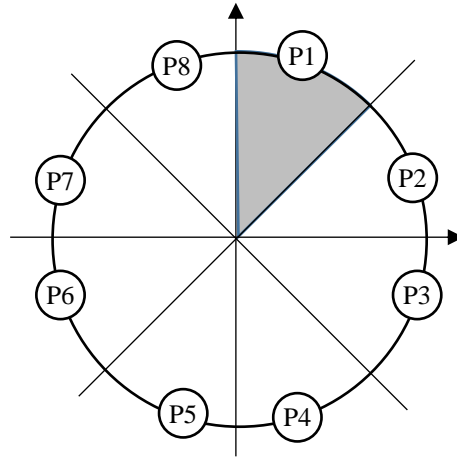
$$\text{vậy suy ra } P_1 = 2Dy - Dx \quad (c)$$

Nhận xét: từ (a), (b), (c) ta có thuật toán MidPoint cho kết quả tương tự như thuật toán Bresenham

2.3. Thuật toán Midpoint vẽ đường tròn

Do tính đối xứng của đường tròn nên ta chỉ cần khảo sát trên 1/8 đường tròn, sau đó lấy đối xứng.

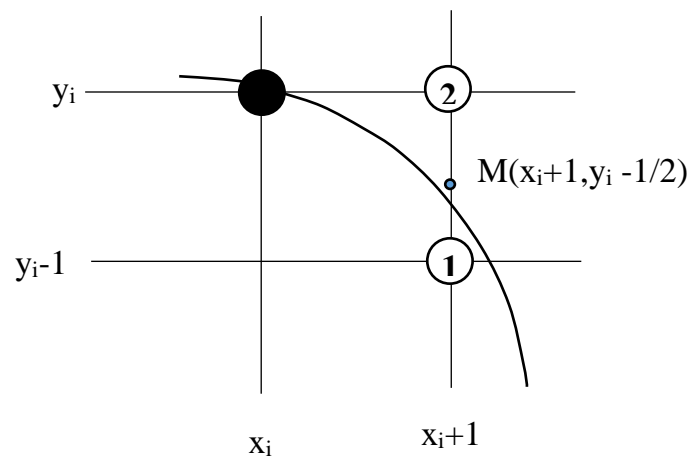
Giả sử $P1(x,y)$ là một điểm bất kỳ trên cung được khảo sát (cung tô màu xám). Các vị trí đối xứng với P qua các trục tọa độ và các đường phân giác là $(\pm x, \pm y)$ và $(\pm y, \pm x)$.



Hình 2.7: Các vị trí đối xứng của đường tròn

Chọn điểm bắt đầu vẽ là điểm $(0,R)$, nếu điểm (x_i, y_i) là điểm nguyên được vẽ ở bước thứ i thì bước thứ $i+1$ có thể chọn vẽ điểm $1(x_i+1, y_i-1)$ hoặc $2(x_i+1, y_i)$

Vậy tại bước $i+1$ ta có:

$$\begin{cases} x_{i+1} = x_i + 1 \\ y_{i+1} \in \{y_i, y_i - 1\} \end{cases}$$


Hình 2.8: Minh họa thuật toán Midpoint vẽ đường tròn

Tương tự như thuật toán MidPoint vẽ đoạn thẳng, việc quyết định chọn một trong hai điểm 1 và 2 sẽ được thực hiện thông qua việc xét dấu của hàm $F(x,y)$ tại điểm M là trung điểm nằm giữa chúng. Trong đó $F(x,y) = x^2 + y^2 - R^2$.

Ta có nhận xét:

- $F(x,y) > 0$ nếu điểm (x,y) nằm ngoài đường tròn
- $F(x,y) = 0$ nếu điểm (x,y) nằm trên đường tròn
- $F(x,y) < 0$ nếu điểm (x,y) nằm trong đường tròn

Đặt $P_i = F(M) = (x_i+1)^2 + (y_i - 1/2)^2 - R^2$

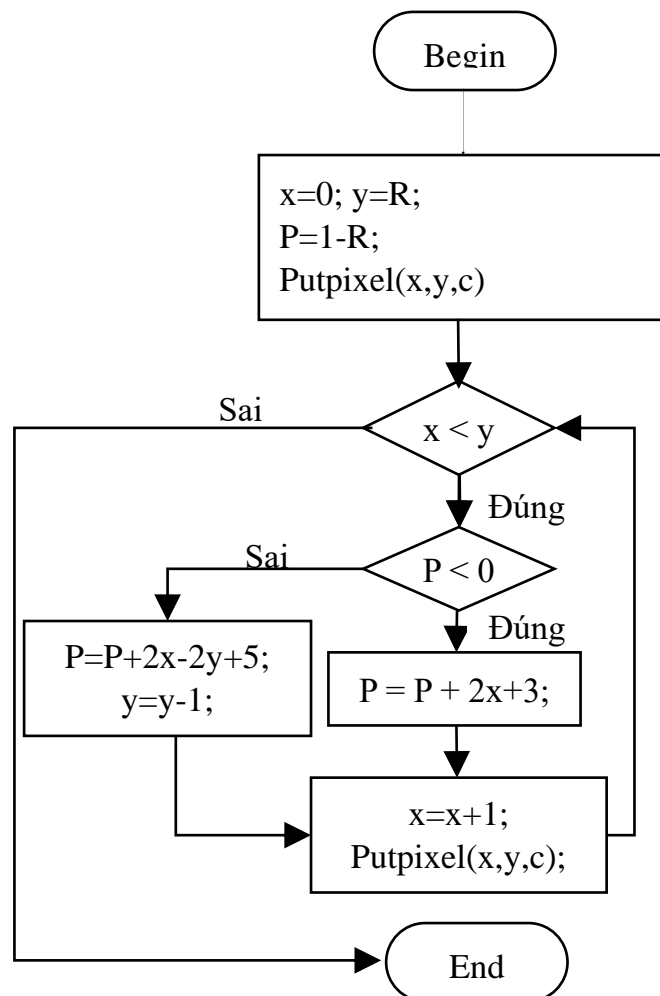
Tính $P_{i+1} - P_i = (x_{i+1}+1)^2 + (y_{i+1} - 1/2)^2 - (x_i+1)^2 - (y_i - 1/2)^2$

$$= 2x_i + 3 + (y_{i+1}^2 - y_i^2) - (y_{i+1} - y_i)$$

- Nếu $P_i = F(M) \geq 0$ thì M nằm trên hoặc nằm phía ngoài đường tròn, vậy bước thứ $i+1$ vẽ điểm 1 hay $y_{i+1} = y_i - 1$, suy ra: $P_{i+1} = P_i + 2x_i - 2y_i + 5$
- Nếu $P_i < 0$ thì M nằm phía trong đường tròn, bước thứ $i+1$ vẽ điểm 2 hay $y_{i+1} = y_i$,
Suy ra: $P_{i+1} = P_i + 2x_i + 3$

Tính P_1 ứng với điểm ban đầu $(0,R)$:

$$P_1 = F(0+1, R-1/2) = 5/4 - R \approx 1-R$$



Hình 2.9: Lưu đồ giải thuật Midpoint vẽ đường tròn

Cài đặt minh họa:

```
void MP_circle(int R)
{
    int x, y;
    x = 0;
    y = R;
    P = 1 - R;
    PutPixel(x, y,c);
    while (x < y)
    {
        if (P < 0)
            P += 2*x + 3;
        else
        {
            P += 2*(x -y) + 5;
            y--;
        } x++;
        PutPixel(x, y,c);
    }
}
```

2.4 Các phép biến đổi cơ sở của đối tượng 2 chiều

2.4.1 Mở đầu

a, Phép biến đổi:

Một trong những ưu điểm quan trọng của đồ họa là cho phép dễ dàng thao tác lên các đối tượng đã được tạo ra. Một nhà quản lí có nhu cầu thu nhỏ các biểu đồ trong một báo cáo, một kiến trúc sư muốn nhìn tòa nhà ở những góc nhìn khác nhau, một nhà thiết kế muốn quan sát và chỉnh sửa các mẫu đối tượng trong quá trình thiết kế, ... Tất cả các thao tác này có thể được hỗ trợ một cách dễ dàng nhờ vào các phép biến đổi hình học.

Các phép biến đổi hình học sẽ làm thay đổi mô tả về tọa độ của các đối tượng, từ đó làm cho đối tượng bị thay đổi về hướng, kích thước và hình dạng. Các phép biến đổi hình học cơ sở bao gồm : tịnh tiến (translation), quay (rotation) và biến đổi tỉ lệ (scaling).

Một phép biến đổi là một ánh xạ T được định nghĩa:

$$T : R^2 \rightarrow R^2$$

$$P(x,y) \mapsto Q(x,y)$$

Nói cách khác, T là hàm số $T(x, y)$ theo hai biến (x, y) :

$$\begin{cases} x' = f(x, y) \\ y' = g(x, y) \end{cases}$$

Phép biến đổi Affine là phép biến đổi với $f(x, y)$ và $g(x, y)$ là các hàm tuyến tính. Phép biến đổi này có dạng :

$$\begin{cases} x' = ax + cy + e \\ y' = bx + dy + f \end{cases} \quad \text{với } a, b, c, d, e, f \in \mathbb{R}, \quad ad - bc \neq 0$$

b, Hệ tọa độ thuần nhất và ma trận thuần nhất:

Tọa độ thuần nhất của một điểm trên mặt phẳng được biểu diễn bằng bộ ba số tỉ lệ (x_h, y_h, h) không đồng thời bằng 0 và liên hệ với các tọa độ (x, y) của điểm đó bởi công thức:

$$x = x_h/h, \quad y = y_h/h \quad \text{với } h \neq 0$$

Nếu một điểm có tọa độ thuần nhất là (x, y, t) thì nó cũng có tọa độ thuần nhất là $(h.x, h.y, h.t)$ trong đó h là số thực khác 0 bất kì. Để đơn giản, ta chọn $h = 1$, lúc này mỗi điểm $P(x, y)$ sẽ được biểu diễn dưới dạng tọa độ thuần nhất là $(x, y, 1)$.

Một phép biến đổi tổng quát:

$$\begin{cases} x' = ax + cy + e \\ y' = bx + dy + f \end{cases} \quad \text{với } a, b, c, d, e, f \in \mathbb{R}, \quad ad - bc \neq 0$$

sẽ có ma trận thuần nhất tổng quát tương ứng là:

$$M = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{pmatrix}$$

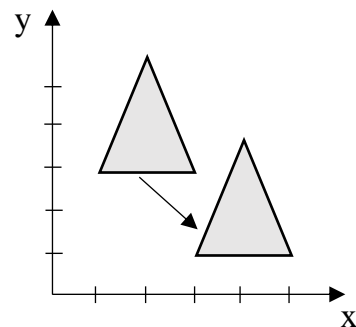
2.4.2 Phép tịnh tiến

Phép tịnh tiến thực hiện di chuyển vị trí của một điểm $P(x, y)$ tới vị trí $Q(x', y')$ theo vector độ dời (dx, dy) , ta có:

$$\begin{cases} x' = x + dx \\ y' = y + dy \end{cases}$$

Ma trận của phép biến đổi:

$$M_T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{pmatrix}$$



Hình 2.10: Minh họa phép tịnh tiến

Phép biến đổi tịnh tiến được biểu diễn như sau: $Q = P.M_T$

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{pmatrix}$$

Ví dụ:

Tịnh tiến điểm $P(6,1)$ theo vector độ dời $(-4,2)$ được ảnh $Q(x',y')$, ta có:

$$\begin{cases} x' = 6 - 4 \\ y' = 1 + 2 \end{cases} \Rightarrow \text{vậy } Q(2,3)$$

Chúng ta có thể dịch chuyển toàn bộ một đối tượng bằng cách áp dụng quy tắc trên cho mọi điểm thuộc đối tượng. Để tịnh tiến một đoạn thẳng, ta thực hiện tịnh tiến hai điểm đầu và cuối của nó rồi sau đó vẽ lại đoạn thẳng nối hai điểm mới. Với đa giác, ta tịnh tiến các đỉnh của nó sau đó vẽ lại đa giác với các đỉnh mới. Tương tự, để tịnh tiến các đối tượng như đường tròn, ellipse, ta tịnh tiến tâm của chúng tới vị trí mới rồi vẽ lại. [1]

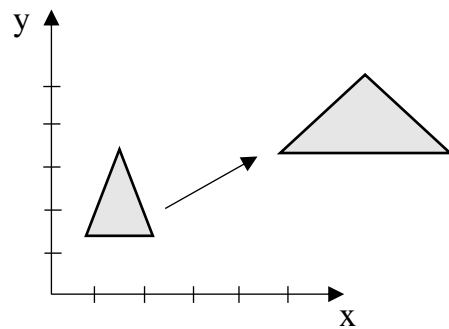
2.4.3 Phép biến đổi tỷ lệ tâm $O(0,0)$

Phép biến đổi tỉ lệ làm thay đổi kích thước, hình dạng của đối tượng. Co hay giãn tọa độ của một điểm $P(x, y)$ theo trục hoành và trục tung với tỷ lệ lần lượt là S_x và S_y được điểm $Q(x',y')$:

$$\begin{cases} x' = x.S_x \\ y' = y.S_y \end{cases}$$

Ma trận của phép biến đổi:

$$M_s = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Hình 2.11: Minh họa phép biến đổi tỷ lệ

Nhận xét:

- Nếu $S_x, S_y > 1$ ta có phép phóng to, và ảnh Q của P sẽ rời xa gốc tọa độ hơn P .
- Nếu $S_x, S_y < 1$ ta có phép thu nhỏ và ảnh Q sẽ tiến lại gần gốc tọa độ hơn P .
- Nếu $S_x = S_y$ ta có phép biến đổi đồng dạng.

Phép biến đổi tỷ lệ được biểu diễn như sau: $Q = P.M_S$

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.4.4 Phép quay tâm O(0,0)

Phép quay làm thay đổi hướng của đối tượng. Một phép quay cần có tâm quay và góc quay. Góc quay dương được quy ước là chiều ngược chiều kim đồng hồ. Ta có công thức biến đổi của phép quay điểm $P(x, y)$ quanh tâm $O(0,0)$ một góc α :

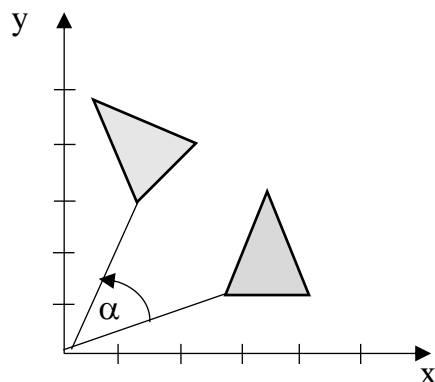
$$\begin{cases} x' = x \cdot \cos\alpha - y \cdot \sin\alpha \\ y' = x \cdot \sin\alpha + y \cdot \cos\alpha \end{cases}$$

Ma trận của phép biến đổi:

$$M_R = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Phép quay được biểu diễn như sau: $Q = P.M_R$

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Hình 2.12: Minh họa phép quay tâm O

Ví dụ: Ảnh của điểm P(22,14) sau phép quay 90^0 quanh tâm O(0,0) là Q(x',y'):

$$(x' \ y' \ 1) = (22 \ 14 \ 1) \cdot \begin{pmatrix} \cos 90^0 & \sin 90^0 & 0 \\ -\sin 90^0 & \cos 90^0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (-14 \ 22 \ 1) \text{ vậy } Q(-14,22)$$

2.4.5. Kết hợp các phép biến đổi.

Với một đối tượng đồ họa cho trước ta có thể thực hiện nhiều phép biến đổi lên chúng để đạt được hình ảnh mong muốn. Quá trình áp dụng các phép biến đổi liên tiếp để tạo nên một phép biến đổi tổng thể được gọi là sự kết hợp các phép biến đổi.

a, Kết hợp các phép tịnh tiến

Nếu ta thực hiện phép tịnh tiến lên P(xP, yP) được Q(xQ,yQ) , rồi lại thực hiện tiếp một phép tịnh tiến khác lên Q, ta được điểm R(xR,yR). Như vậy, R là ảnh của phép biến đổi kết hợp hai phép tịnh tiến liên tiếp:

$$R = Q.M_{T2} = (P.M_{T1}).M_{T2} = P.(M_{T1}.M_{T2})$$

$$(x_R \ y_R \ 1) = (x_P \ y_P \ 1) \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ d_{1x} & d_{1y} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ d_{2x} & d_{2y} & 1 \end{pmatrix}$$

$$\Rightarrow (x_R \ y_R \ 1) = (x_P \ y_P \ 1) \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ d_{1x}+d_{2x} & d_{1y}+d_{2y} & 1 \end{pmatrix}$$

Vậy: Kết hợp của 2 hay nhiều phép tịnh tiến là một phép tịnh tiến có độ dời bằng tổng các độ dời thành phần.

b, Kết hợp các phép biến đổi tỷ lệ

Tương tự như phép tịnh tiến, ta thực hiện biến đổi tỷ lệ lên điểm P(xP, yP) được Q(xQ,yQ), rồi lại thực hiện phép biến đổi tỷ lệ khác lên Q, ta được điểm R(xR,yR). Như vậy, R là ảnh của kết hợp hai phép biến đổi tỷ lệ liên tiếp:

$$R = Q.M_{S2} = (P.M_{S1}).M_{S2} = P.(M_{S1}.M_{S2})$$

$$(x_R \ y_R \ 1) = (x_P \ y_P \ 1) \cdot \begin{pmatrix} S_{1x} & 0 & 0 \\ 0 & S_{1y} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_{2x} & 0 & 0 \\ 0 & S_{2y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(x_R \ y_R \ 1) = (x_P \ y_P \ 1) \cdot \begin{pmatrix} S_{1x} \cdot S_{2x} & 0 & 0 \\ 0 & S_{1y} \cdot S_{2y} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Vậy: Kết hợp của 2 hay nhiều phép biến đổi tỷ lệ là một phép biến đổi tỷ lệ có tỷ lệ biến đổi bằng tích các tỷ lệ biến đổi thành phần.

c, Kết hợp các phép quay tâm O(0,0):

Kết hợp của 2 phép quay, ta có:

$$R = Q \cdot M_{R2} = (P \cdot M_{R1}) \cdot M_{R2} = P \cdot (M_{R1} \cdot M_{R2})$$

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} \cos\alpha_1 & \sin\alpha_1 & 0 \\ -\sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha_2 & \sin\alpha_2 & 0 \\ -\sin\alpha_2 & \cos\alpha_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} \cos(\alpha_1+\alpha_2) & \sin(\alpha_1+\alpha_2) & 0 \\ -\sin(\alpha_1+\alpha_2) & \cos(\alpha_1+\alpha_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Vậy: Kết hợp của 2 phép quay tâm O là một phép quay tâm O có góc quay bằng tổng góc quay thành phần. Mở rộng ra ta có kết hợp của nhiều phép quay tâm O là một phép quay tâm O có góc quay bằng tổng các góc quay thành phần.

d, Kết hợp các phép biến đổi khác nhau

Các phép biến đổi tác động lên một đối tượng đồ họa đều là sự kết hợp của các phép biến đổi cơ sở theo một trình tự nào đó. Ở đây, *thứ tự thực hiện các phép biến đổi lên một đối tượng khác nhau sẽ cho ảnh khác nhau*, điều này được dẫn đến do phép nhân 2 ma trận không có tính chất giao hoán.

Ví dụ a: Cho một điểm P(2,3), thực hiện biến đổi tỷ lệ lên P theo tỷ số $S_x=1.5$, $S_y = 2$ được ảnh Q1. Quay Q1 một góc 90° độ quanh O(0,0) được điểm ảnh R1:

$$\text{ta có: } R1 = Q1 \cdot M_R = P \cdot (M_S \cdot M_R)$$

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow (x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} Sx \cdot \cos\alpha & Sx \cdot \sin\alpha & 0 \\ -Sy \cdot \sin\alpha & Sy \cdot \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(x' \ y' \ 1) = (2 \ 3 \ 1) \cdot \begin{pmatrix} 1.5 \cdot 0 & 1.5 \cdot 1 & 0 \\ -2 \cdot 1 & 2 \cdot 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (-6 \ 3 \ 1).$$

Vậy ảnh cuối cùng của chuỗi biến đổi lên điểm P(2,3) là R1(-6,3) (a)

Ví dụ b: Quay P(2,3) một góc 90⁰ độ quanh O(0,0) được điểm ảnh Q2, thực hiện biến đổi tỷ lệ Q2 theo tỷ số Sx=1.5, Sy = 2 được ảnh R2:

ta có: R2 = Q2.M_S = P.(M_R.M_S):

$$(x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow (x' \ y' \ 1) = (x \ y \ 1) \cdot \begin{pmatrix} Sx \cdot \cos\alpha & Sy \cdot \sin\alpha & 0 \\ -Sx \cdot \sin\alpha & Sy \cdot \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(x' \ y' \ 1) = (2 \ 3 \ 1) \cdot \begin{pmatrix} 1.5 \cdot 0 & 2 \cdot 1 & 0 \\ -1.5 \cdot 1 & 2 \cdot 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (-4.5 \ 4 \ 1).$$

Vậy, ảnh cuối cùng của chuỗi biến đổi lên điểm P(2,3) là R2(-4.5,4) (b)

Từ (a) và (b) ta thấy R1(-6,3) ≠ R2(-4.5,4) do thứ tự thực hiện các phép biến đổi lên một đối tượng khác nhau sẽ cho ảnh khác nhau.

2.5. Một số tính chất của phép biến đổi affine.

2.5.1. Bảo toàn đường thẳng:

Ảnh của một đường thẳng qua phép biến đổi Affine là một đường thẳng.

Vậy, để biến đổi một đường thẳng qua hai điểm P₁, P₂ ta chỉ cần áp dụng phép biến đổi cho hai điểm P₁, P₂ rồi vẽ lại đường thẳng qua hai điểm mới.

Thật vậy, ta có phương trình tham số của đường thẳng qua hai điểm P_1, P_2 là :

$$P(t) = (1-t)P_1 + tP_2 .$$

$Q(t)$ là các điểm ảnh nhận được sau khi thực hiện phép biến đổi có ma trận M :

$$Q(t) = P(t).M = [(1-t)P_1 + tP_2]M = (1-t)P_1M + tP_2M$$

Nếu gọi Q_1, Q_2 lần lượt là ảnh của P_1, P_2 qua phép biến đổi M , ta sẽ có $Q_1 = P_1M, Q_2 = P_2M$. Lúc này $Q(t) = (1-t)Q_1 + tQ_2$. Đây chính là dạng của phương trình tham số đoạn thẳng qua Q_1, Q_2 .

Từ kết quả trên, để biến đổi một đoạn thẳng đi qua hai điểm P_1 và P_2 , ta chỉ cần áp dụng phép biến đổi cho hai điểm P_1, P_2 rồi vẽ lại đoạn thẳng qua hai điểm mới.

2.5.2 Bảo toàn tính song song

Ảnh của hai đường thẳng song song là hai đường thẳng song song.

Giả sử có 2 đường thẳng song song L đi qua P_1 và P_2 và L' đi qua P_1' và P_2' . Ta có phương trình tham số:

$$L = P_1 + (P_2 - P_1)t = P_1 + \beta t \text{ và}$$

$$L' = P_1' + (P_2' - P_1')t = P_1' + \beta' t$$

Do $L \parallel L'$ nên hai đường thẳng cùng phương, hay $\beta = \beta'$. Khi áp dụng phép biến đổi có ma trận M lên hai đường thẳng, dễ dàng nhận ra ảnh của chúng sẽ có phương βM do đó chúng song song.

Một hệ quả quan trọng của tính chất này đó là ảnh của các hình bình hành sau phép biến đổi là các hình bình hành.

Do phép biến đổi Affine không bảo toàn góc nên ảnh của một hình chữ nhật hay một hình vuông sau biến đổi luôn là một hình bình hành.

2.5.3 Bảo toàn tính tỷ lệ

Giả sử C là điểm chia đoạn AB theo tỉ số t . Nếu A', B', C' lần lượt là ảnh A, B, C qua phép biến đổi thì C' cũng sẽ chia $A'B'$ theo tỉ số t .

Như vậy, nếu AM là trung tuyến cạnh BC của tam giác ABC thì $A'M'$ cũng là trung tuyến cạnh $B'C'$ của tam giác $A'B'C'$. Trong đó A', B', C', M' lần lượt là ảnh của A, B, C, M .

2.6 Một số phép biến đổi khác.

2.6.1 Phép biến đổi tỷ lệ tâm bất kỳ

Có thể xem phép biến đổi tỷ lệ tâm bất kỳ $I(x,y)$ một điểm P theo các tỷ lệ S_x, S_y là một phép biến đổi được kết hợp từ các phép biến đổi cơ sở sau:

- Tịnh tiến theo vector $(-x, -y)$ đưa tâm biến đổi về gốc tọa độ, ta có ma trận của phép biến đổi là $M_{T(-I)}$

- Thực hiện phép biến đổi tâm $O(0,0)$ theo tỷ lệ S_x, S_y , ma trận biến đổi là M_S

- Tịnh tiến theo vector (x,y) đưa tâm biến đổi I về vị trí ban đầu (x,y), ma trận biến đổi là $M_{T(I)}$

Ta có ma trận của phép biến đổi như sau:

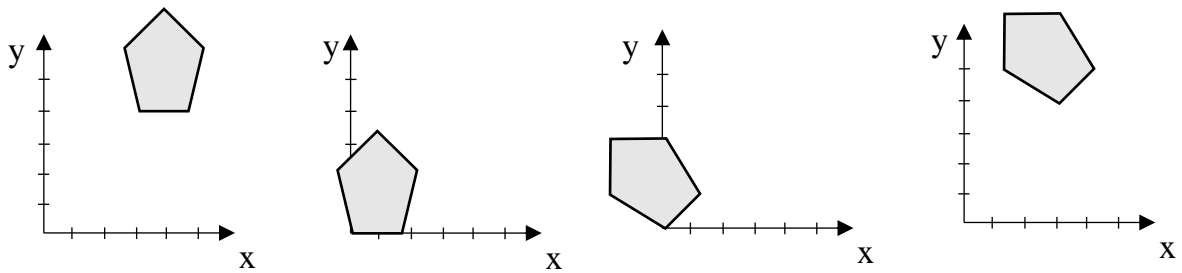
$$M = M_{T(-I)} \cdot M_S \cdot M_{T(I)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x & -y & 1 \end{pmatrix} \begin{pmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x & y & 1 \end{pmatrix}$$

$$\Rightarrow M = \begin{pmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ (1-Sx).x & (1-Sy).y & 1 \end{pmatrix}$$

2.6.2 Phép quay có tâm quay bất kỳ

Tương tự như phép biến đổi tỷ lệ tâm bất kỳ, có thể xem phép quay quanh tâm I(x,y) một góc α được kết hợp từ các phép biến đổi cơ sở sau:

- Tịnh tiến theo vector (-x,-y) đưa tâm quay về gốc tọa độ, ta có ma trận của phép biến đổi là $M_{T(-I)}$
- Thực hiện phép quay tâm O(0,0) một góc α , ma trận biến đổi là M_R
- Tịnh tiến theo vector (x,y) đưa tâm quay về vị trí ban đầu: $M_{T(I)}$



Hình 2.13: Minh họa phép quay có tâm quay bất kỳ

Ta có ma trận của phép biến đổi như sau:

$$M = M_{T(-I)} \cdot M_R \cdot M_{T(I)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x & -y & 1 \end{pmatrix} \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x & y & 1 \end{pmatrix}$$

$$\Rightarrow M = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ (1-\cos\alpha).x+y.\sin\alpha & -x.\sin\alpha+(1-\cos\alpha).y & 1 \end{pmatrix}$$

2.6.3 Phép đối xứng

Ta có ma trận M của các phép đối xứng qua các trục tọa độ Ox, Oy, và tâm O(0,0) lần lượt là:

$$M_{Ox} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad M_{Oy} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad M_O = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2.6.4 Phép biến đổi ngược

Phép biến đổi ngược dùng để undo một phép biến đổi đã thực hiện.

- Q là ảnh của P qua phép biến đổi T có ma trận biến đổi M là: $Q = P.M$
- Phép biến đổi ngược T^{-1} sẽ có ma trận biến đổi là M^{-1} , với M^{-1} là ma trận nghịch đảo của M, như vậy: $P = Q.M^{-1}$

Ta có ma trận tổng quát của các phép biến đổi là: $M = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{pmatrix}$

Với giả thiết ban đầu $ad - bc \neq 0$, ta tính được ma trận nghịch đảo của M là:

$$M^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b & 0 \\ -c & a & 0 \\ cf-de & be-af & ad-bc \end{pmatrix}$$

2.7 Các thuật toán xén đoạn thẳng.

Tại một thời điểm nhất định, người sử dụng chỉ muốn hiển thị một phần hình ảnh đã được tạo ra, phần còn lại phải được xác định và xén bỏ. Ví dụ khi thực hiện các thao tác phóng to (zoom) hình ảnh, chỉ một phần của nó có thể được hiển thị, vì vậy không cần thiết và tốn thời gian để tính toán các phần khác của ảnh, nên chúng được xác định và xén bỏ.

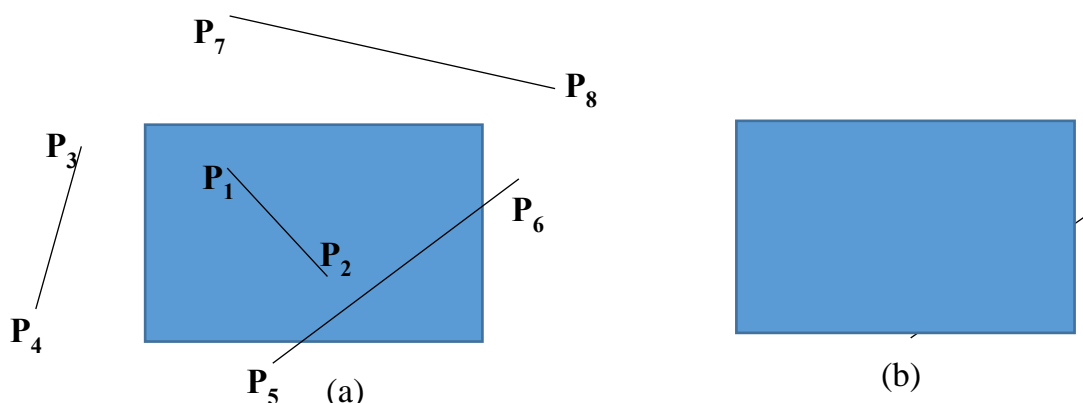
Thao tác xác định các điểm ảnh nằm trong hay ngoài một vùng cho trước được gọi là xén hình.

Vùng được dùng để xén hình gọi là cửa sổ xén (clip window). Tùy thuộc vào từng ứng dụng cụ thể mà cửa sổ xén có thể có dạng là đa giác hay là đường cong khép kín. Trong phần này chúng ta sẽ khảo sát các thuật toán có cửa sổ xén là một hình chữ nhật.

Giả sử cửa sổ xén là cửa sổ hình chữ nhật có tọa độ của các điểm dưới trái và điểm trên phải lần lượt là (x_{\min}, y_{\min}) và (x_{\max}, y_{\max}) . Một điểm $P(x, y)$ được coi là nằm bên trong cửa sổ nếu thỏa mãn hệ bất phương trình:

$$\begin{cases} x_{\min} \leq x \leq x_{\max} \\ y_{\min} \leq y \leq y_{\max} \end{cases}$$

Thao tác xén hình là một trong những thao tác cơ bản của quá trình hiển thị đối tượng, do đó vấn đề tối ưu tốc độ luôn là đích cho các thuật toán nhắm đến. Ý tưởng chung của các thuật toán xén đoạn thẳng đó là loại bỏ phép toán tính giao điểm giữa đoạn thẳng với biên của cửa sổ, như đối với các đoạn thẳng đặc biệt nằm hoàn toàn trong hoặc hoàn toàn bên ngoài cửa sổ (ví dụ như đoạn P_1P_2 , P_3P_4 và P_7P_8 trong hình 15). Đối với các đoạn thẳng có khả năng cắt cửa sổ, cần phải đưa ra cách tính giao điểm thật nhanh.



Hình 2.14: (a) Trước khi thực hiện xén đoạn thẳng, (b) sau khi thực hiện

Ta thấy, các đoạn thẳng mà có cả hai điểm đầu cuối nằm hoàn toàn trong cửa sổ thì cả đoạn thẳng nằm trong cửa sổ, đây cũng chính là kết quả sau khi xén (đoạn thẳng P_1P_2), mặt khác đối với các đoạn thẳng mà có hai điểm đầu cuối nằm về cùng một phía của cửa sổ thì luôn nằm ngoài cửa sổ và sẽ bị mất sau khi xén (đoạn thẳng P_3P_4 , P_7P_8). Với các đoạn thẳng có khả năng cắt cửa sổ (đoạn thẳng P_5P_6) để việc tính giao điểm nhanh cần loại bỏ việc xác định giao điểm ảo với các biên kéo dài.

2.7.1 Thuật toán Cohen-Sutherland

Đây là một trong những thuật toán ra đời sớm nhất và thông dụng nhất. Bằng cách kéo dài các biên của cửa sổ, người ta chia mặt phẳng thành chín vùng gồm cửa sổ và tám vùng xung quanh nó.

a, Khái niệm mã vùng (area code)

Một con số 4 bit nhị phân gọi là mã vùng sẽ được gán cho mỗi vùng để mô tả vị trí tương đối của vùng đó so với cửa sổ. Bằng cách đánh số từ 1 đến 4 theo thứ tự từ phải qua trái, các bit của mã vùng được dùng theo quy ước để xác định vị trí tương đối của vùng so với cửa sổ:

4	3	2	1	0101	0100	0110
Bottom	Top	Right	Left	0001	0000	0010
				1001	1000	1010

Hình 2.15: Cách đánh mã vùng – thuật toán Cohen-Sutherland

Điểm $P(x,y)$ có một mã vùng 4 bit C_P , bit thứ i ($i=1, .. 4$) bằng 1 chỉ ra điểm P ở vị trí tương ứng với cửa sổ. Các giá trị bit trong mã vùng được tính bằng cách xác định tọa độ của điểm $P(x, y)$:

Bit 1 được đặt là 1 nếu $x < x_{\min}$, khi đó bit 2 được đặt bằng 0.

Bit 2 được đặt là 1 nếu $x > x_{\max}$, khi đó bit 1 được đặt bằng 0

Bit 3 được đặt là 1 nếu $y > y_{\max}$, khi đó bit 4 được đặt bằng 0.

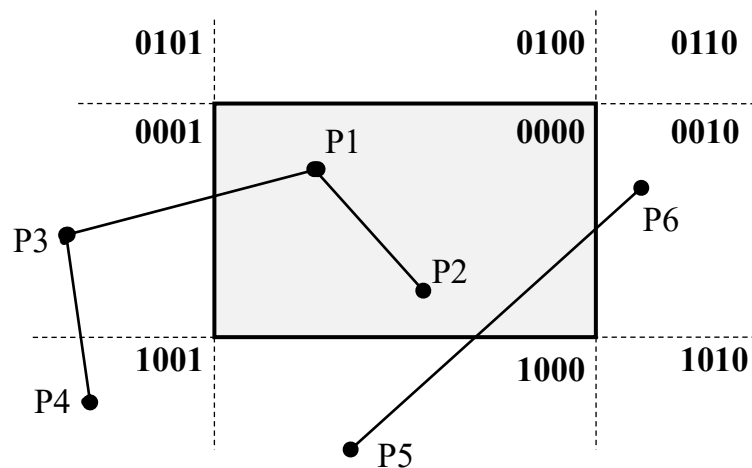
Bit 4 được đặt là 1 nếu $y < y_{\min}$, khi đó bit 3 được đặt bằng 0

b, Ý tưởng thuật toán:

Gán mã C_1, C_2 cho các điểm đầu cuối P_1, P_2 của đoạn thẳng cần xén. Ta có các trường hợp:

- Nếu $C_1=C_2=0000$ ($C_1 \mid \mid C_2=0000$) thì P_1P_2 nằm hoàn toàn bên trong cửa sổ.
- Nếu tại cùng một vị trí bit của C_1 và C_2 đều là 1 ($C_1 \& C_2 \neq 0000$) thì P_1P_2 nằm cùng một phía với cửa sổ, vậy nằm hoàn toàn ngoài cửa sổ.
- Ngoài hai trường hợp trên: Tìm giao điểm $P'(x',y')$ của đoạn thẳng với biên của cửa sổ. Lúc này, đoạn thẳng ban đầu được xén thành $P_1P', P'P_2$.

- Lắp lại thao tác đã xét cho các đoạn thẳng mới cho tới khi xác định được phần nằm trong hoặc loại bỏ toàn bộ đoạn thẳng.



Hình 2.16: Minh họa thuật toán Cohen - Sutherland

Tìm giao điểm $P'(x', y')$ của P_1P_2 với biên của cửa sổ:

- Nếu bit 1 là 1 ta có $x' = x_{\min}$, bit 2 là 1: $x' = x_{\max}$ P_1P_2 cắt cửa sổ tại biên dọc.
- Nếu bit 3 là 1 ta có $y' = y_{\max}$, bit 4 là 1: $y' = y_{\min}$ P_1P_2 cắt cửa sổ tại biên ngang.[2]

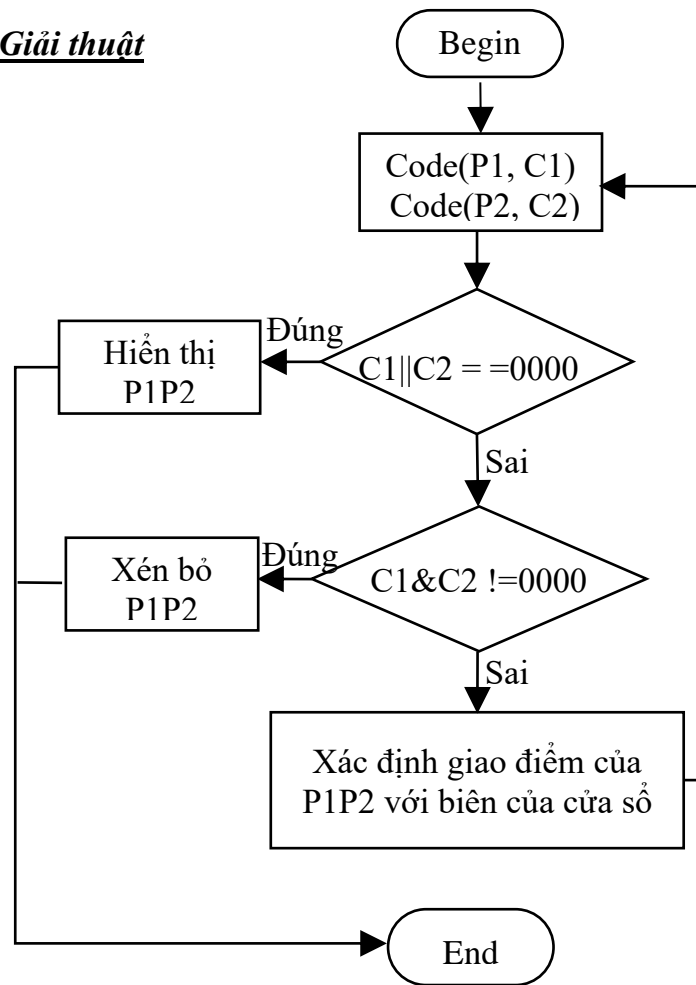
Biên ngang:

$$\begin{cases} y' = y_{\min} \text{ hoặc } y' = y_{\max} \\ x' = x_1 + (y' - y_1)/m \end{cases}$$

Biên dọc:

$$\begin{cases} x' = x_{\min} \text{ hoặc } x' = x_{\max} \\ y' = y_1 + m(x' - x_1) \end{cases}$$

Giải thuật



Hình 2.17: Giải thuật xén đoạn thẳng Cohen-Sutherland

Ví dụ: Cho cửa sổ xén có tọa độ góc trái dưới là (12, 15), tọa độ góc phải trên là (32, 47). Thực hiện xén các đoạn thẳng sau:

- a, A1(8,25)A2(10,50)
- b, B1(18,20)B2(28,40)
- c, C1(21,44)C2(29,52)
- d, D1(8,20)D2(30,53)

Giải:

a, Mã vùng $C_{A1} = 0001$, $C_{A2} = 0101$

Ta có: $C_{A1} \& C_{A2} != 0000$ vậy A1A2 nằm hoàn toàn ngoài cửa sổ.

A1A2 nằm hoàn toàn về bên trái của cửa sổ, do $C_{A1} \& C_{A2} = 0001$: bit 1 bằng 1 tương ứng với vị trí bên trái của cửa sổ.

b, Mã vùng $C_{B1} = 0000$, $C_{B2} = 0000$

Ta có: $C_{B1} || C_{B2} = 0000$ vậy B1B2 nằm hoàn toàn trong cửa sổ.

c, Mã vùng $C_{C1} = 0000$, $C_{C2} = 0100$

Ta có: $C_{C1} || C_{C2} != 0000$ và $C_{C1} \& C_{C2} = 0000$, suy ra C1C2 cắt cửa sổ.

Do bit 3 của C_{C2} bằng 1 nên tọa độ cắt được tính như sau:

$$\begin{cases} y' = y_{\max} \\ x' = x_1 + (y' - y_1)/m \end{cases} \quad \begin{cases} y' = 47 \\ x' = 21 + (47-44).(29-21)/(52-44) \end{cases}$$

với $1/m = Dx/Dy = (29-21)/(52-44)$

Vậy $C1C2$ cắt cửa sổ tại $I(24,47)$

d, Mã vùng $C_{D1} = 0001$, $C_{D2} = 0100$

Ta có: $C_{D1} \parallel C_{D2} = !0000$ và $C_{D1} \& C_{D2} = 0000$, suy ra $D1D2$ cắt cửa sổ.

Do bit 1 của C_{D1} bằng 1 nên tọa độ cắt được tính như sau:

$$\begin{cases} x' = x_{\min} \\ y' = y_1 + (x' - x_1).m \end{cases} \quad \begin{cases} x' = 12 \\ y' = 20 + (12-8).(53-20)/(30-8) \end{cases}$$

$m = Dy/Dx = (53-20)/(30-8) = 1.5$

$$\Rightarrow I1(12,26)$$

Do bit 3 của C_{D2} bằng 1 nên tọa độ cắt được tính như sau:

$$\begin{cases} y' = y_{\max} \\ x' = x_1 + (y' - y_1)/m \end{cases} \quad \begin{cases} y' = 47 \\ x' = 8 + (47-20)/1.5 \end{cases}$$

$$\Rightarrow I2(26,47)$$

Vậy $D1D2$ cắt cửa sổ tại 2 điểm $I1(12,26)$ và $I2(26,47)$

2.7.2 Thuật toán Liang Barsky

Thuật toán Liang-Barsky được phát triển dựa vào việc phân tích dạng tham số của phương trình đoạn thẳng.

Cho 2 điểm $P_1(x_1, y_1)$, $P_2(x_2, y_2)$. Phương trình tham số của đoạn thẳng có 2 điểm đầu cuối P_1 , P_2 là:

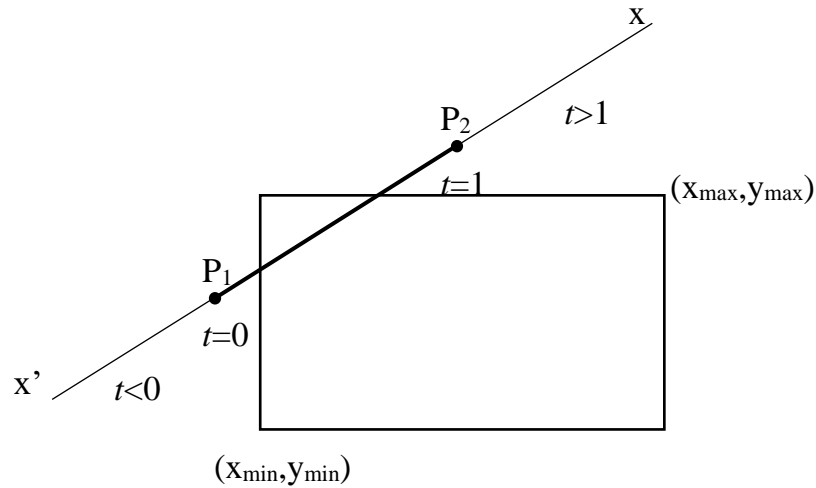
$$\begin{cases} x = x_1 + (x_2 - x_1)t \\ y = y_1 + (y_2 - y_1)t \end{cases} \quad \Leftrightarrow \quad \begin{cases} x = x_1 + tDx \\ y = y_1 + tDy \end{cases}$$

Với $t \in [0, 1]$ ta có phương trình tham số của đoạn P_1P_2

Với $t \in (-\infty, +\infty)$ ta có phương trình tham số của đường thẳng đi qua P_1 , P_2 .

Ứng với mỗi giá trị t , ta sẽ có một điểm $P(x, y)$ tương ứng thuộc đường thẳng:

- Các điểm ứng với $t \geq 1$ sẽ thuộc về tia P_2x .
- Các điểm ứng với $t \leq 0$ sẽ thuộc về tia P_1x' .
- Các điểm ứng với $0 \leq t \leq 1$ sẽ thuộc về đoạn thẳng P_1P_2 .



Hình 2.18: Minh họa thuật toán Liang Barsky

Một điểm $P(x,y)$ là giao điểm của đoạn thẳng và cửa sổ ứng với các giá trị t thỏa hệ bất phương trình:

$$\begin{cases} x_{\min} \leq x \leq x_{\max} \\ y_{\min} \leq y \leq y_{\max} \\ 0 \leq t \leq 1 \end{cases} \Rightarrow \begin{cases} x_{\min} \leq x_1 + t Dx \leq x_{\max} \\ y_{\min} \leq y_1 + t Dy \leq y_{\max} \\ 0 \leq t \leq 1 \end{cases}$$

Đặt:

$$\begin{aligned} p_1 &= -Dx & q_1 &= x_1 - x_{\min} \\ p_2 &= Dx & q_2 &= x_{\max} - x_1 \\ p_3 &= -Dy & q_3 &= y_1 - y_{\min} \\ p_4 &= Dy & q_4 &= y_{\max} - y_1 \end{aligned}$$

Như vậy việc tìm đoạn giao thực chất là tìm nghiệm của hệ bất phương trình này. Có hai khả năng xảy ra đó là:

- Hệ bất phương trình vô nghiệm: đường thẳng không có phần giao với cửa sổ nên sẽ bị loại bỏ.
- Hệ bất phương trình có nghiệm: tập nghiệm là các giá trị t thỏa $t \in [t_1, t_2] \subseteq [0, 1]$.
Ta xét các trường hợp :
 - Nếu $\exists k \in \{1, 2, 3, 4\}$: $p_k = 0$ thì đoạn thẳng song song với biên của cửa sổ, khi đó:
 - Nếu $q_k < 0 \Rightarrow$ Hệ bất phương trình vô nghiệm (đoạn thẳng không cắt cửa sổ)
 - Nếu $q_k \geq 0 \Rightarrow$ Tính nghiệm t_1, t_2 như 2 trường hợp sau:
 - Với bất phương trình $t p_k \leq q_k$ mà $p_k < 0$, ta có $t \geq q_k / p_k$, tính $t_1 = \max\{0, q_k/p_k\}$

- Với bất phương trình $tp_k \leq q_k$ mà $p_k > 0$, ta có $t \leq q_k / p_k$, tính $t2 = \min\{1, q_k/p_k\}$
Nếu $t1 > t2$ thì đoạn thẳng nằm hoàn toàn ngoài cửa sổ.

$$\text{Nếu } t1 \leq t2: \begin{cases} x = x_1 + t Dx \\ y = y_1 + t Dy \end{cases} \quad [2][3]$$

Cài đặt minh họa:

```
int ClipTest(int p, int q, float &t1, float &t2)
```

```
{
float r;
if (p<0)
{
r = float(q)/p;
if (r>t2)
return FALSE;
else
if (r>t1)
t1 = r;
} else
{
if (p>0)
{
r = float(q)/p;
if (r<t1)
return FALSE;
else
if (r<t2)
t2 = r;
} else // p=0
{
if (q<0)
return FALSE;
}
}
return TRUE;
}
```

```
int LiangBarskyClipping(POINT P1, POINT P2, RECT R, POINT *Q1, POINT *Q2)
{
```



```

float t1, t2;
int Dx, Dy, x1, y1, x2, y2, xmin, ymin, xmax, ymax;
t1 = 0;
t2 = 1;
x1 = P1.x; y1 = P1.y;
x2 = P2.x; y2 = P2.y;
Dx = x2 - x1; Dy = y2 - y1;
xmin = R.Left; ymin = R.Top;
xmax = R.Right; ymax = R.Bottom;
if (ClipTest(-Dx, x1 - xmin, t1, t2)) // Giai he bat phuong trinh 1
{
    if (ClipTest(Dx, xmax - x1, t1, t2)) // Giai he bat phuong trinh 2
    {
        if (ClipTest(-Dy, y1 - ymin, t1, t2)) // Giai he bat phuong trinh 3
        {
            if (ClipTest(Dy, ymax - y1, t1, t2)) // Giai he bat pt 4
            {
                Q1.x = x1 + t1. Dx;
                Q1.y = y1 + t1. Dy;
                Q2.x = x1 + t2. Dx;
                Q2.y = y1 + t2. Dy;
                return TRUE;
            } // Giai he bat phuong trinh 4
        } // Giai he bat phuong trinh 3
    } // Giai he bat phuong trinh 2
} // Giai he bat phuong trinh 1
return FALSE;
} // LiangBarskyClipping

```

Ví dụ 1: Cho cửa sổ xén có tọa độ góc trái dưới là (5,7), góc phải trên là (25,30) và đoạn thẳng P_1P_2 có tọa độ là $P_1(10,11)$, $P_2(27,28)$. Tìm vị trí tương đối của P_1P_2 với cửa sổ và giao điểm P' (nếu có) của P_1P_2 với cửa sổ:

Giải:

Ta có: $p_1 = -17$ $q_1 = 5$

$p_2 = 17$ $q_2 = 15$

$p_3 = -17$ $q_3 = 4$

$p_4 = 17$ $q_4 = 19$

$p_1, p_3 < 0 \Rightarrow t_l = \max\{0, -5/17, -4/17\} = 0$

$$p_2, p_4 > 0 \Rightarrow t_2 = \min \{1, 15/17, 19/17\} = 15/17$$

$$t_1 < t_2 \Rightarrow x = 10 + 17 \cdot 15/17 = 25$$

$$y = 11 + 17 \cdot 15/17 = 26$$

$\Rightarrow P_1P_2$ cắt cửa sổ tại $P'(25,26)$

Ví dụ 2: Cho cửa sổ xén có tọa độ góc trái dưới là (6,4), góc phải trên là (20,18) và đoạn thẳng P_1P_2 có tọa độ là $P_1(5,15)$, $P_2(9,19)$. Tìm vị trí tương đối của P_1P_2 với cửa sổ và giao điểm P' (nếu có) của P_1P_2 với cửa sổ.

Giải:

$$\text{Ta có: } p_1 = -4 \quad q_1 = -1$$

$$p_2 = 4 \quad q_2 = 15$$

$$p_3 = -4 \quad q_3 = 11$$

$$p_4 = 4 \quad q_4 = 3$$

$$p_1, p_3 < 0 \Rightarrow t_1 = \max \{0, 1/4, -11/4\} = 1/4$$

$$p_2, p_4 > 0 \Rightarrow t_2 = \min \{1, 15/4, 3/4\} = 3/4$$

$$t_1 < t_2 \Rightarrow P_1' \begin{cases} x_1 = 5 + 4 \cdot 1/4 = 6 \\ y_1 = 15 + 4 \cdot 1/4 = 16 \end{cases}$$

$$P_2' \begin{cases} x_1 = 5 + 4 \cdot 3/4 = 8 \\ y_1 = 15 + 4 \cdot 3/4 = 18 \end{cases}$$

$\Rightarrow P_1P_2$ cắt cửa sổ tại $P_1'(6,16)$ và $P_2'(8,18)$

Câu hỏi ôn tập chương:

- 1, Trình bày thuật toán DDA, Bresenham, Midpoint vẽ đường thẳng.
- 2, Thuật toán Bresenham và Midpoint có cải tiến gì so với thuật toán DDA.
- 3, Trình bày ý tưởng và thuật toán Midpoint vẽ đường tròn.
- 4, Trình bày về các phép biến đổi cơ bản: Phép tịnh tiến, phép quay, phép biến đổi tỷ lệ.
- 5, Xây dựng ma trận của phép biến đổi tỷ lệ tâm bất kỳ và phép quay tâm bất kỳ.
- 6, Trình bày các tính chất của phép biến đổi Affine.
- 7, Trình bày thuật toán xén đoạn thẳng Liang-Barsky và thuật toán Cohen-Sutherland.

Bài tập ôn tập:

- 1, Tìm các điểm được vẽ của đoạn thẳng đi qua 2 điểm đầu cuối $P_1(17,3)$, $P_2(23,15)$ bằng cả 3 thuật toán và so sánh kết quả.
- 2, Tìm các điểm được vẽ của đoạn thẳng đi qua 2 điểm đầu cuối $P_1(2,3)$, $P_2(11,9)$ bằng cả 3 thuật toán và so sánh kết quả.
- 3, Tìm các điểm được vẽ của cung tròn thứ nhất của hình tròn có bán kính $R=21$.
- 4, Tìm ảnh của điểm $P(5,7)$ qua phép tịnh tiến có độ dời theo trục Ox là $dx=-3$, theo trục Oy là $dy=10$.
- 5, Tìm ảnh của điểm $P(8,3)$ qua phép quay tâm $O(0,0)$ một góc $\alpha = 90^\circ$.
- 6, Tìm ảnh của điểm $P(19,23)$ qua phép quay tâm $I(2,4)$ một góc $\alpha = -60^\circ$.
- 7, Tìm ảnh của điểm $P(7,3)$ qua phép biến đổi tỷ lệ tâm $O(0,0)$ với tỷ lệ biến đổi $S_x=1.5$, $S_y=0.5$.
- 8, Tìm ảnh của điểm $P(22,16)$ qua phép biến đổi tỷ lệ tâm $I(7,11)$ với tỷ lệ biến đổi $S_x=2.5$, $S_y=2$.
- 9, Tìm ảnh của $P(3,4)$ qua phép quay 90° rồi biến đổi theo tỷ lệ $S_x=2$, $S_y=3.3$.
- 10, Tìm ảnh của $P(7,19)$ qua phép biến đổi theo tỷ lệ $S_x=2.5$, $S_y=1.2$ rồi quay 45° .
- 11, Thực hiện xén đoạn thẳng P_1P_2 bằng thuật toán Cohen-Sutherland, biết tọa độ $P_1(13,28)$, $P_2(27,42)$ và tọa độ góc trái dưới và góc phải trên của cửa sổ xén lần lượt là $(15,31)$, $(40,43)$.
- 12, Thực hiện xén đoạn thẳng P_1P_2 bằng thuật toán Liang-Barsky, biết tọa độ $P_1(60,54)$, $P_2(69,72)$ và tọa độ góc trái dưới và góc phải trên của cửa sổ xén lần lượt là $(66,41)$, $(70,68)$.
- 13, Tìm câu trả lời đúng nhất :
Phép biến đổi Affine luôn biến :
 - a, Một góc vuông thành một góc vuông
 - b, Một góc bẹt thành một góc bẹt
 - c, Một góc nhọn thành một góc nhọn

d, Một góc tù thành một góc tù

14, Trọn phát biểu đúng nhất:

Thuật toán vẽ đường thẳng DDA đã loại bỏ được:

- a, Các phép nhân và phép cộng trên tập số thực
- b, Phép nhân trên tập số thực
- c, Phép cộng trên tập số thực
- d, Các phép toán trên tập số thực

15, Trọn phát biểu đúng nhất:

Hạn chế của thuật toán DDA là:

- a, Còn tồn tại phép nhân và phép cộng trên tập số thực
- b, Còn tồn tại phép nhân và phép làm tròn trên tập số thực
- c, Còn tồn tại phép làm tròn và phép cộng trên tập số thực
- d, Còn tồn tại phép làm tròn trên tập số thực

16, Tìm câu trả lời đúng nhất :

Phép biến đổi Affine biến :

- a, Một hình chữ nhật thành một hình chữ nhật
- b, Một hình chữ nhật thành một hình vuông
- c, Một hình chữ nhật thành một hình bình hành
- d, Một hình chữ nhật thành một hình thoi.

17, Trong phép biến đổi Affine, kết hợp của các phép tịnh tiến là một phép tịnh tiến, có:

- a, véctor độ dời bằng tổng các véctor độ dời thành phần
- b, véctor độ dời bằng hiệu các véctor độ dời thành phần
- c, véctor độ dời bằng tích các véctor độ dời thành phần
- d, Cả ba câu trên đều sai.

18, Trong phép biến đổi Affine, kết hợp của các phép biến đổi tỷ lệ là một phép biến đổi tỷ lệ, có:

- a, Tỷ lệ biến đổi bằng tổng các tỷ lệ thành phần
- b, Tỷ lệ biến đổi bằng tích các tỷ lệ thành phần
- c, Tỷ lệ biến đổi bằng hiệu các tỷ lệ thành phần
- d, Cả ba câu trên đều sai

19, Trong phép biến đổi Affine, kết hợp của các phép quay là một phép quay, có:

- a, Góc quay bằng tổng các góc quay thành phần
- b, Góc quay bằng tích các góc quay thành phần
- c, Góc quay bằng hiệu các góc quay thành phần

d, Cả ba câu trên đều sai

20, Tìm câu trả lời đúng nhất :

Phép biến đổi Affine biến :

- a, Một đường thẳng thành một điểm
- b, Một đường thẳng thành một đường thẳng
- c, một đường thẳng thành một mặt phẳng
- d, cả ba ý trên đều sai.

21. Trong mặt phẳng, tìm phương án không đúng khi biểu diễn một điểm:

- a. $[a \ b \ 0]$
- b. $[a \ b \ 1]$
- c. $[a \ b \ 2]$
- d. $[a \ b \ 3]$

22. Trong mặt phẳng, tìm phương án đúng khi biểu diễn một điểm:

- a. $[a \ 0 \ 0]; a \neq 0$
- b. $[0 \ a \ 0]; a \neq 0$
- c. $[0 \ 0 \ a]; a \neq 0$
- d. $[0 \ 0 \ 0]$

23. Trong 2D tọa độ vector của điểm P(3,6) vậy trong các tọa độ sau tọa độ nào không phải là tọa độ đồng nhất của P:

- a. (6, 12)
- b. (6, 12, 2)
- c. (1.5, 3, 0.5)
- d. (3*a, 6*a, a) trong đó $a \neq 0$

24. Trong 2D tọa độ vector của điểm P(4,8) vậy trong các tọa độ sau tọa độ nào không phải là tọa độ đồng nhất của P:

- a. (8, 16, 2)
- b. (6, 12, 1.5)
- c. (5, 10, 2)
- d. (3*a, 6*a, a) trong đó $a \neq 0$

25. Phép biến đổi Affine trong mặt phẳng là: $x' = ax + cy + e$; $y' = bx + dy + f$; Vậy phép đối xứng qua tâm O là:

- a. $b=c=e=f=1$ và $a=d=0$
- b. $b=c=e=f=0$ và $a=d=1$
- c. $b=c=e=f=0$ và $a=d=-1$
- d. $b=c=e=f=-1$ và $a=d=0$

26. Phép biến đổi Affine trong mặt phẳng là: $x' = ax + cy + e$; $y' = bx + dy + f$; Vậy phép làm béo một đối tượng gấp 2 lần ban đầu là:

- a. $b=c=0$, $e=f=1$, $a=d=2$
- b. $b=c=1$, $e=f=0$ và $a=d=2$
- c. $b=c=2$ và $a=d=e=f=0$
- d. $b=c=e=f=0$ và $a=d=2$

27. Phép biến đổi Affine trong mặt phẳng là: $x' = ax + cy + e$; $y' = bx + dy + f$; Vậy ma trận của phép tịnh tiến là:

a. $b=c=0$, $a=d=1$, $e=dx$ và $f=dy$

b. $b=c=e=f=0$, $a=Sx$ và $d=Sy$

c. $b=c=0$, $e=f=1$, $a=d=1$

d. $b=c=1$, $e=dx$ $f=dy$ và $a=d=0$

28. Phép biến đổi Affine trong mặt phẳng là: $x'=ax+cy+e$; $y'=bx+dy+f$; Vậy ma trận của phép của phép biến đổi tỷ lệ là:

a. $b=c=0$, $a=d=1$, $e=dx$ và $f=dy$

b. $b=c=e=f=0$, $a=Sx$ và $d=Sy$

c. $b=c=0$, $e=f=1$ $a=Sx$ và $d=Sy$

d. $b=c=1$, $e=dx$ $f=dy$ và $a=d=0$

29. Phép biến đổi Affine trong mặt phẳng là: $x'=ax+cy+e$; $y'=bx+dy+f$; Vậy ma trận của phép của phép quay là:

a. $a=\sin\alpha$, $b=-\sin\alpha$, $c=d=\cos\alpha$ và $e=f=0$

b. $a=\sin\alpha$, $c=-\sin\alpha$, $b=d=\cos\alpha$ và $e=f=0$

c. $b=\sin\alpha$, $c=-\sin\alpha$, $a=d=\cos\alpha$ và $e=f=0$

d. $c=\sin\alpha$, $b=-\sin\alpha$, $a=d=\cos\alpha$ và $e=f=0$

30. Cho cửa sổ cắt tia góc trái dưới (2,12) và góc phải trên (14,27). Mã vùng 4bit của điểm A(22,6) là:

a. 1000

b. 1010

c. 0101

d. 0001

31. Cho cửa sổ cắt tia có góc trái dưới (2,13) và phải trên (5,26), mã vùng 4bit của điểm M(0,20) là:

a. 1010

b. 1100

c. 0101

d. 0001

Bài tập thảo luận:

- 1, Trình bày thuật toán DDA, Bresenham, Midpoint vẽ đường thẳng trong các trường hợp còn lại của hệ số góc m .
- 2, Cài đặt các thuật toán vẽ đường thẳng và đường tròn, hình chữ nhật
- 3, Xây dựng ma trận của phép lấy đối xứng qua tâm bất kỳ
- 4, Xây dựng ma trận của các phép biến đổi ngược của các phép biến đổi cơ sở.
- 5, Cài đặt các thuật toán Cohen-Sutherland và Liang-Barsky.

CHƯƠNG 3: ĐỒ HỌA 3 CHIỀU

Mục tiêu của chương: Sinh viên cần nắm được cách biểu diễn đối tượng đồ họa 3 chiều từ các đối tượng đơn giản như các hình khối, các đa diện ...

Có thể thực hiện tính toán được ảnh của các điểm trong không gian 3 chiều sau các phép biến đổi: Tỷ lệ, tịnh tiến, phép quay tâm $O(0,0,0)$ và quay tâm bất kỳ, các phép lấy đối xứng qua mặt phẳng, qua trục tọa độ.

Sinh viên cần nắm được các phương pháp chiếu các đối tượng 3D xuống mặt phẳng thiết bị hiển thị. Có thể tính được hình chiếu của các điểm trong một số phương pháp chiếu song song và phối cảnh.

3.1. Tổng quan về đồ họa ba chiều

Các đối tượng trong thế giới thực phần lớn là các đối tượng ba chiều, nên việc thể hiện các đối tượng ba chiều trên máy tính là một công việc hết sức cần thiết để đưa tin học gần gũi với thực tế hơn. Cũng giống như các cách biểu diễn các đối tượng ba chiều trên mặt phẳng ở các thiết bị khác (như của máy ảnh, camera,...), biểu diễn bằng máy tính cũng phải tuân theo các quy luật về phối cảnh, sáng, tối,... nhằm giúp người xem có thể tưởng tượng lại hình ảnh một cách gần đúng nhất. Ngoài ra biểu diễn trên máy tính có ưu thế giúp ta có thể quan sát đối tượng ở nhiều góc cạnh khác nhau, ở các khoảng cách khác nhau.

Khi chúng ta mô hình hóa và hiển thị một cảnh ba chiều, ta cần phải xem xét rất nhiều khía cạnh và vấn đề khác nhau chứ không đơn giản là thêm vào tọa độ thứ ba cho các đối tượng.

Bề mặt đối tượng có thể xây dựng bởi nhiều tổ hợp khác nhau của các mặt phẳng và các mặt cong. Ngoài ra, đôi khi chúng ta cũng cần mô tả một số thông tin về bên trong các đối tượng. Các công cụ hỗ trợ đồ họa (graphics package) thường cung cấp một số hàm hiển thị các thành phần bên trong, những đường nét tiêu biểu hoặc hiển thị một phần của đối tượng ba chiều (solid object). Ngoài ra, các phép biến đổi hình học trong đồ họa 3 chiều thường được sử dụng nhiều hơn và đa dạng hơn trong đồ họa hai chiều. Phép biến đổi hệ quan sát trong không gian ba chiều phức tạp hơn nhiều so với trong không gian hai chiều do chúng ta phải chọn lựa nhiều tham số hơn khi mô tả một cảnh ba chiều sẽ xuất hiện trên màn hình như thế nào.

3.2. Biểu diễn các đối tượng ba chiều

Các cảnh đồ họa có thể chứa nhiều dạng đối tượng khác nhau: cây, hoa, mây, núi, nước, sắt thép, cỏ,... Chính vì vậy, có nhiều phương pháp khác nhau được sử dụng để mô tả các đối tượng sao cho phù hợp với thuộc tính của các loại đối tượng này.

Sơ đồ biểu diễn một đối tượng lập thể thường được chia ra làm hai loại, dù không phải tất cả các biểu diễn đều có thể được phân chia một cách rõ ràng thuộc loại nào trong hai loại này.

- Phương pháp biểu diễn bề mặt B-reps mô tả các đối tượng ba chiều bằng một tập hợp các bề mặt giới hạn phần bên trong của đối tượng với môi trường bên ngoài. Ví dụ kinh điển của B-reps là việc biểu diễn các mặt đa giác và các mảnh tròn xoay.

- Phương pháp biểu diễn theo phân hoạch không gian (space-partitioning representation) thường được dùng để mô tả các thuộc tính bên trong của đối tượng bằng cách phân hoạch phần bên trong của đối tượng thành một tập hợp nhiều đối tượng nhỏ hơn.

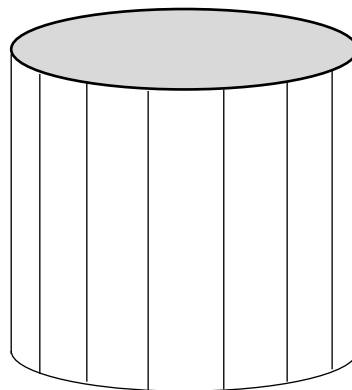
Trong đồ họa máy tính, các đối tượng lập thể có thể được mô tả bằng các bề mặt (surfaces) của chúng. Ví dụ : Một hình lập phương được xây dựng từ sáu mặt phẳng, một hình trụ được xây dựng từ sự kết hợp của một mặt cong và hai mặt phẳng, và hình cầu được xây dựng chỉ từ một mặt cong.

Thông thường để biểu diễn một đối tượng bất kì, người ta dùng các phương pháp xấp xỉ để đưa các mặt về dạng các mặt đa giác (polygon faces). Tuy nhiên trong trường hợp các đối tượng phức tạp, người ta thường dùng một hay nhiều mặt cong trơn (smoothly curved surfaces) ghép nối lại với nhau. Mỗi thành phần dùng để ghép nối được gọi là patch.

3.2.1. Biểu diễn mặt đa giác

Phương pháp chung nhất thường dùng để biểu diễn các đối tượng ba chiều là dùng một tập hợp các mặt đa giác xác định bề mặt của đối tượng. Rất nhiều hệ thống đồ họa lưu trữ các đối tượng như là một tập hợp các mặt đa giác. Với cách biểu diễn này ta có thể đơn giản hóa việc biểu diễn và tăng tốc độ hiển thị các đối tượng bởi vì tất cả các bề mặt đều được mô tả bởi các phương trình tuyến tính. Do đó, mô tả các đối tượng thông qua các mặt đa giác thường được dùng cho các đối tượng đồ họa cơ sở.

Trong một số trường hợp, ta chỉ có một khả năng chọn lựa là sử dụng biểu diễn đa giác. Tuy nhiên, một số hệ thống đồ họa còn cho phép các khả năng biểu diễn khác ví dụ như bằng các mặt cong spline.



Hình 3.1: Mô hình wireframe của một hình trụ

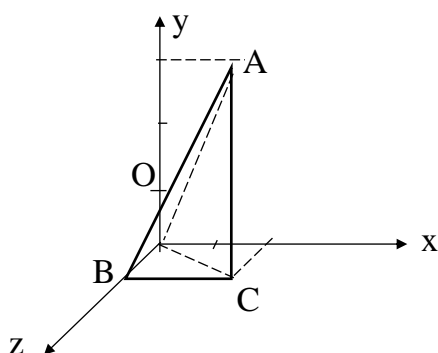
Biểu diễn bằng mặt đa giác của các đa diện cho chúng ta một định nghĩa chính xác về các đặc tính của các đối tượng này. Nhưng đối với những đối tượng khác ta chỉ nhận được một biểu diễn gần đúng. Hình 20 cho chúng ta biểu diễn một hình trụ như là một tập hợp các mặt đa giác. Biểu diễn dạng wireframe cho phép chúng ta hiển thị đối tượng rất nhanh. Khi cần thể hiện đối tượng thực hơn, ta có thể dùng kỹ thuật tạo bóng nội suy (interpolating shading).

a, Mô hình khung nối kết (Wireframe Model)

Mô hình khung nối kết là phương pháp thông dụng và đơn giản để mô hình hóa đối tượng. Mô hình này sử dụng một danh sách các đỉnh và một danh sách các cạnh nối các đỉnh đó. Danh sách các đỉnh cho biết vị trí các đỉnh, danh sách các cạnh xác định thông tin về sự kết nối, cho biết cặp các đỉnh tạo ra cạnh.

Các đối tượng 3 chiều khi được thể hiện bằng mô hình này là các đối tượng rỗng. Để hoàn thiện hơn, người ta dùng các kỹ thuật tạo bóng, các kỹ thuật loại bỏ các đường và mặt khuất. Mô hình khung kết nối thường được sử dụng trong việc xem phác thảo (preview) các đối tượng.

Ví dụ: Cho hình tứ diện OABC có tọa độ các đỉnh $O(0,0,0)$, $A(2,3,1)$, $B(0,0,1)$, $C(2,0,1)$



Danh sách các đỉnh			
Đỉnh	x	y	z
O	0	0	0
A	2	3	1
B	0	0	1
C	2	0	1

Danh sách các cạnh		
Cạnh	Đỉnh	Đỉnh
OA	O	A
OB	O	B
OC	O	C
AB	A	B
BC	B	C
CA	C	A

Hình 3.2: Minh họa mô hình khung kết nối

b, Biểu diễn bằng bảng đa giác

Ta biểu diễn một mặt đa giác bằng một tập hợp các đỉnh và các thuộc tính kèm theo. Khi thông tin của mỗi mặt đa giác được nhập, dữ liệu điền vào trong các bảng sẽ được dùng cho các xử lý tiếp theo như hiển thị và biến đổi. Các bảng dữ liệu mô tả mặt đa giác có thể tổ chức thành hai nhóm: các bảng hình học và các bảng thuộc tính.

Các bảng lưu trữ dữ liệu hình học chứa tọa độ của các đỉnh và các tham số cho biết về định hướng trong không gian của mặt đa giác. Thông tin về thuộc tính của các đối tượng chứa các tham số mô tả độ trong suốt, tính phản xạ và các thuộc tính cho biết đặc điểm bề mặt của đối tượng.

Một cách tổ chức thuận tiện để lưu trữ các dữ liệu hình học là tạo ra ba danh sách: một bảng lưu đỉnh, một bảng lưu cạnh và một bảng lưu đa giác. Các giá trị tọa độ cho mỗi đỉnh trong đối tượng được chứa trong bảng lưu đỉnh. Bảng cạnh chứa các con trỏ trỏ đến bảng đỉnh cho biết đỉnh nào được nối với một cạnh của đa giác. Và cuối cùng, bảng lưu đa giác chứa các con trỏ trỏ tới bảng lưu cạnh cho biết những cạnh nào tạo nên đa giác.

Ngoài ra, ta cũng có thể thêm một số thông tin bổ sung vào các bảng trên để xử lý nhanh hơn khi cần truy xuất thông tin.

Ví dụ, ta có thể thêm một con trỏ từ một cạnh đến các đa giác chứa nó. Tương tự, ta có thể thêm thông tin trong bảng lưu đỉnh để biết những cạnh nào kề với một đỉnh cho trước ... Vì các bảng lưu thông tin về đối tượng có thể rất phức tạp nên việc kiểm tra tính đúng đắn và đầy đủ của dữ liệu là rất quan trọng.

c, Phương trình mặt phẳng

Để thực hiện việc hiển thị một đối tượng ba chiều, ta phải xử lý dữ liệu nhập thông qua một quy trình gồm nhiều bước. Trong một số bước này, đôi khi ta cần thông tin về định hướng của đối tượng và cả thông tin về định hướng của từng mặt của đối tượng trong không gian. Những thông tin này có thể lấy được thông qua tọa độ của các đỉnh và phương trình mô tả các mặt đa giác của đối tượng. Phương trình biểu diễn mặt phẳng có dạng:

$$Ax + By + Cz + D = 0$$

Để xác định phương trình mặt phẳng, ta chỉ cần biết ba điểm không thẳng hàng trên mặt phẳng này. Như vậy, để xác định phương trình mặt phẳng qua một đa giác, ta sẽ sử dụng tọa độ của ba đỉnh: (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , trong đa giác này.

$$\text{vậy ta có: } \begin{cases} Ax_1 + By_1 + Cz_1 + D = 0 \\ Ax_2 + By_2 + Cz_2 + D = 0 \\ Ax_3 + By_3 + Cz_3 + D = 0 \end{cases}$$

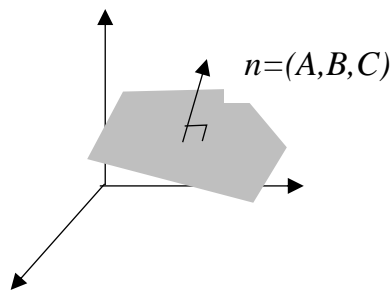
Dùng quy tắc Cramer, ta có thể xác định A , B , C , D theo công thức:

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \quad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}$$

$$C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

Khai triển các định thức trên ta được công thức tường minh của các hệ số:

$$\begin{cases} A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \\ B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \\ C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \\ D = x_1(y_3z_2 - y_2z_3) + x_2(y_1z_3 - z_1y_3) + x_3(y_2z_1 - y_1z_2) \end{cases}$$



Hình 3.3: Vector pháp tuyến của mặt phẳng

Hướng của mặt phẳng được xác định thông qua vector pháp tuyến của nó. Vector pháp tuyến $n = (A, B, C)$, trong đó A, B, C là các hệ số của phương trình mặt phẳng ta vừa tính được.

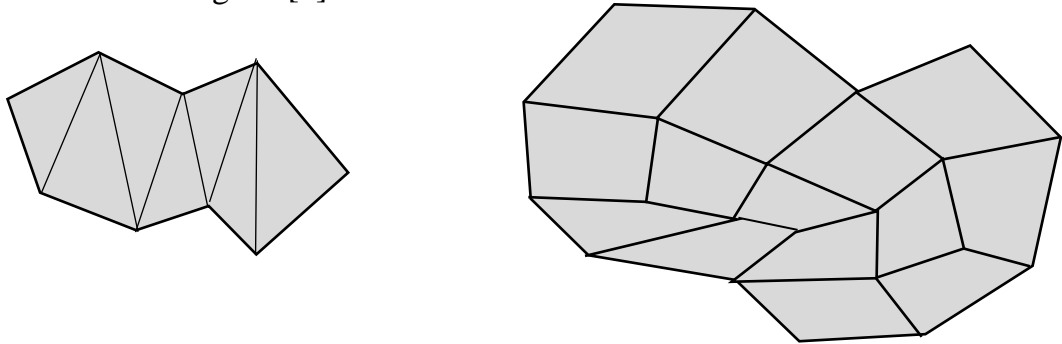
3.2.2. Lưới đa giác (polygon meshes)

Một số hệ đồ họa cung cấp một số hàm cho phép mô hình hóa các đối tượng. Việc biểu diễn một mặt phẳng là khá đơn giản, nhưng khi đối tượng có cấu tạo bề mặt từ nhiều mặt cong ta cần lợp nhiều patch liên tiếp, điều này được thực hiện nhờ việc dùng các hàm lưới (mesh function).

Một dạng thông dụng của lưới đa giác là dãy các tam giác (triangle strip). Hàm này vẽ $n-2$ tam giác kề nhau khi biết n đỉnh. Dạng này của lưới đa giác dùng trong hầu hết các thư viện đồ họa chuẩn hiện nay như OpenGL hay DirectX.

Một dạng hàm tương tự là lưới các tứ giác (quadrilateral mesh). Hàm này vẽ một lưới $(n-1)*(m-1)$ tứ giác lồi từ dãy $n*m$ đỉnh.

Khi đa giác được mô tả bởi nhiều hơn ba đỉnh, các đỉnh của nó có thể không đồng phẳng. Điều này có thể dẫn đến các lỗi tính toán. Một phương pháp đơn giản là phân đa giác này thành các tam giác. [6]



Hình 3.4: Triangle strip và quadrilateral mesh

3.2.3. Các đường cong và mặt cong

Hình ảnh của các đường cong và mặt cong có thể được tạo ra từ một tập hợp các hàm toán học hoặc từ một tập hợp các điểm trên đối tượng. Đối với đối tượng có thể được mô tả bằng các hàm toán học, thư viện đồ họa cung cấp sẵn các hàm cho phép biểu diễn các đối tượng này lên mặt phẳng thiết bị hiển thị.

Đối với các đường cong, các hàm trong thư viện sẽ vẽ các điểm theo hình chiếu của chúng. Đối với các mặt cong, một lưới đa giác xấp xỉ với mặt cong sẽ được tạo ra. Các hệ đồ họa tạo ra các lưới tam giác để đảm bảo tính đồng phẳng của các cạnh thuộc cùng một polygon patch. Một đường cong hoặc mặt cong có thể được diễn tả bằng phương trình toán học dạng tham số hoặc không tham số. Tuy nhiên, trong đồ họa máy tính, dạng tham số sẽ thuận tiện cho xử lý hơn.

Khi đối tượng được mô tả bởi một tập hợp các điểm rời rạc, đối tượng sẽ được hiển thị thông qua một mặt cong xấp xỉ nào đó dựa trên những điểm đã cho. Các loại đường cong và mặt cong dạng spline hoặc Bezier là những đường cong và mặt cong xấp xỉ thường dùng. Các mặt cong có thể có hình dạng rất phức tạp, đặc biệt khi nó bao gồm nhiều patch kết hợp lại với nhau.

3.3. Các phép biến đổi hình học ba chiều

Các phép biến đổi cơ sở trong đồ họa ba chiều là sự mở rộng của các phép biến đổi trong đồ họa hai chiều bằng cách thêm vào việc xem xét tọa độ thứ ba, tọa độ z. Bây giờ, chúng ta sẽ tịnh tiến một đối tượng thông qua việc mô tả một vector tịnh tiến ba chiều. Vector này xác định độ dời của vật theo ba chiều trong không gian. Tương tự như vậy, ta có thể thu phóng đối tượng với các tỉ lệ biến đổi theo cả ba chiều. Sự mở rộng của phép quay ít hiển nhiên hơn hai phép biến đổi cơ sở trên.

Khi khảo sát các phép quay trong mặt phẳng hai chiều Oxy, ta chỉ cần khảo sát phép quay quanh một tâm, hay nói cách khác, phép quay quanh một trục vuông góc với mặt phẳng Oxy. Trong không gian ba chiều, ta có thể chọn một trục quay có phương bất kì.

Phần lớn các hệ đồ họa xử lý phép quay trong không gian ba chiều như là tổ hợp của ba phép quay với trục quay là các trục tọa độ x, y và z. Như vậy, người dùng có thể dễ dàng xây dựng một phép quay bất kì bằng cách mô tả trục quay và góc quay.

Cũng như khi trình bày các phép biến đổi trong đồ họa hai chiều, trong chương này, ta sẽ khảo sát các phép biến đổi trong đồ họa ba chiều dưới dạng ma trận. Một chuỗi bất kì các phép biến đổi sẽ được biểu diễn bằng một ma trận duy nhất là tích của các ma trận tương ứng với các phép biến đổi thành phần.

3.3.1 Các phép biến đổi cơ bản

Phép tịnh tiến, quay, biến đổi tỉ lệ, và phép biến dạng là các ví dụ của các phép biến đổi hình học. Chúng còn được biết tới như là các phép biến đổi affine cơ sở. Trong số đó, phép quay có thể nói là quan trọng và hữu dụng nhất vì nó cho phép chúng ta nhìn các đối tượng theo các hướng khác nhau, điều này cho phép chúng ta cảm nhận các hình vẽ ba chiều trực quan hơn, dễ chịu hơn.

Phép biến đổi affine là phép biến đổi tuyến tính, khả nghịch. Phép biến đổi này bảo toàn tính song song của các đường thẳng cũng như bảo toàn tỉ lệ về khoảng cách của các đoạn thẳng. Tuy nhiên, phép biến đổi này không bảo toàn góc nghiêng và chiều dài các đoạn thẳng. Các phép biến đổi này cũng bảo toàn tỉ lệ về khoảng cách.

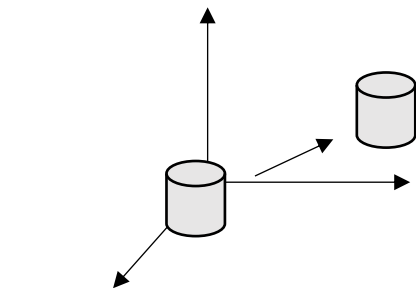
a, Phép tịnh tiến:

Phép tịnh tiến biến điểm $P(x,y,z)$ trong không gian 3D thành một điểm $Q(x',y',z')$ theo vector tịnh tiến $d(dx, dy, dz)$. Trong đó:

$$\begin{cases} x' = x + dx \\ y' = y + dy \\ z' = z + dz \end{cases}$$

Ma trận M cho phép tịnh tiến có dạng như sau:

$$M_T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{pmatrix}$$



Hình 3.5: Minh họa phép tịnh tiến

Ví dụ: Cho điểm $P(4,26,12)$ tịnh tiến theo vector $d(3,-3,-11)$ được điểm $Q(x',y',z')$.

Ta có: $Q = P \cdot M_T$

$$(x' \quad y' \quad z' \quad 1) = (4 \quad 26 \quad 12 \quad 1) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3 & -3 & -11 & 1 \end{pmatrix}$$

Vậy ảnh của P sau phép tịnh tiến là $Q(7,23,1)$

b, Phép biến đổi tỷ lệ tâm $O(0,0,0)$

Phép biến đổi tỷ lệ thực hiện co giãn một đối tượng theo 3 hướng x, y, z theo các tỷ lệ lần lượt là S_x, S_y, S_z .

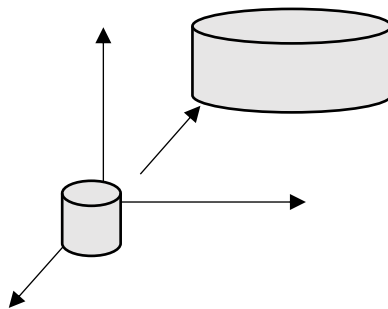
Gọi $Q(x',y',z')$ là ảnh của $P(x,y,z)$ sau khi thực hiện biến đổi tỷ lệ, ta có:

$$\begin{cases} x' = x.S_x \\ y' = y.S_y \\ z' = z.S_z \end{cases}$$

Trong đó:

- Nếu $S_x=S_y=S_z$ ta có phép biến đổi đồng dạng
- $S_x, S_y, S_z > 1$ ta có phép phóng to.
- $S_x, S_y, S_z < 1$ Ta có phép thu nhỏ

Ma trận của phép biến đổi tỷ lệ: $M_s = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$



Hình 3.6: Minh họa phép biến đổi tỷ lệ

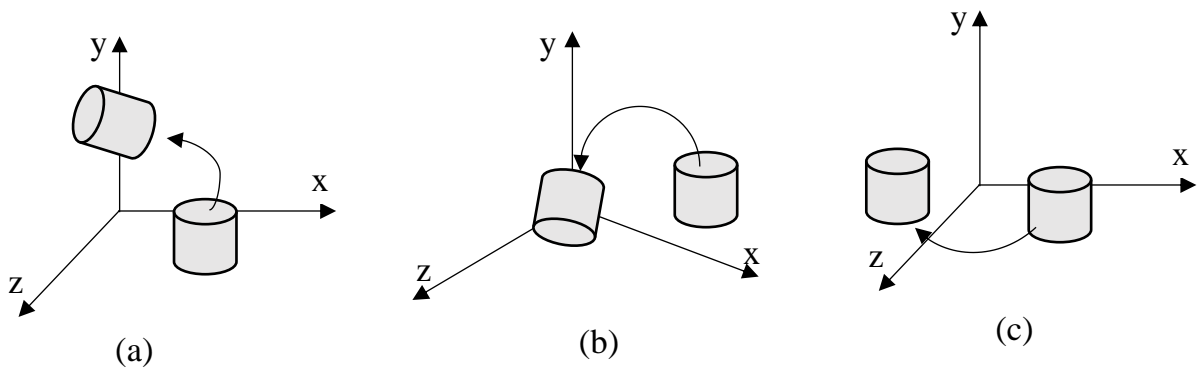
c, Phép quay tâm $O(0,0,0)$

Khác với phép quay trong mặt phẳng quanh một điểm bất kì, trong không gian ba chiều ta có phép quay quanh một trục tọa độ.

Ở đây, sử dụng hệ tọa độ theo quy ước bàn tay phải và quy định chiều quay dương khi nhìn từ ngoài vào gốc tọa độ là ngược chiều kim đồng hồ:

- Quay quanh trục x : từ trục dương y đến trục dương z
- Quay quanh trục y : từ trục dương z đến trục dương x

- Quay quanh trục z : từ trục dương x đến trục dương y



Hình 3.7: Phép quay, (a) quay quanh trục z, (b) quanh trục x, (c) quanh trục y

Ta có các ma trận biểu diễn các phép quay quanh trục x, y, z một góc α lần lượt là $M(z, \alpha)$, $M(x, \alpha)$, $M(y, \alpha)$:

Quay quanh trục Oz :

$$M(z, \alpha) = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Quay quanh trục Ox:

$$M(x, \alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Quay quanh trục Oy:

$$M(y, \alpha) = \begin{pmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ví dụ: Tìm ảnh $Q(x', y', z')$ của điểm $P(21, 15, 49)$ qua phép quay quanh Ox một góc 90° .

Ta có: $Q=P.M(x,\alpha)$

$$(x' \ y' \ z' \ 1) = (21 \ 15 \ 49 \ 1) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Vậy ảnh của P sau phép tịnh tiến là $Q(21,-49,15)$

3.3.2 Các phép đối xứng

a, Phép đối xứng qua mặt phẳng tọa độ:

Ta có ma trận M của các phép đối xứng qua các mặt phẳng tọa độ xOy, xOz, yOz lần lượt là:

$$M(xOy) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M(xOz) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M(yOz) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b, Phép đối xứng qua trục tọa độ:

Ta có ma trận M của các phép đối xứng qua các trục tọa độ Ox, Oy, Oz lần lượt là:

$$M(Ox) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M(Oy) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M(Oz) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

c, Phép đối xứng qua gốc tọa độ:

Phép lấy đối xứng qua gốc tọa độ tương đương với phép kết hợp của phép lấy đối xứng qua mặt phẳng xOy và phép lấy đối xứng qua trục Oz:

$$M(O) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3.3.3 Phép biến đổi tỷ lệ tâm bất kỳ

Phép biến đổi tỷ lệ tâm bất kỳ $I(x_i, y_i, z_i)$ được xem là sự kết hợp từ các phép biến đổi cơ sở sau:

- Tịnh tiến theo vector $(-x_i, -y_i, -z_i)$ để dịch chuyển tâm biến đổi tỷ lệ về gốc tọa độ, ta có ma trận của phép tịnh tiến là $M_{T(-I)}$
- Thực hiện biến đổi các đối tượng theo các tỷ lệ (S_x, S_y, S_z) , ma trận M_s
- Tịnh tiến theo vector (x_i, y_i, z_i) để đưa tâm biến đổi tỷ lệ về lại vị trí ban đầu, ma trận $M_{T(I)}$

$$M = M_{T(-I)} M_s M_{T(I)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_i & -y_i & -z_i & 1 \end{pmatrix} \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_i & -y_i & -z_i & 1 \end{pmatrix}$$

$$\Rightarrow M = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ (1-S_x)x_i & (1-S_y)y_i & (1-S_z)z_i & 1 \end{pmatrix}$$

3.4. Các phép chiếu

Hình ảnh của các đối tượng trong thế giới thực là hình ảnh 3 chiều. Để có thể biểu diễn chúng trên màn hình các thiết bị hiển thị (2 chiều) một cách sinh động người ta cần thực hiện các phép chiếu, các phép tô chất (render) để gây ảo giác (illusion) về độ sâu.

Đồ họa 3D là các kỹ thuật biểu diễn thế giới tự nhiên trên các thiết bị hiển thị 2 chiều.

Khái niệm về phép chiếu:

Một cách tổng quát, phép chiếu là phép chuyển đổi những điểm của đối tượng trong hệ thống tọa độ n chiều thành những điểm trong hệ thống tọa độ có số chiều nhỏ hơn n .

Thông thường, ta chiếu các đối tượng đồ họa 3 chiều, miêu tả thế giới thực, xuống mặt phẳng 2 chiều của thiết bị hiển thị.

Khái niệm về hình chiếu:

Hình chiếu là ảnh của các đối tượng được hình thành từ phép chiếu, là giao của các tia chiếu (projector) với mặt phẳng chiếu.

Tia chiếu là tia xuất phát từ một điểm gọi là tâm chiếu (center of projection) đi qua các điểm của đối tượng. Mặt phẳng chiếu (projection plan) là mặt phẳng thiết bị hiển thị.

3.4.1 Phép chiếu song song

Phép chiếu song song (Parallel Projections) là phép chiếu mà ở đó các tia chiếu song song với nhau hay các tia chiếu xuất phát từ một điểm ở vô cùng.

Phân loại phép chiếu song song dựa trên hướng của tia chiếu (*Direction Of Projection*) tới mặt phẳng chiếu (*projection plane*)

a, Phép chiếu trực giao

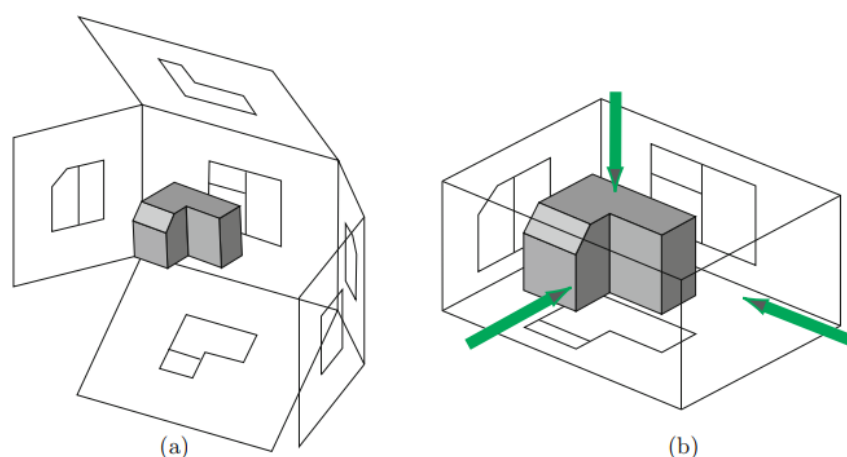
Là phép chiếu song song và tia chiếu vuông góc với mặt phẳng chiếu.

Về mặt toán học, phép chiếu trực giao là phép chiếu mà mặt phẳng chiếu là một trong các mặt phẳng toạ độ, thường dùng mặt phẳng $z=0$ (mặt phẳng xOy), ngoài ra có các mặt phẳng $x=0$ và $y=0$. Ứng với mỗi mặt phẳng chiếu ta có một ma trận chiếu tương ứng.

$$T_x = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T_y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Với các đối tượng hình hộp, người ta không sử dụng cả 6 mặt phẳng để suy diễn ngược hình của đối tượng (*Hình 26a,*) mà chỉ sử dụng một trong số chúng như: hình chiếu bằng, đứng, cạnh (*Hình 26b,*). Cả sáu góc nhìn đều có thể thu được từ một mặt phẳng chiếu thông qua các phép biến đổi hình học như quay, dịch chuyển hay lấy đối xứng.

Giả sử chúng ta có hình chiếu bằng trên mặt phẳng $z=0$, với phép quay đối tượng quanh trục một góc 90° sẽ cho ta hình chiếu cạnh. Đối với các đối tượng mà các mặt của chúng không song song với một trong các mặt phẳng hệ toạ độ thì phép chiếu này không



Hình 3.8: Phép chiếu trực giao, (a) phép chiếu 6 mặt, (b) phép chiếu 3 mặt

cho hình dạng thật của vật thể. Muốn nhìn vật thể chính xác hơn người ta phải hình thành

phép chiếu thông qua việc quay và dịch chuyển đối tượng sao cho mặt phẳng đó song song với các trục toạ độ.

Hình của đối tượng quá phức tạp cần thiết phải biết các phần bên trong của đối tượng đôi lúc chúng ta phải tạo mặt cắt đối tượng

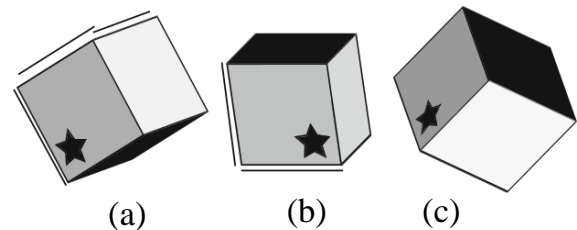
b, Phép chiếu trục lượng (Axonometric)

Phép chiếu trục lượng là phép chiếu mà hình chiếu thu được sau khi quay đối tượng sao cho ba mặt của đối tượng được trông thấy rõ nhất (thường mặt phẳng chiếu là $z=0$).

Kích thước hình chiếu của các đối tượng nằm trên mặt phẳng song song với mặt phẳng chiếu bằng với kích thước của đối tượng. Hình chiếu của các đối tượng nằm trên các mặt còn lại thường biến dạng sau phép chiếu theo tỷ lệ co SF (Shortening Factor).

SF là tỷ số của kích thước hình chiếu với kích thước đối tượng thực. Trên cơ sở SF phép chiếu trục lượng được chia làm ba loại sau:

- Phép chiếu Trimetric
- Phép chiếu Dimetric
- Phép chiếu Isometric



Hình 3.9: Minh họa phép chiếu trục lượng

- **Phép chiếu Trimetric**

Là phép chiếu hình thành từ việc quay tự do đối tượng trên một trục hay trên cả 3 trục của hệ toạ độ và chiếu đối tượng đó bằng phép chiếu song song lên mặt phẳng chiếu (thường là mặt phẳng $z=0$).

Hệ số co SF của phép chiếu Trimetric theo các hướng x, y, z là khác nhau.

- **Phép chiếu Dimetric**

Phép chiếu Dimetric là phép chiếu Trimetric có 2 hệ số co bằng nhau, giá trị thứ 3 còn lại là tùy ý. Phép chiếu được xây dựng bằng cách quay đối tượng quanh trục y theo một góc ϕ , tiếp đó quanh trục ox một góc ϕ và sau cùng là phép chiếu song song lên mặt phẳng $z=0$.

- **Phép chiếu Isometric**

Phép chiếu Isometric là phép chiếu Trimetric mà ở đó hệ số co SF trên 3 trục là bằng nhau.

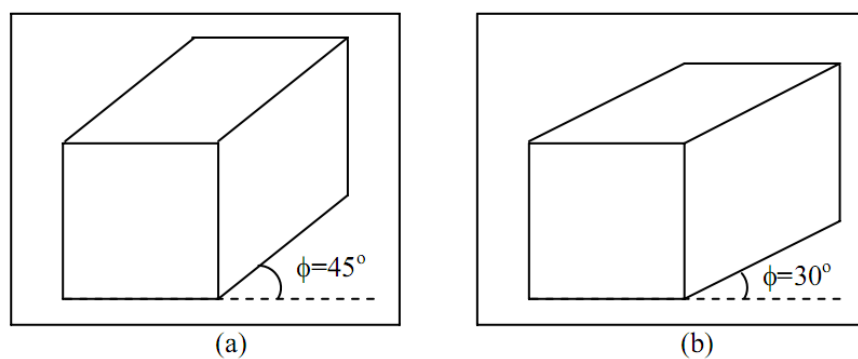
c, Phép chiếu xiên

Là phép chiếu song song, nhưng tia chiếu không vuông góc với mặt phẳng chiếu mà nó hợp với mặt phẳng chiếu một góc ϕ .

- **Phép chiếu Cavalier**

Phép chiếu Cavalier là phép chiếu xiên được tạo thành khi các tia chiếu làm thành với mặt phẳng chiếu một góc $\phi=45^\circ$ hoặc $\phi=30^\circ$.

Hệ số co trên các hệ trục toạ độ bằng nhau.

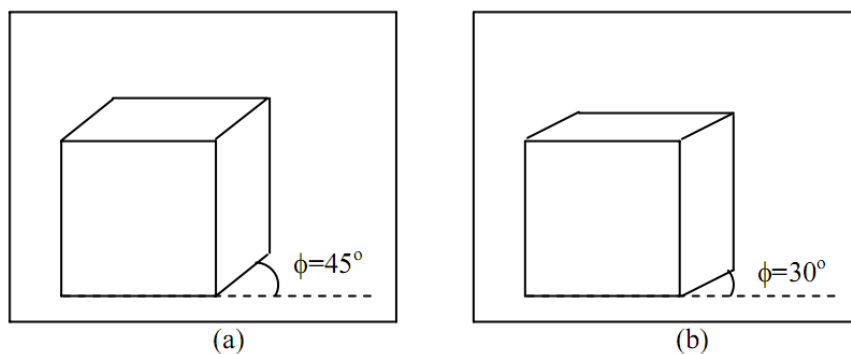


Hình 19: Phép chiếu Cavalier, (a) góc $\phi=45^0$, (b) $\phi=30^0$

- Phép chiếu Cabinet

Phép chiếu xiên với hệ số co tỉ lệ $f = 1/2$.

Với phép chiếu Cabinet, giá trị của ϕ có thể thay đổi tùy ý. Các giá trị thường được sử dụng là $\phi=30^0$ và $\phi=45^0$.

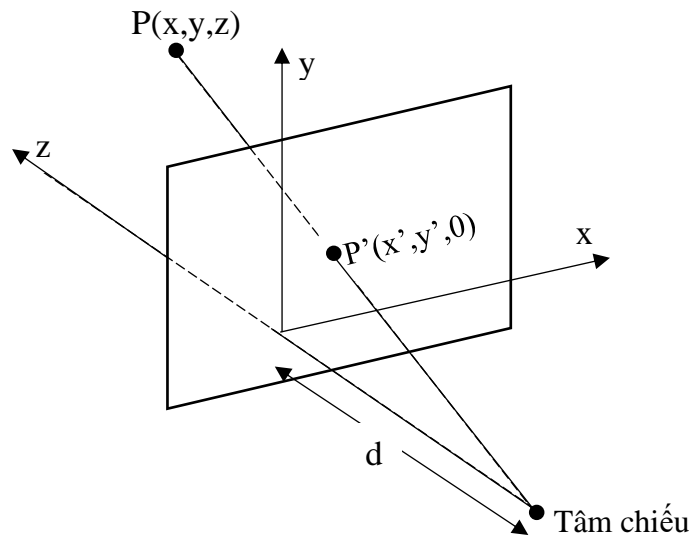


Hình 20: Phép chiếu Cabinet, (a) góc $\phi=45^0$, (b) $\phi=30^0$

3.4.2 Phép chiếu phối cảnh

Phép chiếu phối cảnh là phép chiếu mà các tia chiếu không song song với nhau mà xuất phát từ một điểm gọi là tâm chiếu.

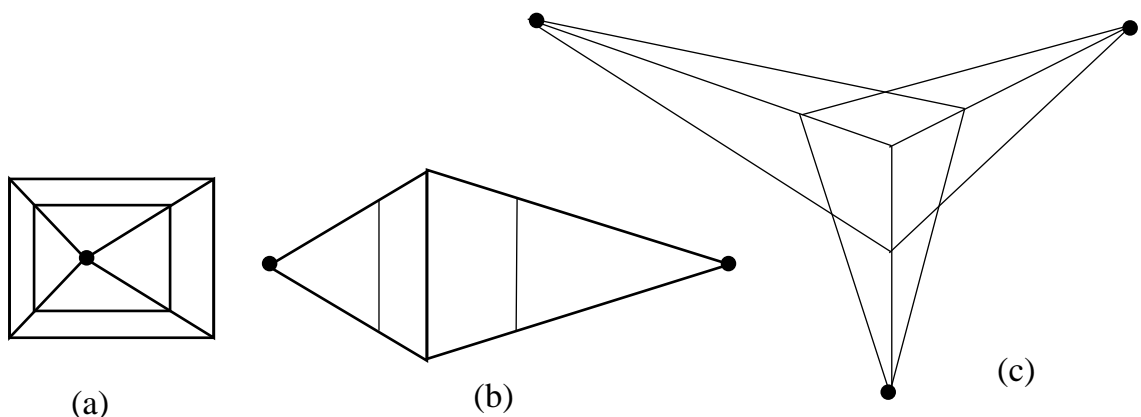
Phép chiếu phối cảnh tạo ra hiệu ứng về luật xa gần tạo cảm giác về độ sâu của đối tượng trong thế giới thật mà phép chiếu song song không lột tả được.



Hình 212: Minh họa phép chiếu phối cảnh

Để đạt được phép chiếu phối cảnh của đối tượng ba chiều, chúng ta chiếu các điểm theo đường thẳng chiếu để các đường này gặp nhau ở tâm chiếu. Trong hình 29, tâm chiếu trên trục Oz và có giá trị âm, cách một khoảng d phía sau mặt phẳng chiếu. Bất kỳ điểm nào cũng có thể được chọn làm tâm của phép chiếu, tuy nhiên việc chọn một điểm dọc theo trục Oz sẽ làm đơn giản việc tính toán trong các phương trình biến đổi.

Trong trường hợp đối tượng là các đường thẳng song song và cũng song song với mặt phẳng chiếu thì ảnh chiếu nhận được là các đường thẳng song song. Ngược lại, khi đối tượng là các đường thẳng song song nhưng không song song với mặt phẳng chiếu thì ảnh chiếu nhận được sẽ là các đường hội tụ tại một điểm, điểm đó được gọi là tâm chiếu.



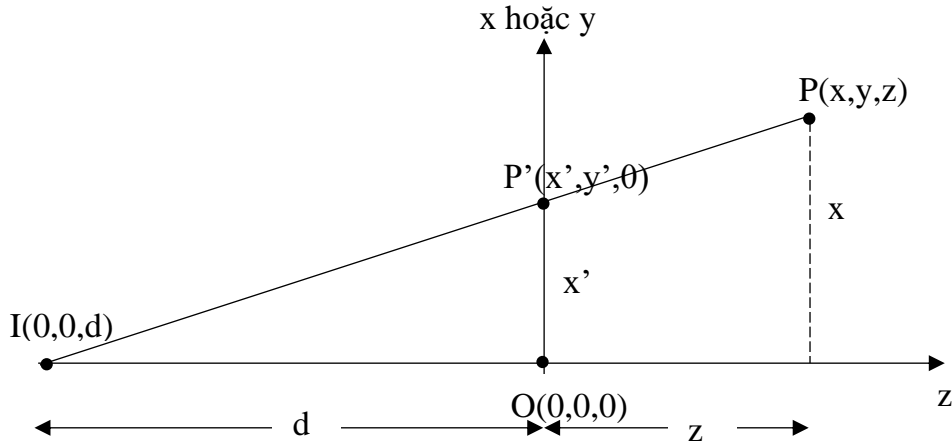
Hình 22: (a) Một tâm chiếu, (b) hai tâm chiếu, (c) ba tâm chiếu

Mỗi tập hợp các đường thẳng song song được chiếu như thế sẽ có một tâm chiếu riêng.

Số lượng các tâm chiếu trong một phép chiếu được xác định bởi số lượng các trục của hệ tọa độ thực cắt mặt phẳng chiếu. Hình trên minh họa hình ảnh của các phép chiếu phối cảnh một tâm chiếu (a), hai tâm chiếu (b) và ba tâm chiếu (c) của một hình lập phương.[7]

- **Phép chiếu phối cảnh một tâm chiếu:**

Cho điểm $P(x,y,z)$, tìm ảnh $P'(x',y',z')$ của P trên mặt phẳng chiếu $z=0$ với tâm chiếu $I(0,0,d)$ nằm trên trục Oz .



Hình 23: Phép chiếu phối cảnh tâm chiếu $I(0,0,d)$

Từ hình 31, ta có:
$$\begin{cases} x'/d = x/(z+d) \\ y'/d = y/(z+d) \end{cases} \Rightarrow \begin{cases} x' = x/(z/d+1) \\ y' = y/(z/d+1) \end{cases}$$

$P'(x', y', 0)$ được viết dưới dạng tọa độ thuần nhất là $P'(x', y', 0, h)$ trong đó $h=1$.

Khi đặt $h'=z/d+1$, ta có tọa độ thuần nhất của P' là $P'(x,y,0,h')$.

Từ đó ta có thể đưa ra ma trận của phép chiếu có dạng:

$$M_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tương tự như vậy, để tìm ảnh $P'(x',y',z')$ của $P(x,y,z)$ trên mặt phẳng chiếu $x=0$ ($y=0$) với tâm chiếu $I(k,0,0)$ ($I(0,r,0)$) nằm trên trục Ox (Oy), ta có:

$$M_x = \begin{pmatrix} 1 & 0 & 0 & 1/k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1/r \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Câu hỏi ôn tập chương:

- 1, Trình bày phương pháp biểu diễn mặt đa giác và lưới đa giác.
- 2, Trình bày các phép biến đổi 3 chiều: Phép tịnh tiến, phép biến đổi tỷ lệ, phép quay và các phép lấy đối xứng.
- 3, Thế nào là phép chiếu song song, có những loại chiếu song song nào?
- 4, Thế nào là phép chiếu phối cảnh, có những loại chiếu phối cảnh nào?
- 5, Xây dựng ma trận của phép lấy đối xứng qua tâm bất kỳ.

Hướng dẫn: Phép lấy đối xứng qua tâm bất kỳ là tổng hợp từ các phép biến đổi sau:

- Phép tịnh tiến đưa tâm về gốc tọa độ
- Phép lấy đối xứng tâm $O(0,0,0)$
- Phép tịnh tiến đưa tâm đối xứng về vị trí ban đầu.

- 6, Xây dựng ma trận của phép lấy đối xứng qua mặt phẳng $y = x$.

Hướng dẫn: Phép lấy đối xứng qua mặt phẳng $y = x$ là tổng hợp từ các phép biến đổi sau: - Phép quay 45° quanh trục Oz để đưa mặt phẳng $y=x$ về mặt phẳng yOz

- Phép lấy đối xứng qua mặt phẳng yOz
- Phép quay -45° quanh trục Oz để đưa về vị trí ban đầu của mặt phẳng $y=x$.

Bài tập ôn tập:

- 1, Tìm ảnh của điểm $P(5,7,1)$ qua phép tịnh tiến có độ dời theo trục Ox là $dx=-3$, theo trục Oy là $dy=10$, theo trục Oz là $dz=7$.
- 2, Tìm ảnh của điểm $P(4,5,9)$ qua phép quay quanh trục Ox một góc $\alpha = -90^\circ$.
- 3, Tìm ảnh của điểm $P(2,5,11)$ qua phép quay quanh trục Oy một góc $\alpha = 60^\circ$.
- 4, Tìm ảnh của điểm $P(6,17,14)$ qua phép biến đổi tỷ lệ tâm $O(0,0,0)$ với tỷ lệ biến đổi $S_x=1.5$, $S_y=0.5$, $S_z=2$.
- 5, Tìm ảnh của điểm $P(2,12,6)$ qua phép biến đổi tỷ lệ tâm $I(3,4,9)$ với tỷ lệ biến đổi $S_x=2.5$, $S_y=2$, $S_z=1.5$.
- 6, Tìm ảnh của đoạn thẳng $P_1(2,4,1)P_2(7,10,22)$ sau khi thực hiện phóng to gấp 2 lần, cho biết ảnh của P_1 vẫn là chính nó.
- 7, Tìm ảnh của điểm $P(9,1,23)$ qua phép lấy đối xứng tâm $I(3,4,17)$.
- 8, Tìm ảnh của điểm $P(24,6,19)$ qua phép lấy đối xứng mặt phẳng xOy, trục Ox, tâm $O(0,0,0)$
- 9, Tìm ảnh của đoạn $P_1(4,16,23)P_2(5,17,2)$ sau khi thực hiện chiếu phối cảnh một tâm chiếu tại:

- a, $I_x(5,0,0)$
- b, $I_y(0,12,0)$
- c, $I_z(0,0,4)$

10, Cho tam giác ABC có tọa độ: A(1,3,11), B(5,15,23), C(8,18,31). Tìm hình chiếu trực giao của tam giác ABC trên mặt phẳng chiếu xOy.

11, Chứng minh rằng trong các phép biến đổi, tâm biến đổi là điểm bất động (có ảnh là chính nó).

12. Anh (chị) hãy tron phát biểu đúng nhất:

Trong phép biến đổi tỷ lệ, S_x, S_y được gọi là các hệ số tỷ lệ:

a, Nếu $S_x=S_y=S_z$ ta có phép biến đổi tỷ lệ đồng dạng

b, Nếu $S_x > 1, S_y > 1, S_z > 1$ ta có phép phóng to, đồng thời ảnh sẽ rời xa gốc tọa độ

c, Nếu $S_x < 1, S_y < 1, S_z < 1$ ta có phép thu nhỏ, đồng thời ảnh sẽ về gần gốc tọa độ

d, Cả 3 ý trên đều đúng

13. Trong 3D có điểm P(13,25,1), đối xứng qua trục oy được Q là:

a. (13,-25,-1)

b. (-13,25,-1)

c. (-25,13,1)

d. (25,-13,1)

14. Trong 3D có điểm P(6,10,5), đối xứng qua gốc tọa độ O(0,0,0) được Q là:

a. (-6,10,5)

b. (6,-10,5)

c. (-6,-10,-5)

d. (-10,-6,-5)

15. Sau khi thực hiện tịnh tiến P(10,8,1) theo véc tơ độ dời (-6,3,9) rồi biến đổi theo tỷ lệ $S_x=0.5, S_y=2, S_z=1$ ta được Q là:

a. (2,22,10)

b. (22,2,10)

c. (-1,19,5)

d. (19,-1,5)

16. Trong không gian, lấy đối xứng điểm A(6,9,8) qua trục Oz rồi quay 90 độ quanh Ox ta được điểm B(x,y,z) có tọa độ là:

a. B(6,9,8)

b. B(8,-9,6)

c. B(-6,-9,8)

d. B(-6,-8,-9)

17. Trong không gian, lấy đối xứng điểm A(2,5,3) qua trục Ox ta được điểm B(x,y,z) có tọa độ là:

a. B(2,-5,-3)

b. B(-2,5,-3)

c. B(-5,2,3)

d. B(5,-3,2)

CHƯƠNG 4: CÁC HỆ MÀU

Mục tiêu của chương: Sinh viên cần nắm được khái niệm về ánh sáng và màu sắc theo hướng tiếp cận của kỹ thuật đồ họa, hiểu được cơ chế cảm nhận các sắc màu khác nhau của mắt người.

Hiểu rõ về các chế độ màu RGBA và chỉ mục màu, biết xác định kích thước của bộ đệm dành cho bảng màu khi cho trước số bit để hiển thị màu cho 1 pixel. Hiểu về kỹ thuật phối màu để tạo ra một màu không có trong bảng màu.

Sinh viên cần nắm được các cơ chế phối màu và hiển thị màu trên các hệ màu RGB, CMY, CMY-K, HSV...

4.1 Ánh sáng và sự cảm nhận màu sắc của mắt

Về mặt vật lý, ánh sáng bao gồm các photon - các hạt ánh sáng nhỏ, mỗi hạt đi dọc theo đường đi riêng của nó và mỗi hạt lại dao động theo tần số riêng (hay bước sóng, năng lượng riêng). Các photon có bước sóng nằm trong khoảng từ 390 nanomet (nm) (tím) và 720 nm (đỏ) bao phủ các màu của quang phổ nhìn thấy được, tạo thành các màu của cầu vồng (tím, chàm, xanh dương, xanh lục, vàng, cam, đỏ). Tuy nhiên, mắt của chúng ta cảm nhận được rất nhiều màu sắc không có trong cầu vồng - ví dụ như trắng, đen, nâu, hồng... Làm thế nào điều này xảy ra?

Những gì mắt thực sự nhìn thấy là một hỗn hợp các photon có tần số khác nhau. Nguồn sáng thực được đặc trưng bởi sự phân bố tần số photon mà chúng phát ra. Ánh sáng trắng lý tưởng bao gồm một lượng ánh sáng bằng nhau của tất cả các tần số. Ánh sáng laser thường rất tinh khiết và tất cả các photon đều có tần số gần như giống hệt nhau (cũng như hướng và pha). Ánh sáng từ đèn hơi natri có nhiều ánh sáng hơn ở tần số vàng. Ánh sáng từ hầu hết các ngôi sao trong không gian có sự phân bố phụ thuộc nhiều vào nhiệt độ của chúng. Sự phân bố tần số ánh sáng từ hầu hết các nguồn trong môi trường phức tạp hơn.

Mắt người nhận biết màu sắc khi các tế bào trong võng mạc (tế bào hình nón) được kích thích sau khi tiếp nhận các photon ánh sáng. Ba loại tế bào hình nón khác nhau phản ứng tốt nhất với ba bước sóng ánh sáng khác nhau: một loại tế bào hình nón phản ứng tốt nhất với ánh sáng đỏ, một loại với màu xanh lá cây và loại kia là màu xanh lam. (Một người bị mù màu thường thiếu một hoặc nhiều loại tế bào hình nón.) Khi một chùm photon đi vào mắt, các tế bào hình nón trong võng mạc bị kích thích với mức độ nào đó. Mỗi mức độ kích thích tương đương với một màu sắc nhất định.

Mắt có thể cảm nhận được các màu không có trong quang phổ do lăng kính hoặc cầu vồng tạo ra. Ví dụ, nếu chiếu một chùm photon đỏ (red) và xanh (blue) để cả hai loại tế bào hình nón màu đỏ và màu xanh trong võng mạc bị kích thích, mắt ta sẽ thấy nó là màu đỏ tươi (magenta), không có trong quang phổ. Các kết hợp khác cho màu nâu, màu xanh lam và màu tím nhạt... một số màu không có trong quang phổ màu.

Một màn hình đồ họa máy tính mô phỏng các màu có thể nhìn thấy bằng cách kết hợp ánh sáng đỏ (red), lục (green) và lam (blue) theo các tỷ lệ khác nhau để gây ra các mức độ kích thích trên các tế bào hình nón sao cho phù hợp với các màu sắc cổ găng mô phỏng. Nếu con người có nhiều loại tế bào hình nón hơn, ví dụ như một số tế bào nhạy cảm với màu vàng, thì màn hình màu cũng có thể có cơ chế phát sinh ánh sáng vàng.

Để hiển thị một màu cụ thể, màn hình sẽ gửi đúng lượng ánh sáng đỏ, lục và lam theo một tỷ lệ cụ thể để kích thích thích hợp các loại tế bào hình nón khác nhau trong mắt. Một màn hình màu có thể phát sinh hàng triệu các tỷ lệ khác nhau của màu đỏ, xanh lục và xanh lam cho mỗi pixel và mắt nhìn thấy hàng triệu màu sắc khác nhau ấy.

4.2 Biểu diễn màu trên màn hình đồ họa

4.2.1 Chế độ hiển thị màu RGBA

Trên màn hình màu, mỗi pixel phát ra các lượng ánh sáng đỏ, lục và lam theo các tỷ lệ khác nhau. Mỗi màu được hiển thị tương ứng với một tỷ lệ nhất định mà ta gọi là bộ màu RGB. Đôi khi có giá trị thứ tư, được gọi là alpha hoặc A. Thông tin màu tại mỗi pixel có thể được lưu trong chế độ RGBA, trong đó các giá trị R, G, B và A được chỉ định cho mỗi pixel ở chế độ chỉ mục màu. Mỗi chỉ số màu biểu thị một mục trong bảng xác định một tập hợp các giá trị R, G và B cụ thể. Một bảng như vậy được gọi là bản đồ màu.

Mỗi hệ thống phần cứng đồ họa khác nhau sẽ khác nhau rất nhiều về kích thước của màn hình và số lượng màu sắc có thể được hiển thị ở mỗi pixel. Trên một hệ thống đồ họa nhất định, mọi pixel có cùng một dung lượng nhớ để lưu trữ màu của nó, và phần bộ nhớ của cả màn hình được gọi là bộ đệm màu. Kích thước của bộ đệm thường được đo bằng bit, do đó, bộ đệm 8 bit có thể lưu trữ 8 bit dữ liệu (256 màu khác nhau có thể) cho mỗi pixel. Kích thước của bộ đệm có thể thay đổi từ máy này sang máy khác.

Các giá trị R, G và B có thể nằm trong khoảng từ 0,0 đến 1,0. Ví dụ: R = 0,0, G = 0,0 và B = 1,0 đại diện cho màu xanh (blue) sáng nhất có thể. Nếu R, G và B của một pixel đều là 0,0, pixel đó có màu đen; nếu tất cả là 1,0, pixel được vẽ bằng màu trắng sáng nhất mà màn hình có thể hiển thị. Pha trộn màu xanh lá cây (green) và màu xanh (blue) tạo ra sắc thái của màu lục lam (cyan). Màu xanh (blue) và đỏ (red) kết hợp cho màu đỏ tươi (magenta). Màu đỏ và màu xanh lá cây tạo ra màu vàng...

Trong chế độ chỉ mục màu hoặc chế độ RGBA, một lượng dữ liệu màu nhất định được lưu trữ cho mỗi pixel. Độ lớn của lượng dữ liệu này được xác định bởi số lượng bit nhớ trong bộ đệm khung. Nếu có 8 bit màu cho mỗi pixel ta sẽ có $2^8 = 256$ giá trị hoặc màu khác nhau có thể được lưu trữ.

Bit nhớ thường được chia đều cho việc lưu trữ cho các thành phần R, G và B (nghĩa là hệ thống 24 bit sẽ dành 8 bit cho mỗi màu đỏ, xanh lục và xanh lam), nhưng điều này không phải lúc nào cũng đúng.

Các giá trị R, G và B thường được lưu trữ dưới dạng số nguyên thay vì số dấu phẩy động và chúng được chia tỷ lệ theo số bit để lưu trữ và truy xuất. Ví dụ: nếu một hệ thống

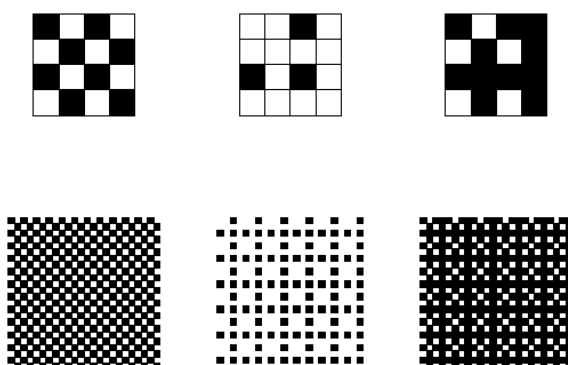
8 bit cho thành phần R, tương ứng ta có miền giá trị số nguyên từ 0 đến 255 được lưu trữ cho các mức màu; do đó, các giá trị 0, 1, 2, ..., 255 sẽ tương ứng với các giá trị R là $0/255 = 0.0$, $1/255$, $2/255$, ..., $255/255 = 1.0$ trong bảng chỉ mục. Bất kể số lượng bit nhớ là bao nhiêu, 0.0 chỉ định cường độ tối thiểu và 1.0 chỉ định cường độ tối đa.

Số lượng màu sắc riêng biệt có thể được hiển thị tại một pixel phụ thuộc vào số lượng bit nhớ và khả năng của phần cứng để diễn giải các bit nhớ đó. Số lượng màu riêng biệt không thể vượt quá 2^n , trong đó n là số lượng bit nhớ. Do đó, một cỗ máy có 24 bit nhớ cho RGB có thể hiển thị tới 16,77 triệu màu khác biệt.

Một số phần cứng đồ họa sử dụng phối màu để tăng số lượng màu hiển thị. Phối màu là kỹ thuật sử dụng kết hợp một số màu để tạo nên các hiệu ứng về màu sắc. Để minh họa cách phối màu hoạt động, giả sử một hệ thống chỉ có một bit cho mỗi màu red, green và blue, do đó, nó chỉ có thể hiển thị tám màu: đen, trắng, đỏ, xanh dương, xanh lục, vàng, lục lam và đỏ tươi. Để hiển thị một vùng màu hồng, phần cứng có thể lấp đầy vùng đó theo cách xen kẽ các pixel màu đỏ và trắng. Nếu vị trí của mắt đủ xa so với màn hình để không thể phân biệt các pixel riêng lẻ, vùng đó sẽ xuất hiện màu hồng - trung bình của màu đỏ và trắng. Có thể điều chỉnh mức độ của màu hồng bằng cách điều chỉnh tỷ lệ số các pixel có ánh sáng đỏ và trắng, v.v.

Với kỹ thuật này, không có pixel được hiển thị màu hồng. Cách để đạt được hiệu ứng "màu hồng" là sử dụng kỹ thuật phối màu. Như vậy, nếu chỉ định giá trị RGB cho màu không khả dụng để tô vào một đa giác, phần cứng sẽ lấp đầy các pixel ở bên trong đa giác bằng hỗn hợp các giá trị màu gần với màu đó. Các giá trị màu này cho màu trung bình mà mắt cảm nhận được tương ứng với màu mong muốn.

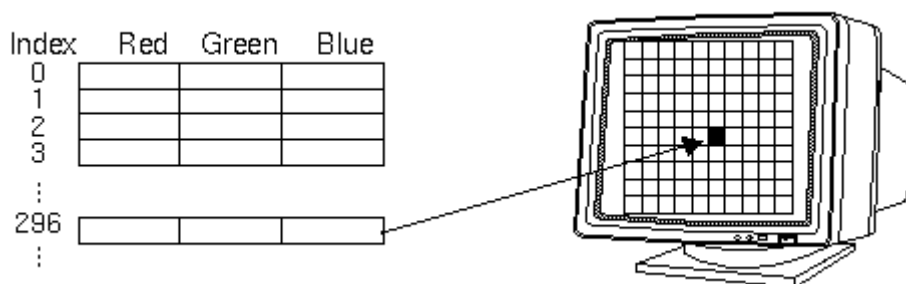
Hình 34 minh họa một số phối màu đơn giản của các pixel đen và trắng để tạo ra các sắc thái của màu xám. Từ trái sang phải, các mẫu 4×4 ở trên cùng đại diện cho các mẫu phối màu cho 50% số pixel màu đen, 19% số pixel màu đen và 69% số pixel màu đen. Trong mỗi mẫu, ta có thể thấy các bản sao giảm lặp đi lặp lại của từng mẫu. Nếu nhìn ở vị trí đủ xa 3 mẫu sẽ xuất hiện dưới dạng ba cấp độ màu xám khác nhau.



Hình 4.1: Minh họa phối màu

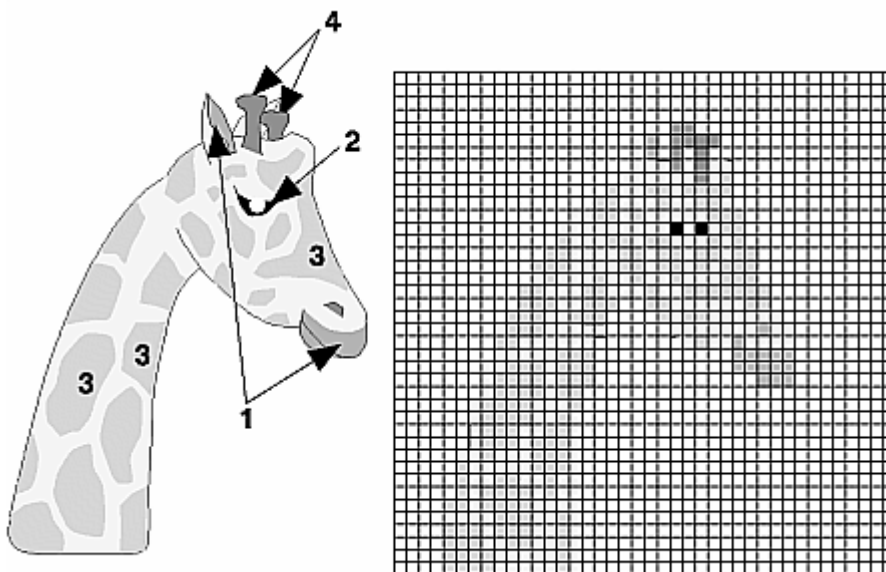
4.2.2. Chế độ hiển thị màu chỉ mục

Với chế độ chỉ mục màu, bản đồ màu của máy tính cung cấp các chỉ số cho biết các bộ giá trị của màu đỏ, xanh lục và xanh lam.



Hình 4.2: Ảnh xạ màu lên pixel

Một cảnh được chia ra thành các vùng tô màu được đánh số. Các vùng có cùng màu sẽ được đánh cùng số. Một “họa sĩ” thực hiện tô màu theo số sẽ chọn màu từ bảng màu và tô vào các vùng được đánh số tương ứng với màu đó. Máy tính lưu trữ chỉ số màu bitplane trong bit nhớ cho mỗi pixel. Các giá trị bitplane tham chiếu đến bản đồ màu và màn hình được vẽ bằng các bộ giá trị đỏ, lục và lam tương ứng từ bản đồ màu.



Hình 4.3: Minh họa sử dụng bản đồ màu để vẽ

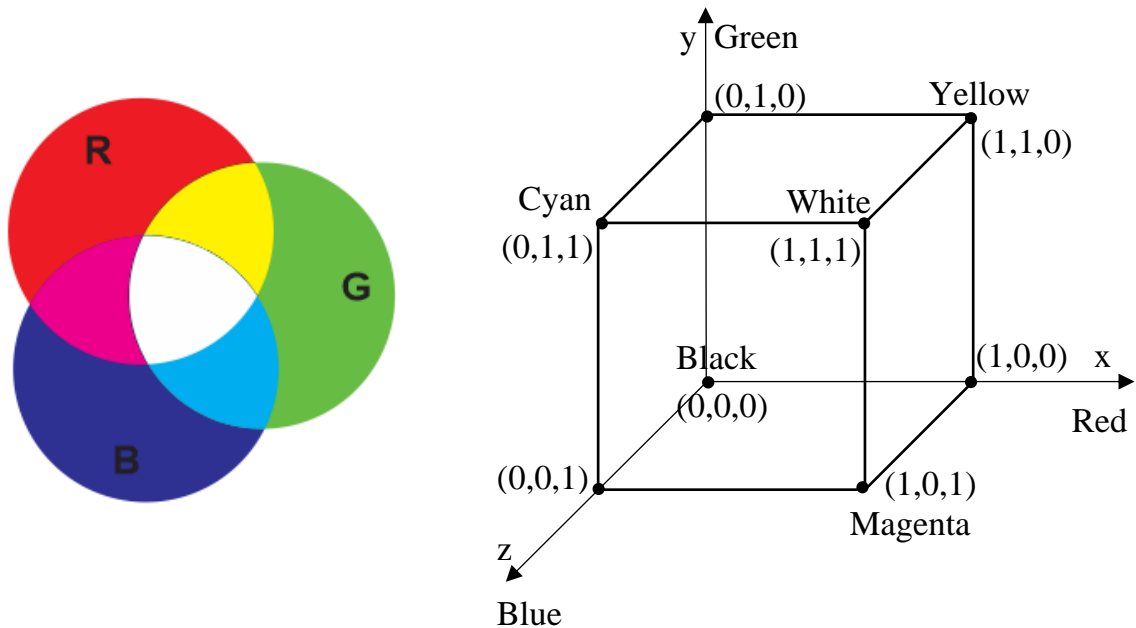
Trong chế độ chỉ mục màu, số lượng màu có sẵn đồng thời bị giới hạn bởi kích thước của bản đồ màu và số lượng bit nhớ trên mỗi pixel. Kích thước của bản đồ màu được xác định bởi phần cứng thiết bị dành riêng cho nó. Kích thước điển hình nằm trong khoảng từ 256 (2^8) đến 4096 (2^{12}). Kích thước của bản đồ màu là lũy thừa của 2, được lập chỉ mục bằng số lượng bit nhớ trong chế độ chỉ mục màu. Nếu có 2^n chỉ số trong bản đồ màu và m bit nhớ, số lượng mục có thể sử dụng là nhỏ hơn 2^n và 2^m .

Trong chế độ RGBA, màu của mỗi pixel độc lập với các pixel khác. Tuy nhiên, trong chế độ chỉ mục màu, mỗi pixel có cùng chỉ mục được lưu trong bit nhớ của nó có

chung vị trí bản đồ màu. Nếu nội dung của một mục trong bản đồ màu thay đổi, thì tất cả các pixel của chỉ mục màu đó sẽ thay đổi màu của chúng.

4.3. Hệ RGB (Red – Green – Blue):

Không gian màu RGB mô tả màu sắc bằng 3 thành phần chính là Red - Green và Blue. Không gian này được xem như một khối lập phương 3 chiều với màu red là trục x, màu Green là trục y, và màu Blue là trục z. Mỗi màu trong không gian này được xác định bởi 3 thành phần R, G, B. Ứng với các tổ hợp khác nhau của 3 màu này sẽ cho ta một màu mới.



Hình 4.4: Hệ màu RGB

Nhận xét:

- Trong hình lập phương trên, mỗi màu gốc (R,G,B) có các góc đối diện là các màu bù với nó. Hai màu được gọi là bù nhau khi kết hợp hai màu này lại với nhau ra màu trắng. Ví dụ : Green - Magenta, Red - Cyan, Blue - Yellow.
- Màn hình dựa vào cường độ màu cơ bản từ 0 – 255 kết hợp 256 màu Red, 256 màu Green, 256 màu Blue. Tổng số lượng màu sẽ là 16,7 triệu màu ($256 * 256 * 256 = 16,7$ triệu màu)

Biểu diễn không gian màu RGB trên máy tính:

1, Biểu diễn dạng số 24 bit:

Khi biểu diễn dưới dạng số, các giá trị RGB trong mô hình 24 bpp (bit per pixel) thông thường được ghi bằng cặp ba số nguyên giữa 0 và 255, mỗi số đại diện cho cường độ của màu đỏ, xanh lá cây, xanh lam.

Ví dụ: (0, 0, 0) là màu đen;

(255, 255, 255) là màu trắng;

(255, 0, 0) là màu đỏ;

(0, 255, 0) là màu xanh lá cây;
(0, 0, 255) là màu xanh lam;
(255, 255, 0) là màu vàng;
(0, 255, 255) là màu xanh ngọc;
(255, 0, 255) là màu hồng sẫm.

2, Kiểu 16 bit:

Còn có kiểu 16 bpp (bit per pixel), trong đó hoặc là có 5 bit cho mỗi màu, gọi là kiểu 555 hay thêm một bit còn lại cho màu xanh lá cây (vì mắt có thể cảm nhận màu này tốt hơn so với các màu khác), gọi là kiểu 565. Kiểu 24 bpp nói chung được gọi là thật màu, trong khi kiểu 16 bpp được gọi là cao màu.

3, Kiểu 32 bit:

Kiểu 32 bpp phần lớn là sự đồng nhất chính xác với kiểu 24 bpp, do ở đây thực sự cũng chỉ có 8 bit cho mỗi màu thành phần, tám bit dư đơn giản là không sử dụng (ngoại trừ khả năng sử dụng như là kênh alpha). Lý do của việc mở rộng của kiểu 32 bpp là vận tốc cao hơn mà phần lớn các phần cứng ngày nay có thể truy cập các dữ liệu được sắp xếp trong các địa chỉ byte có thể chia được ngang nhau theo cấp số của 2.

4, Kiểu 48 bit :

"Kiểu 16-bit" để chỉ tới 16 bit cho mỗi màu thành phần, tạo ra trong kiểu 48 bpp. Kiểu này làm cho nó có khả năng biểu thị 65.535 sắc thái mỗi màu thành phần thay vì chỉ có 255. Nó đầu tiên được sử dụng trong chỉnh sửa hình ảnh chuyên nghiệp, như Photoshop của Adobe để duy trì sự chính xác cao hơn khi có hơn một thuật toán lọc hình ảnh được sử dụng đối với hình ảnh đó. Trong trường hợp 8 bit cho mỗi màu, các sai số làm tròn có xu hướng tích lũy sau mỗi thuật toán lọc hình ảnh được sử dụng và làm biến dạng kết quả cuối cùng.

RGBA Với nhu cầu về các hình ảnh ghép đã xuất hiện phương án của RGB trong đó thêm vào kênh 8 bit dư cho độ trong suốt, vì thế tạo ra định dạng 32 bpp. Kênh trong suốt được biết đến phổ biến hơn như là kênh alpha, vì thế định dạng này có tên là RGBA. Cũng lưu ý rằng vì nó không thay đổi bất kỳ cái gì trong mô hình RGB, nên RGBA không phải là một mô hình màu khác biệt, nó chỉ là định dạng tệp (file) trong đó bổ sung thêm thông tin về độ trong suốt cùng với thông tin về màu trong cùng một tệp.

Màu sắc được sử dụng trong thiết kế web thông thường được biểu diễn với việc sử dụng RGB; xem các màu web để có giải thích cho việc sử dụng màu sắc trong ngôn ngữ HTML và các ngôn ngữ liên quan khác. Ban đầu, sự giới hạn độ sâu màu của phần lớn các màn hình đã dẫn tới sự giới hạn bảng màu là 216 màu RGB - được định nghĩa bởi Netscape Color Cube. Tuy nhiên, với sự thống trị của các thiết bị hiển thị 24-bit, việc sử dụng toàn bộ 16,7 triệu màu bằng các mã màu RGB trong mã HTML sẽ không phải là vấn đề với phần lớn người sử dụng. Nói ngắn gọn, bảng màu an toàn của web chứa 216 tổ hợp của đỏ,

xanh lá cây, xanh lam và mỗi màu có thể có 1 trong 6 giá trị (trong hệ thập lục phân hay số hex) là : #00, #33, #66, #99, #CC, hay #FF. Rõ ràng là, $6^3 = 216$.

Mô hình màu RGB cho HTML đã được chấp nhận về mặt hình thức là tiêu chuẩn Internet trong HTML 3.2, tuy nhiên nó đã được sử dụng từ trước.

4.4. Hệ CMY và CMYK (Cyan - Magenta - Yellow):

4.4.1 Hệ màu CMY:

CMY là từ viết tắt tiếng Anh của cơ chế hệ màu trừ, thường được sử dụng trong in ấn. Nó bao gồm các màu sau:

C = Cyan (xanh)

M = Magenta (hồng)

Y = Yellow (vàng)

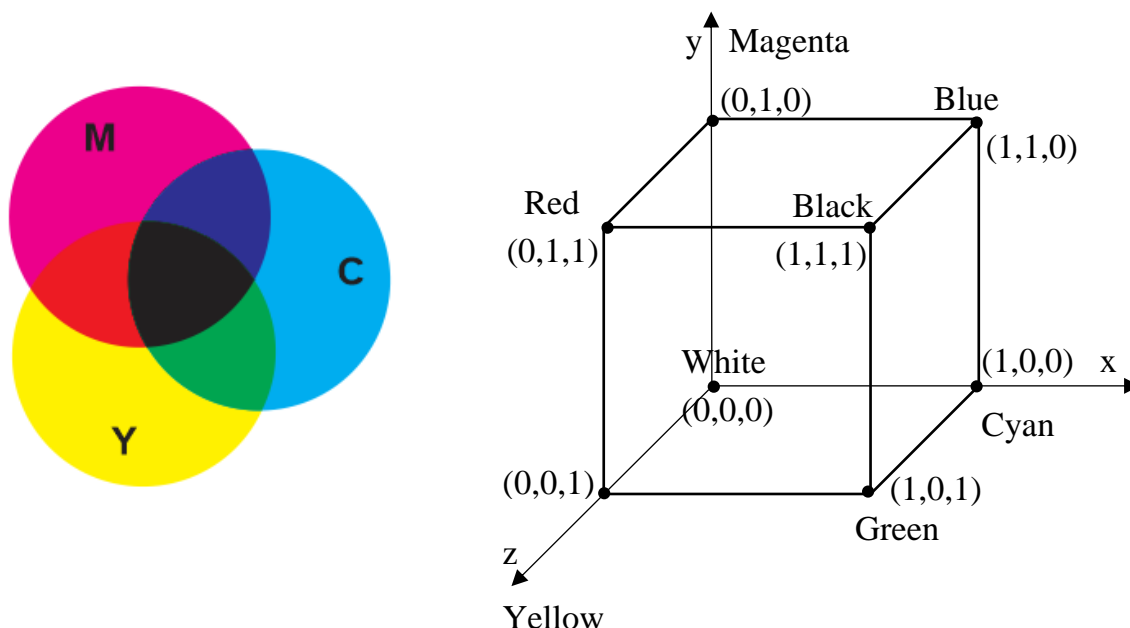
Nguyên lý làm việc của hệ CMY là hấp thụ ánh sáng. Màu mà người ta nhìn thấy là từ phần của ánh sáng không bị hấp thụ, hay nói cách khác, chúng hoạt động trên cơ chế những vật không tự phát ra ánh sáng mà chỉ phản xạ ánh sáng từ các nguồn khác chiếu tới.

Khi bật đèn bàn cho ánh sáng trắng rơi vào trang sách, ta thấy chữ in đen trên nền giấy trắng. Nền "trắng toát" (không có mực) là nơi ánh sáng của đèn bị giấy phản xạ hoàn toàn trước khi đi đến mắt. Chữ "đen thui" là chỗ ánh sáng của đèn bị mực hấp thụ hoàn toàn và không thể đi đến mắt. Nếu mực chỉ hấp thụ thành phần đỏ của ánh sáng trắng, phản xạ hoàn toàn thành phần xanh lá cây và xanh dương, ta sẽ thấy chữ có màu xanh trời nhạt (Cyan). Quả thực, màu xanh dương và màu xanh lá cây được pha trộn với liều lượng như nhau trong ánh sáng phản xạ cho ta màu xanh trời nhạt. Nói khác đi, mực có màu xanh trời nhạt là mực có tác dụng hấp thụ ánh sáng đỏ. Tương tự, mực hấp thụ ánh sáng xanh lá cây có màu tím đỏ (Magenta) do sự pha trộn của màu đỏ và màu xanh dương trong ánh sáng phản xạ. Mực hấp thụ ánh sáng xanh dương có màu vàng (Yellow) do sự pha trộn của màu đỏ và màu xanh lá cây trong ánh sáng phản xạ.

Ba màu xanh trời nhạt, tím đỏ và vàng (lần lượt tương ứng với sự hấp thụ hoàn toàn màu đỏ, xanh chuối, xanh dương của ánh sáng phát xạ) được xem là ba màu căn bản của ánh sáng phản xạ. Tùy liều lượng của mực xanh trời nhạt, tím đỏ và vàng, sự hấp thụ các thành phần đỏ, xanh lá cây và xanh dương trong ánh sáng trắng xảy ra mạnh yếu khác nhau, ta sẽ thấy những màu sắc khác nhau trên giấy.

Một cách khái quát, màu bất kỳ của vật phản xạ được xác định bằng bộ ba trị số biểu diễn cường độ màu xanh da trời, tím đỏ và vàng. Người ta gọi cách thức xác định màu sắc như vậy là mô hình màu CMY (CMY color model). CMY là viết tắt của Cyan – Magenta – Yellow. Màu trắng trong mô hình này được biểu diễn dưới dạng: (C = 0, M = 0, Y = 0) hoặc gọn hơn (0C-0M-0Y).

Với màu CMY: Trộn màu xanh (Cyan) với màu hồng (Magenta) sẽ cho ra màu xanh dương (Blue), màu hồng (Magenta) với màu vàng (Yellow) sẽ cho ra màu đỏ (Red), màu xanh (Cyan) với màu vàng (Yellow) sẽ cho ra màu xanh lá cây (Green), ba màu Cyan, Magenta, Yellow kết hợp lại sẽ cho ra màu Đen (Black).



Hình 4.5: Hệ màu CMY

Màu trắng tương ứng với gốc tọa độ trong một không gian 3 chiều được định vị bởi 3 trục tọa độ C, M và Y. Màu đen và các sắc xám tạo bởi liều lượng bằng nhau của các thành phần C, M và Y. Mô hình RGB và mô hình CMY chẳng qua chỉ là hai "cách nhìn" đối với màu sắc. Trong mô hình RGB, màu sắc được tạo ra bởi sự đóng góp của các thành phần RGB với cường độ khác nhau. Cường độ cao của các thành phần RGB cho ta màu trắng. Khi cường độ của chúng bằng 0, ta có màu đen (không có ánh sáng).

Trong mô hình CMY, ta lại chú ý đến hiệu quả của các thành phần CMY trong việc trừ khử các thành phần RGB của ánh sáng tới. Liều lượng cao của các thành phần CMY cho ta màu đen. Liều lượng của chúng bằng 0 cho ta màu trắng. Do vậy, người ta gọi mô hình RGB là mô hình màu cộng tính (additive) và mô hình CMY là mô hình màu hiệu tính (subtractive).

4.4.2 Hệ màu CMYK:

- Là mô hình mở rộng của CMY ứng dụng trong máy in màu. Giá trị đen (K) bổ xung vào thay thế cho hàm lượng màu bằng nhau của 3 màu cơ bản. Từ CMYK (hay đôi khi là YMCK) là từ viết tắt trong tiếng Anh để chỉ mô hình màu trừ phổ sử dụng trong in ấn màu. Chữ K bắt nguồn từ màu đen - black.

Nguyên lý làm việc của CMYK là trên cơ sở hấp thụ ánh sáng. Màu mà người ta nhìn thấy là từ phần của ánh sáng không bị hấp thụ. Trong hệ màu CMYK, màu hồng sẫm (magenta) kết hợp với màu vàng sẽ cho màu đỏ, hồng sẫm (magenta) kết hợp với xanh lơ

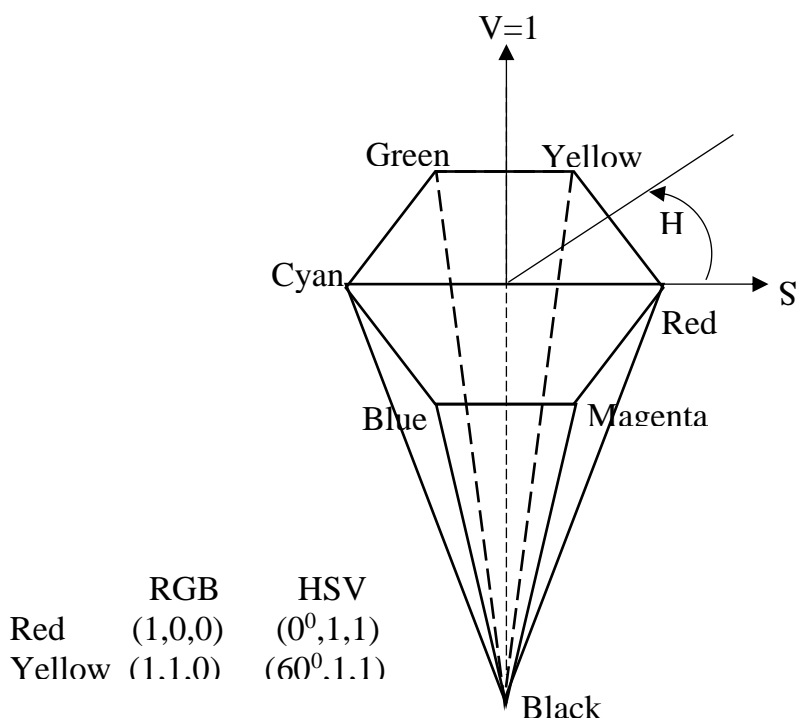
(cyan) cho màu xanh lam (blue), xanh lơ (cyan) kết hợp với vàng sinh ra màu xanh lá cây và tổ hợp của các màu xanh lơ, hồng sẫm và vàng tạo ra màu đen. Vì màu 'đen' sinh ra bởi việc trộn các màu gốc là không thực sự giống như mực đen, hay màu đen của vật đen tuyệt đối (là vật hấp thụ toàn bộ ánh sáng), việc in ấn trên cơ sở bốn màu (đôi khi gọi là in các màu mặc dù điều này không chính xác) phải sử dụng mực đen để bổ sung thêm vào với các màu gốc loại trừ là các màu vàng, hồng sẫm và xanh lơ.

Việc sử dụng công nghệ in ấn bốn màu sinh ra kết quả in ấn cuối cùng rất cao cấp với độ tương phản cao hơn. Tuy nhiên màu của vật thể mà người ta nhìn thấy trên màn hình máy tính thông thường có sự sai khác chút ít với màu của nó khi in ra vì các mô hình màu CMYK và RGB (sử dụng trong màn hình máy tính) có các gam màu khác nhau. Mô hình màu RGB là mô hình dựa trên cơ sở phát xạ ánh sáng (màu bổ sung) trong khi mô hình CMYK làm việc theo cơ chế hấp thụ ánh sáng (màu loại trừ).

4.5 Một số hệ màu khác

4.5.1. Hệ màu HSV (Hue - Saturation - Value):

Thực chất của không gian này là sự biến đổi của không gian RGB. Không gian HSV được mô tả bằng lệnh lập phương RGB quay trên đỉnh Black. H (Hue) là góc quay trục V



Hình 4.6: Hệ màu HSV

(value) qua 2 đỉnh Black và White (Hình 4-8).

Các giá trị biến thiên của H, S, V như sau :

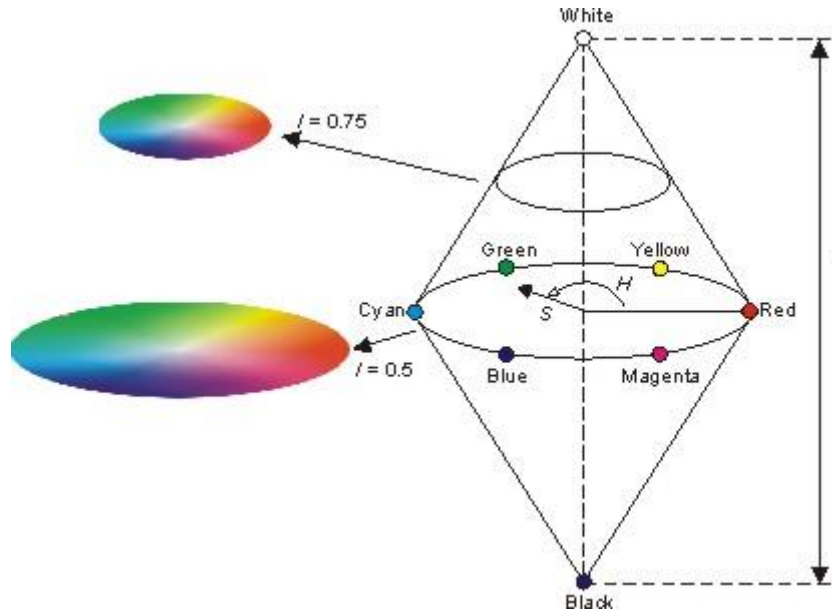
- H (Hue) chỉ sắc thái có giá trị từ 0° - 360° .
- S (Saturation) chỉ độ bão hoà.

- V (Value) có giá trị từ 0 - 1.

Các màu đạt giá trị bão hòa khi $s = 1$ và $v = 1$.

4.5.2. Hệ màu HSI (Hue - Saturation - Intensity):

Hệ thống màu mã hóa thông tin màu sắc bằng cách chia giá trị Intensity (I) từ 2 giá trị được mã hóa thuộc về độ hội tụ màu - Hue (H) và Saturation (S).



Hình 4.7: Hệ màu HSI

Thành phần không gian màu HSI gồm có 3 phần:

- Hue được định nghĩa có giá trị $0-2\pi$, mang thông tin về màu sắc.
- Saturation có giá trị 0-1, mang giá trị về độ thuần khiết của thành phần Hue.
- Intensity (value) có giá trị 0-1 mang thông tin về độ sáng của điểm ảnh.

Có thể hình dung không gian màu HSI như là vật hình nón với trục chính biểu thị cường độ sáng Intensity, khoảng cách đến trục biểu thị độ tập trung Saturation, góc xung quanh trục biểu thị cho sắc màu Hue. Đôi khi, hệ thống màu HSI được coi như là hệ thống màu HSV dùng Value thay vì Intensity.

Hệ thống màu HSI thì thích hợp hơn với một số thiết kế đồ họa bởi vì nó cung cấp sự điều khiển trực tiếp đến ánh sáng và sắc độ. Hệ thống màu HSI cũng hỗ trợ tốt hơn những thuật toán xử lý ảnh vì tiêu chuẩn hóa về ánh sáng và tập trung vào hai tham số về độ hội tụ màu và cường độ màu. Hệ thống màu HSI có sự phân chia rõ rệt giữa ánh sáng và màu sắc do đó có khả năng rất lớn được áp dụng cho việc tính đặc trưng và so sánh sự giống về màu sắc của hai ảnh nên thích hợp cho việc tìm kiếm ảnh dựa vào màu. Sự giống và khác nhau giữa hai ảnh về màu sắc đối với mắt người chỉ là tương đối.[5]

Ứng dụng phối màu

Màu đơn sắc (Monochromatic)

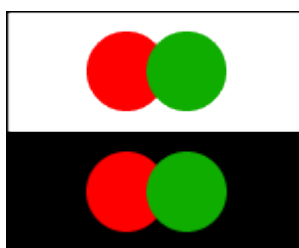
Monochromatic là cách trộn màu bằng Tones, Shades và Tints trong một màu cụ thể. Nó là cách trộn màu đơn giản nhất



Hình 4.8: Màu đơn sắc (Monochromatic)

Kết hợp màu kiểu bổ sung

Những màu ở đối diện nhau trong phổ màu được gọi là màu bổ sung (Ví dụ: đỏ và xanh lá). Khi sử dụng cách kết hợp này nên cân nhắc kỹ lưỡng vì 2 màu bổ sung thực chất tương phản nhau nên có thể gây khó chịu, chói mắt cho người xem



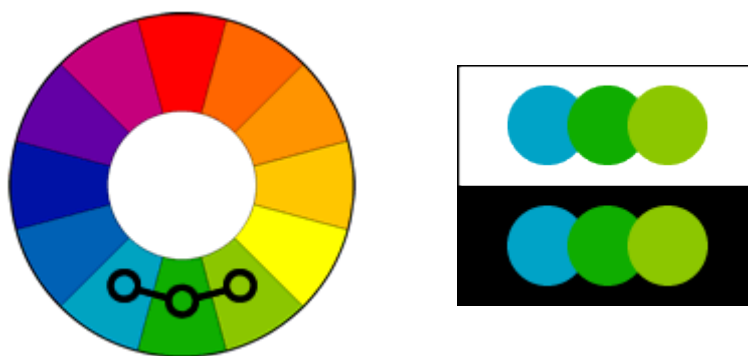
Hình 4.9: Kết hợp màu kiểu bổ sung

Nếu dùng trên diện tích lớn có thể khá khó khăn và ít hiệu quả, nhưng nếu muốn tạo điểm nhấn ở một vài đối tượng thì nó sẽ là công cụ hiệu quả

Kết hợp màu kiểu tương tự

Màu tương tự là những màu nằm cạnh nhau trong phổ màu. Trong tự nhiên ta có thể dễ dàng thấy những thứ có kết hợp màu theo kiểu này, như trên một cái cây có lá đậm, lá nhạt, lá non, lá già... nhưng đều mang màu sắc tương cận nhau. Kiểu kết hợp này sẽ tạo cảm giác dễ chịu cho mắt khi nhìn vào.

Nhưng có một bất lợi đó là phải chọn màu sao cho chúng có mức độ tương phản nhất định với nhau. Lời khuyên đặt ra, nên chọn một màu làm chủ đạo trước đã, chọn một màu nữa làm màu phụ và màu thứ 3 (cùng với đen, trắng, xám) chỉ như một chất phụ gia thêm vào thôi

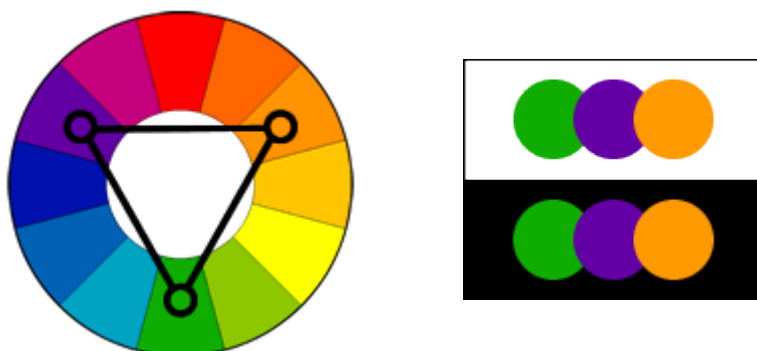


Hình 4.9: Kết hợp màu kiểu bổ sung

Kết hợp màu kiểu bộ ba

Cách kết hợp này được thực hiện bằng việc chọn ba màu ở ba đỉnh của một tam giác đều đặt vào phổ màu. Cần lưu ý rằng dù có sử dụng màu trông hơi tái hay có cấp độ bão hòa thấp, kiểu màu bộ 3 cũng tạo cảm giác chói mắt.

Hãy lưu ý cân bằng màu. Có thể chọn một màu làm màu chính và để 2 màu còn lại như công cụ tôn màu chính

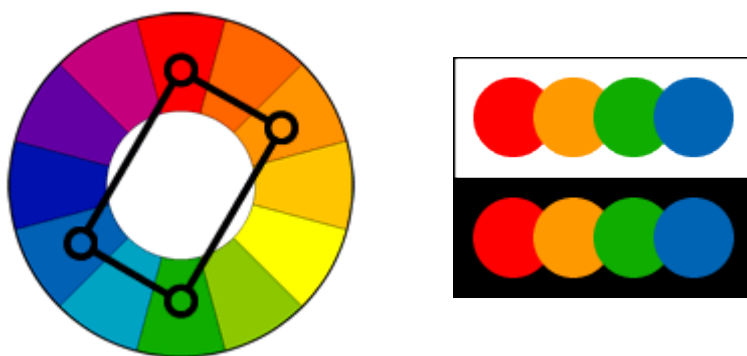


Hình 4.10: Kết hợp màu kiểu bộ ba

Kết hợp màu kiểu hình chữ nhật

Cách kết hợp theo kiểu chữ nhật chọn ra bốn màu được sắp xếp thành hai cặp tương phản nhau. Cách lựa chọn này đưa ra rất nhiều lựa chọn khác nhau, chú ý tới sự cân bằng giữa màu nóng và màu lạnh trong thiết kế.

Ví dụ: Đỏ, cam, lá cây, xanh dương

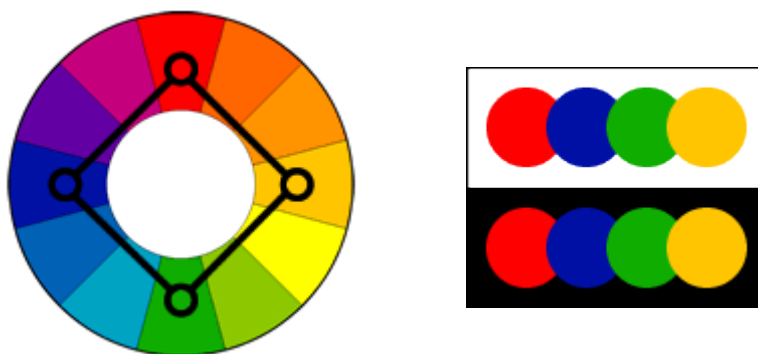


Hình 4.11: Kết hợp màu kiểu hình chữ nhật

Kết hợp màu kiểu hình vuông

Cách kết hợp màu theo kiểu hình vuông cũng tương tự với kiểu hình chữ nhật, nhưng với tất cả bốn màu đặt xếp cách đều nhau xung quanh vòng tròn màu sắc.

Giống kiểu hình chữ nhật, cũng nên chọn một màu chủ đạo và chú ý cân bằng nóng - lạnh trong màu sắc để đạt được hiệu quả tối đa như mong muốn.



Hình 4.12: Kết hợp màu kiểu hình vuông

Câu hỏi ôn tập chương:

1. Anh (chị) hãy cho biết mô hình màu nào trong các mô hình đã học là mô hình màu định hướng phần cứng, mô hình nào định hướng người sử dụng?
2. Anh (chị) hãy trình bày mối liên hệ giữa các mô hình màu.
3. Anh (chị) hãy cho biết có thể in màu blue trên một tấm bìa màu cyan được không, nếu được thì ta cần cho thêm mực màu gì vào?
4. Anh (chị) hãy cho biết có thể in màu blue trên một tấm bìa màu yellow được không, nếu được thì ta cần cho thêm mực màu gì vào?
5. Anh (chị) hãy giải thích việc thêm màu đen – black vào hệ màu CMY ?

Câu hỏi trắc nghiệm

1. Mô hình màu CMY là gam màu thể hiện :
 - a. Đỏ tươi - Lục - Lam
 - b. Xanh tím - Đỏ tươi - Vàng
 - c. Hồng nhạt - Đỏ Sậm - Vàng nhạt
 - d. Đỏ Sậm - Vàng nhạt - Tím
2. Mô hình màu CMY là :
 - a. Đây là mô hình màu bù (Subtractive color models) hiển thị ánh sáng và màu sắc phản xạ từ mực in
 - b. Hệ thống có quy tắc cho việc tạo khoảng màu từ tập các màu cơ bản
 - c. Mô hình kết quả của ánh sáng truyền dẫn - transmitted
 - d. Mô hình màu bù sử dụng mực in - printing inks
3. Trong mô hình màu CMY, khi bề mặt không phủ mực thì ánh sáng phản xạ là :
 - a. Ánh sáng đen – Black
 - b. Ánh sáng trắng - White
 - c. Ánh sáng Xanh Tím – Cyan
 - d. Ánh sáng Đỏ - Red
4. Trong mô hình màu CMY, khi 3 màu có cùng giá trị cho ra :
 - a. Màu Trắng
 - b. Màu Đen
 - c. Màu Xám
 - d. Màu Nâu
5. Trong mô hình màu CMY, các giá trị đạt max cho ra :
 - a. Màu Trắng
 - b. Màu Đen
 - c. Màu Xám
 - d. Màu Nâu
6. Trong mô hình màu CMY, Red +Cyan ; Green +Magenta ; Blue + Yellow sẽ cho kết quả là :
 - a. Màu Trắng
 - b. Màu Đen
 - c. Màu Xám
 - d. Màu Nâu

d. HSI

16. Hệ màu mà con người cảm nhận là:

a. CIE

b. CMY (Cyan - xanh tím, Magenta - đỏ tươi và Yellow - vàng)

c. Hue (sắc màu), Saturation (độ bão hoà) và Lightness (độ sáng)

d. RGB (Red - đỏ, Green - lục, Blue - lam)

17. Ta có ba hộp màu nước đỏ (Red), lục (Green) và lam (Blue) đem trộn các màu bão hoà và cân bằng thì thu được màu:

a. Đen

b. Trắng

c. Đỏ

d. Chưa xác định

18. Ta có 3 hộp màu Cyan (xanh tím), Magenta (đỏ tươi) và Yellow (vàng) trộn bão hoà thì được màu (theo lý thuyết):

a. Xanh lục

b. Đen

c. Trắng

d. Đỏ

19. Chiếu 1 chùm sáng trắng vào 1 tấm bìa, ta nhận thấy tấm bìa ấy có màu cyan, thì có nghĩa là những màu cơ bản nào của chùm sáng đã bị tấm bìa hấp thụ?

a. Yellow, magenta

b. Red

c. Green, blue

c, Magenta

20. Chiếu 1 chùm sáng trắng vào 1 tấm bìa, ta nhận thấy tấm bìa ấy có màu magenta, thì có nghĩa là những màu cơ bản nào của chùm sáng đã được tấm bìa phản xạ?

a. Yellow, cyan

b, Red, blue

c. Green

c, Magenta

21. Muốn in màu đỏ (red) lên một tấm bìa màu vàng (yellow) thì ta cần bổ sung mực màu gì?

a. Magenta

b. Blue

c. Green

d. Cyan

22. Muốn in màu xanh lục (green) lên một tấm bìa màu vàng (yellow) thì ta cần bổ sung mực màu gì?

a. Magenta

b. Blue

c. Green

d. Cyan

23. Trong mô hình màu RGB (Red, Green, Blue) thì màu đen có tọa độ là:

a. (0,0,0)

b. (1,0,0)

c. (1,1,1)

d. (0,0,1)

24. Trong mô hình màu RGB (Red, Green, Blue) thì màu trắng có tọa độ là:

a. (0,0,0)

b. (1,0,0)

c. (1,1,1) d. (0,0,1)

25. Trong mô hình màu CMY-K(mô hình ứng dụng trong máy in) thì màu trắng có tọa độ là:

a. (0,0,1) b. (1,0,0)

c. (1,1,1) d. (0,0,0)

26. Khoảng màu mà chúng ta tạo ra với tập các màu cơ bản gọi là :

a. Gam màu hệ thống b. Gam màu riêng gamut

c. Ánh sáng phản xạ d. Tất cả đều sai

27. Hệ HSV có H (sắc màu) chạy từ :

a. 0° đến 360° b. 0° đến 180°

c. 0° đến 90° d. 90° đến 360°

28. Hệ HSV có S (độ bão hòa) và V (giá trị cường độ ánh sáng) thuộc khoảng :

a. [0.. 0,1] b. [0..1]

c. [1..2] d. [0..2]

29. Hệ HSV có H (sắc màu) màu đỏ ở :

a. 0° b. 45°

c. 90° d. 180°

CHƯƠNG 5: CÁC KỸ THUẬT ĐỒ HỌA TRONG BỘ CÔNG CỤ ADOBE AFTER EFFECT

Mục tiêu của chương: Sinh viên biết sử dụng công cụ Adobe After Effect, nắm được quy trình tạo ra các hiệu ứng đồ họa và một số kỹ thuật tạo hình cơ bản. Hiểu được và thực hiện được một số cách làm kỹ xảo, chuyển động, timing, ... đơn giản.

Sinh viên nắm được bố cục, thành phần của giao diện cũng như cách quản lý một Project hay những effect thường dùng trong phim ảnh.

5.1 Giới thiệu về After Effect

After Effects (AE) là một trong các chương trình xử lý phim chuyên nghiệp (Edit film) và dễ sử dụng do hãng Adobe phát triển. Cũng giống như chương trình Photoshop xử lý ảnh tĩnh (still images), AE cho các chức năng xử lý ảnh động (movie), còn gọi là làm kỹ xảo phim ảnh. AE tương thích với các sản phẩm đồ họa, xử lý ảnh của hãng Adobe như: Illustrator, Photoshop, Premiere Pro, Encore hay Flash...

Chức năng chính của After Effects giúp ta có thể làm được những chuyển động đồ họa ấn tượng, đẹp mắt. Ví dụ như những clips quảng cáo hay thước phim sống động, đẹp mắt, hấp dẫn và đầy lôi cuốn. Nếu ta là một người yêu thích ngành thiết kế đồ họa, kỹ xảo trong TVC quảng cáo hay game hoặc truyền hình thì đây là một công cụ không thể thiếu. Để dễ hình dung AE chính là phần mềm đứng sau những thước phim hoạt hình đẹp mắt. Những đoạn quảng cáo, với hiệu ứng lung linh huyền ảo. Những kỹ xảo phim ảnh trong các bộ phim võ thuật cổ trang. Những hiệu ứng cháy nổ phần nhiều đều được làm ra bởi phần mềm đồ họa này.

5.1.1 Ứng dụng của after effect:

Adobe After Effects (phần mềm kỹ xảo) giúp tạo ra những chuyển động đồ họa hay và ấn tượng, hỗ trợ làm phim và chỉnh sửa ảnh chuyên. Không những thế phần mềm còn có khả năng tương thích tốt với nhiều phần mềm chỉnh sửa ảnh và mang đến hệ thống theo dõi camera 3D, công nghệ 3D cho text và hình khối.

Adobe After Effects tương thích hợp với những phần mềm đồ họa nổi tiếng như Adobe Illustrator nên ta có thể kết hợp để nâng cao hiệu quả chỉnh sửa ảnh. Còn để tạo hiệu ứng hình ảnh và đồ họa chuyển động nhanh hơn, kiểm soát chiều sâu, bóng, phản xạ, công nghệ 3D Camera Tracker có khả năng theo dõi những thành phần 3D thì Adobe After Effects tương thích với những công nghệ đồ họa hiện đại như Global Performance Cache

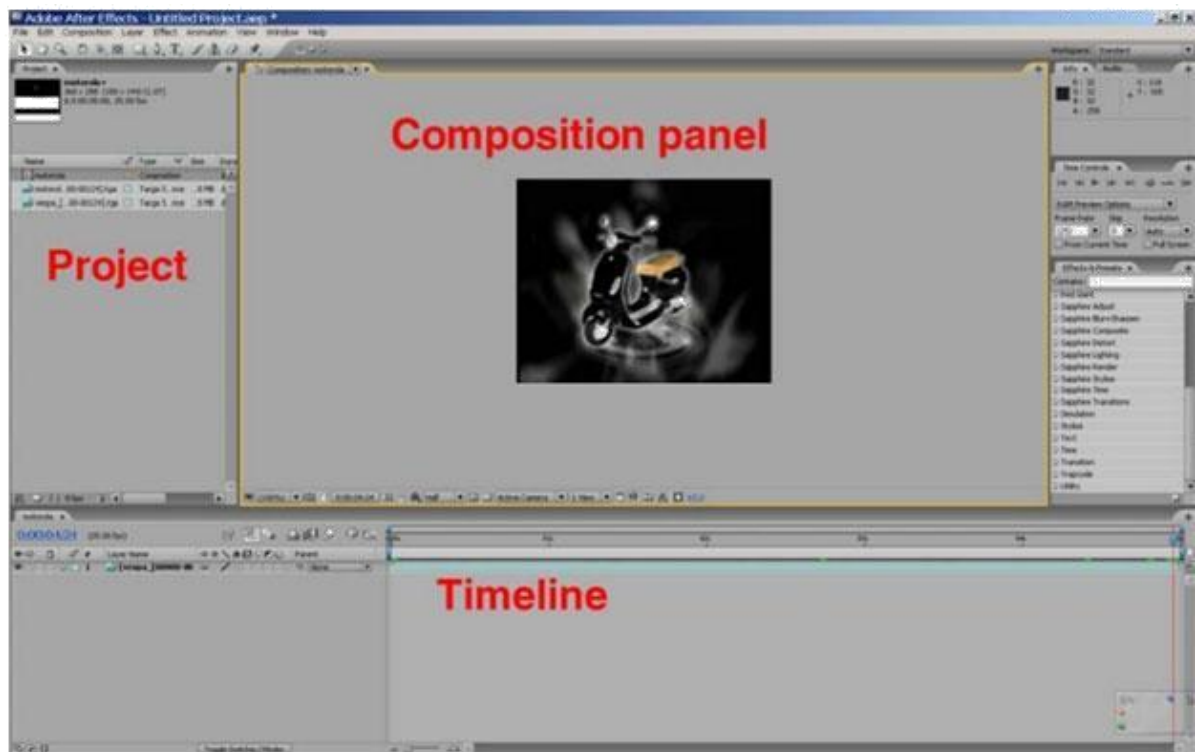
5.1.2 Tính năng chính của after effect cs6:

- Global Performance Cache: tạo hiệu ứng hình ảnh và đồ họa chuyển động nhanh hơn.

- 3D Camera Tracker: theo dõi những thành phần 3D, kiểm soát chiều sâu, bóng, phản xạ.
- Dò tia, ép text và khối
- Tạo đường viền chuẩn
- Tích hợp với Adobe Illustrator
- Sửa màn chụp cuộn
- Hiệu ứng mới và cập nhật
- Import file Avid AAF và FCP 7 XML với Pro Import AE.

5.2. Giao diện

Đầu tiên chúng ta sẽ làm quen về Giao diện (Interface), hay Vùng làm việc (WorkSpace) của After Effects (AE).



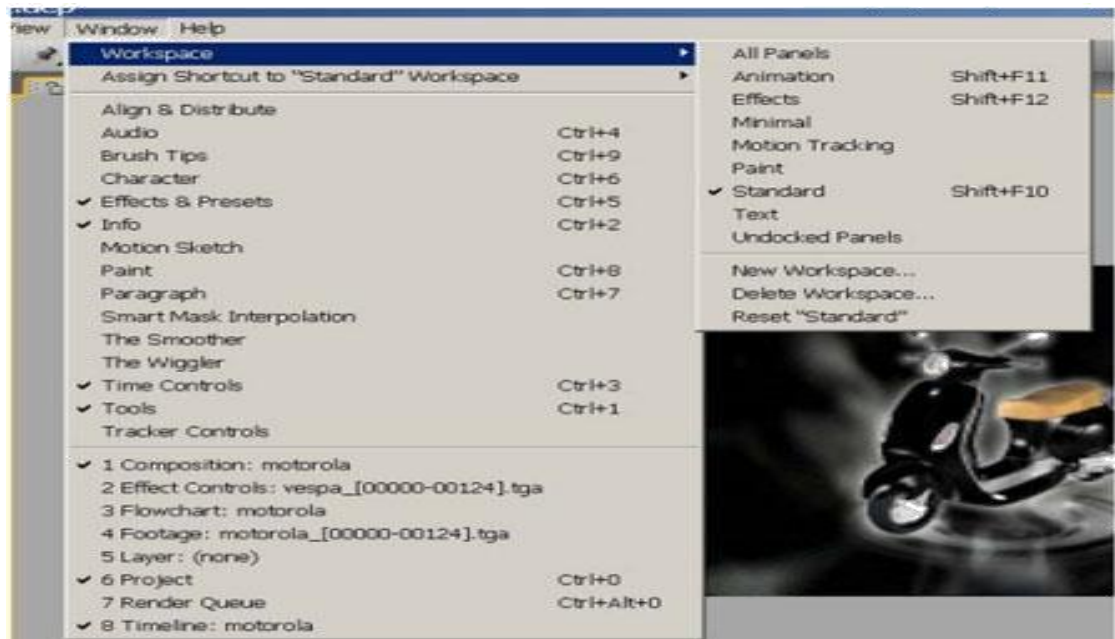
Hình 5.1: Giao diện After Effect

Vùng nhìn Project: Quản lý các files import.

Composition panel: Giống như monitor, sẽ hiển thị kết quả sau khi xử lý.

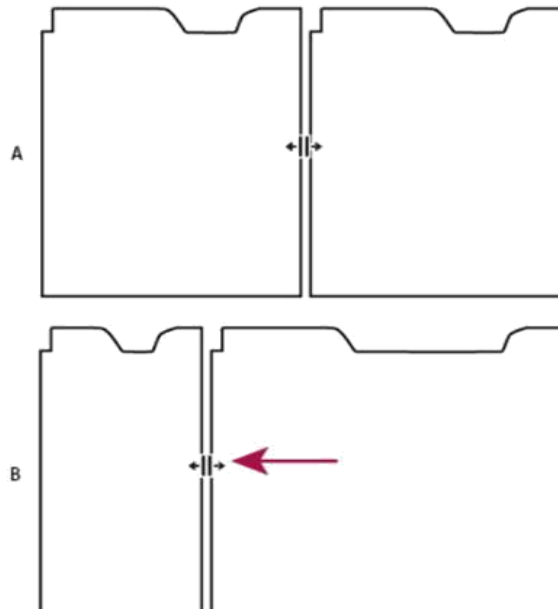
Timeline: Điều khiển animation, effects... theo thời gian.

Có thể chọn open một workspace định sẵn của AE:



Hình 5.2: Mở workspace của After Effect

Hoặc chúng ta điều chỉnh kích thước của từng panel, bằng cách đưa con trỏ chuột vào giữa hai panel:

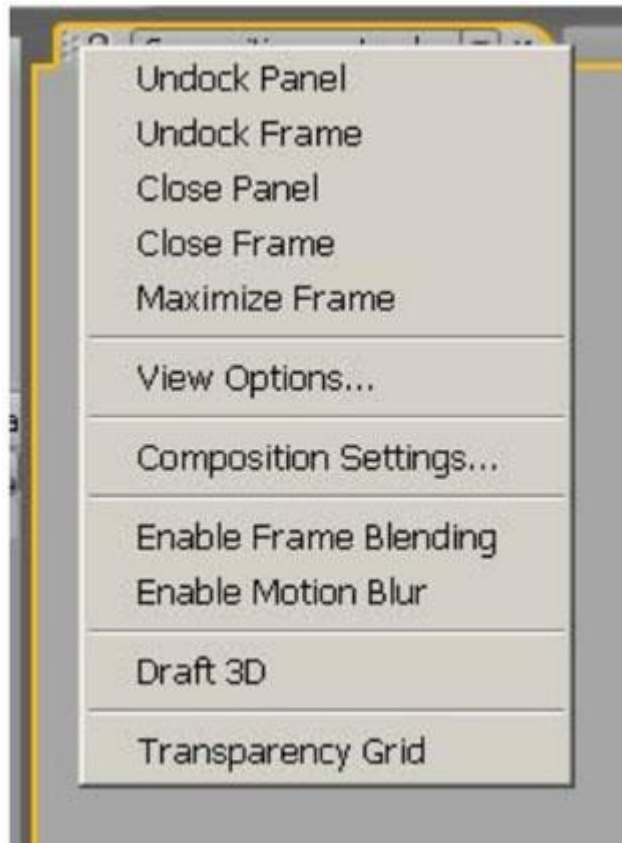


Hình 5.3: Panel trong After Effect

Dock, group, or float panels:

Có thể di chuyển các panel bằng cách kéo các chúng sắp xếp theo từng nhóm (group) khác nhau.

Hoặc right-click góc trên panel, chọn undock panel.



Hình 5.4: Các thao tác với Panel trong After Effect

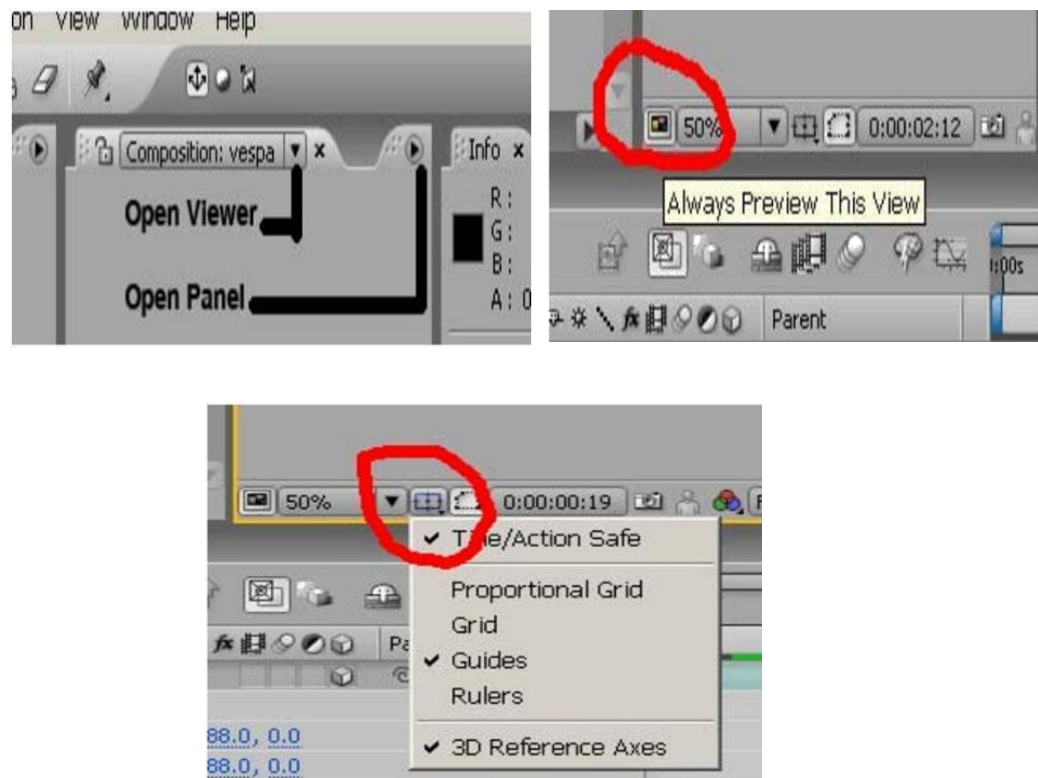
Các icon trong panel:

- Dùng mouse wheel, có thể phóng lớn thu nhỏ vùng nhìn composite.
- Alt + mouse wheel, phóng lớn vùng nhìn Timeline.
- Giữ Spacebar + mouse left để pan.
- Right click trên toolbar có thể ẩn hoặc hiện các cột như hình vẽ.

Hoặc tại góc mỗi viewer có các icon tam giác ta có thể click vào :

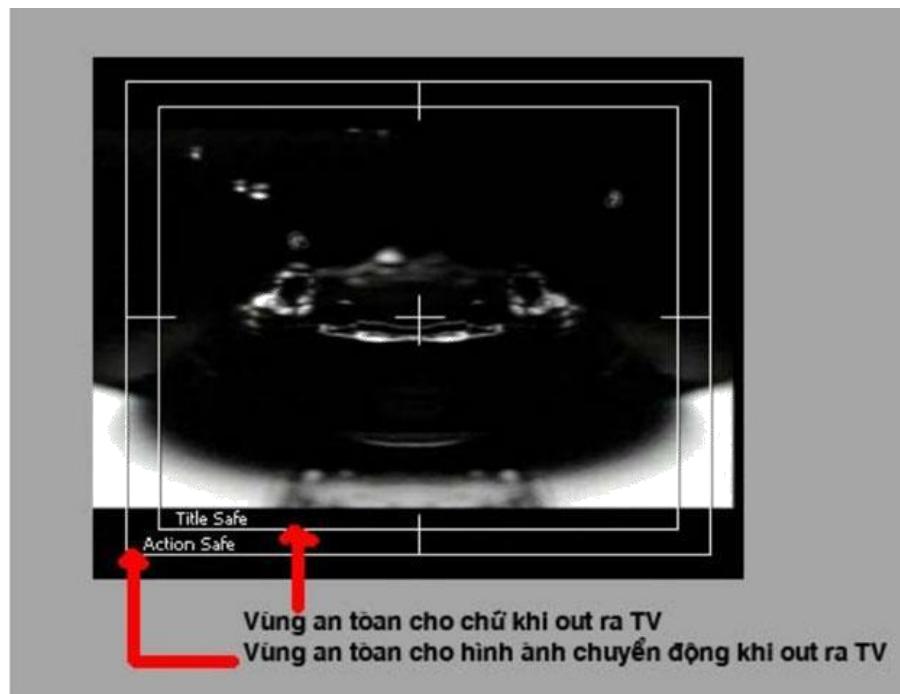
Tại vùng nhìn Composition (góc trái bên dưới), click vào icon (Always Preview This view) thì mỗi lần thực hiện preview chỉ cửa sổ này mới view lên.

- Ở icon kế bên, bật Title/Action Safe :



Hình 5.5: Các icon Panel trong After Effect

Lúc này sẽ xuất hiện vùng nhìn an toàn trên viewer composition :



Hình 5.6: Vùng an toàn trong After Effect

Cũng như các chương trình khác, thanh Toolbar trên góc trái màn hình có các chức năng :

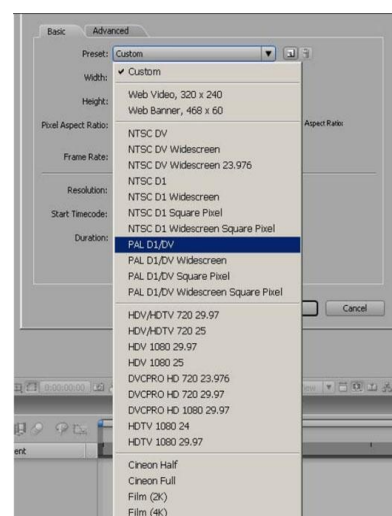
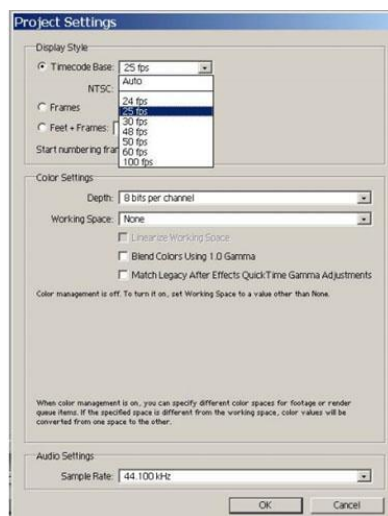
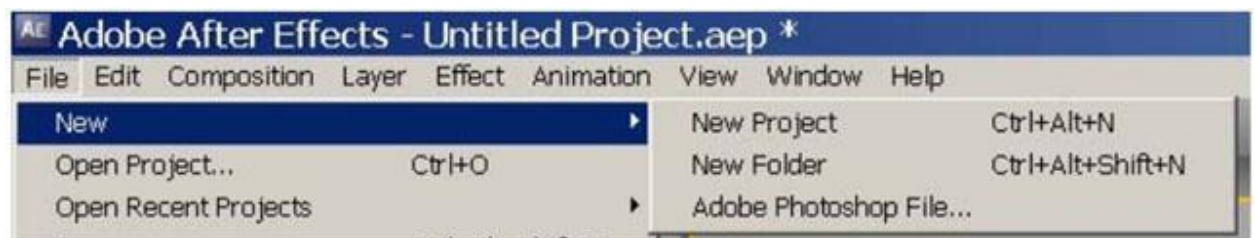


Hình 5.7: Toolbar trong After Effect

5.3. Thực hiện một Project

Trước khi bắt tay vào edit hay làm effects một đoạn phim nào đó, chúng ta phải sắp xếp các hình ảnh, âm thanh, kịch bản.... Tức là thực hiện một Project.

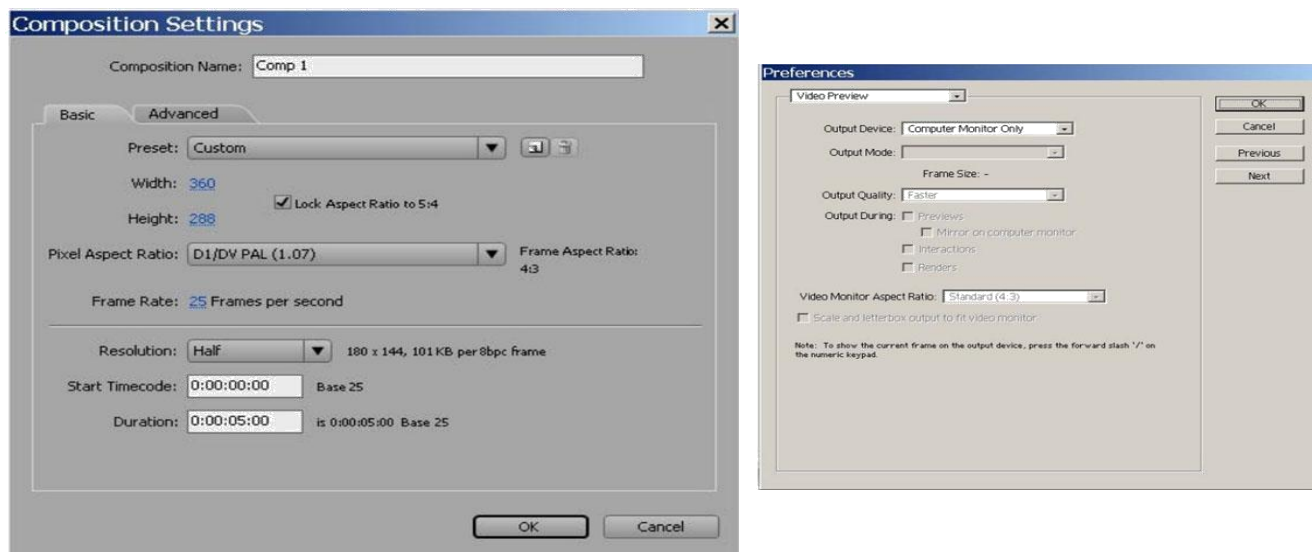
1 – Setup Projects: Click vào File/New/ New Project/project settings:



Hình 5.8: Chức năng New Project trong After Effect

Tại hàng Timecode base, chọn 25fps (25 frame per secon cho hệ PAL video) và Audio setting chọn Sample Rate 41.100 kHz. Tiếp đến chọn tại menu Composition/New Composition (Ctrl+N):

Tại thanh Preset chọn kích cỡ của phim (Nếu làm hệ PAL thì chọn PAL D1/DV mà các kênh truyền hình trong nước thường phát). Hoặc chọn các chuẩn mà nhà sản xuất yêu cầu.

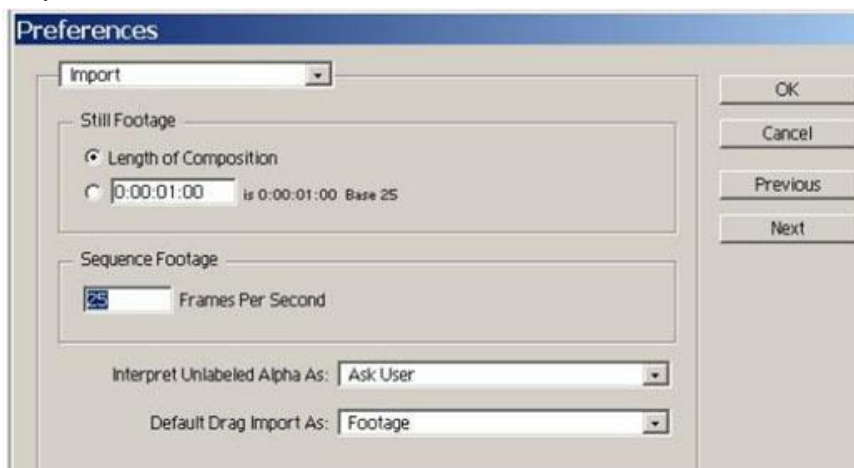


Hình 5.9: Hiệu chỉnh Preferences trong After Effect

Đặt tên tại Composition Name

Ở thanh Resolution (độ phân giải) chọn half hay full..., thường chọn half cho nhẹ máy. Duration (thời lượng đoạn phim), chọn đúng chiều dài đoạn phim mà ta chuẩn bị xử lý (giả sử đoạn phim Quảng các 30s, gõ vào số 30 thay thế vào số 05 này-giờ:phút:giây:frame) .

Tiếp tục chọn ở menu Edit/Preferences/General... bấm next đến thanh Import:



Hình 5.10: Hiệu chỉnh Preferences trong After Effect

Chọn tại Sequence Footage 25 Frames Per Second nếu là hệ PAL video.

Phần Still Footage (ảnh tĩnh) chọn Length of Composition.

Nhấn next qua phần Video preview, chọn Output Device: Computer Monitor Only nếu chỉ preview trên máy tính (không có card dựng phim)

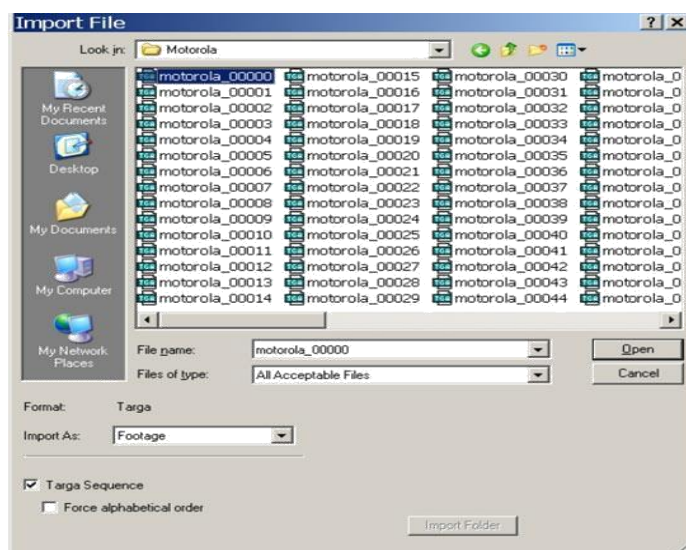
Như vậy chúng ta đã chuẩn bị xong một project theo cấu hình yêu cầu.

2 - Importing:

Bước kế tiếp là lấy hình ảnh, movie, capture, âm thanh... vào chuẩn bị xử lý.

Chọn tại menu File/Import hoặc Double click vào project panel:

Phần Files of type sẽ nhìn thấy những file mà AE cho phép import. Phần Targa sequence sẽ active nếu ta có một chuỗi hình được đánh thứ tự liên nhau



Hình 5.11: Import File trong After Effect

- Nếu chọn Import là file Photoshop thì sẽ xuất hiện menu:

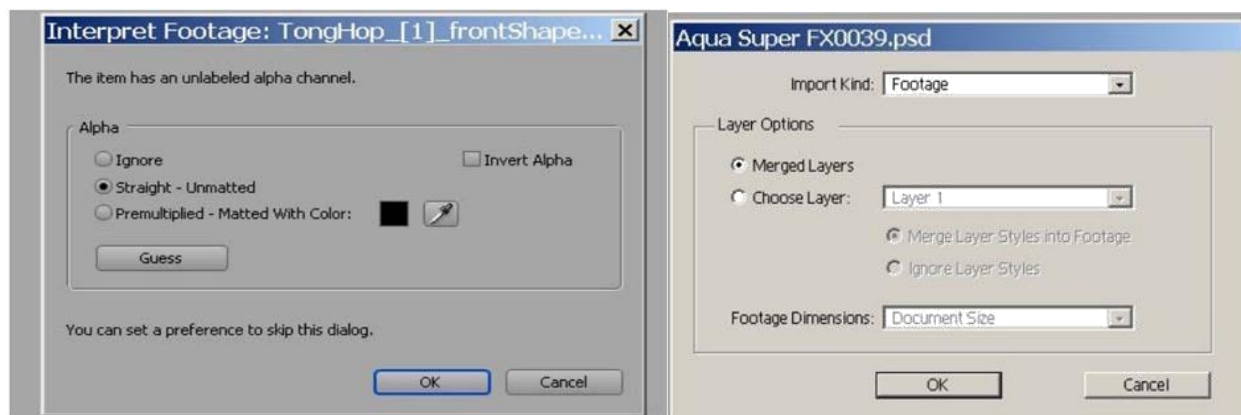
Và chọn từng layer hay merged lại tùy nhu cầu.

- Nếu file ta chọn có alpha channel (32bit) thì sẽ xuất hiện:

- Ignore: bỏ qua channel alpha.
- Straight – Unmatted: nếu file 32bit (như Targa 30bit)

Premultiplied – Matted with color: chương trình sẽ tự tạo phần mate theo màu background mà ta chọn (BG phải là một màu đồng nhất).

Sau khi import vào, ta kéo các files từ Project panel thả vào Timeline panel. Ta có thể import nhiều file và kéo thả nhiều files vào Timeline panel này, lúc đó sẽ xuất hiện nhiều layer như layer bên Photoshop.



Hình 5.12: Hiệu chỉnh thông số Import File trong After Effect

5.4. Kỹ thuật Storyboard

Storyboard là cách nhà sản xuất thể hiện kịch bản bằng hình vẽ, những điều không phải hình vẽ được thể hiện, đặc tả ở ngoài khung hình

Storyboard như khung sườn mà từ khung sườn này diễn viên và đoàn làm phim có thể hiểu và cùng nhau làm việc một cách hiệu quả nhất.

Các Bước Xây Dựng Một Storyboard:

1. Lên shot List

Shot list là danh sách tất cả những cảnh quay sẽ xuất hiện trong đoạn phim quảng cáo. Thực hiện sắp xếp những cảnh quay này theo bối cảnh. Khi đến địa điểm thật, dựa theo shot list này để đảm bảo không quay thiếu bất cứ cảnh nào trong phim:

- Bối cảnh ở đâu, sắp xếp trật tự bối cảnh như thế nào?
- Có bao nhiêu diễn viên trong phân cảnh (shot hình) đó.
- Cần bao nhiêu đạo cụ hỗ trợ, đạo cụ nào quan trọng trong phân cảnh này.
- Shot hình nào sẽ được lấy trong phân cảnh này (cận cảnh, cảnh rộng, cú máy thiết lập...).
- Góc máy trong phân cảnh này là gì (góc máy cao hay thấp).
- Phân đoạn này sẽ có nhân vật nào di chuyển hay thiết bị, phương tiện/ đạo cụ nào chuyển động hoặc xuất hiện.
- Hướng di chuyển của camera theo diễn viên, sẽ có bao nhiêu camera trong cảnh quay, camera nào có nhiệm vụ chính trong phân cảnh này.
- Ánh sáng, hiệu ứng nào đặc biệt được dùng trong phân cảnh của TVC.

2. Chọn khung hình:

Khung hình là đôi mắt của khán giả dõi theo đoạn phim quảng cáo.

Khung hình xấu hoặc không phù hợp với nội dung, thông điệp sẽ không diễn tả trọn vẹn được ý tưởng đoạn phim quảng cáo.

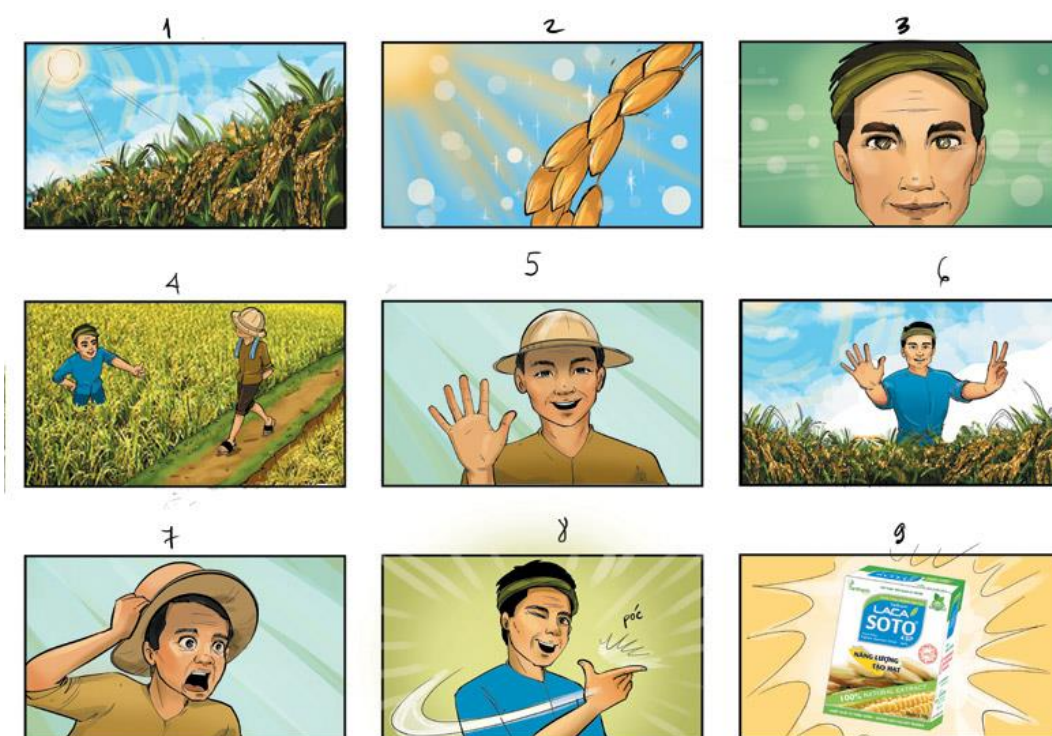
Với các thiết bị quay bỏ túi như máy ảnh số hay điện thoại, có thể dễ dàng thử góc máy trước khi tiến hành quay.

3. Sắp xếp khung hình:

Sau khi đã chụp và chọn lọc những khung hình phù hợp, ta sử dụng phần mềm vi tính để ghép chúng lại thành một storyboard hoàn hảo

Những ai giỏi phần mềm đồ họa như Photoshop thậm chí có thể ghép mặt diễn viên vào khung hình để dễ hình dung.

Thậm chí, nếu ta giỏi về vẽ, có thể tự vẽ phác thảo trên 1 tờ giấy hoặc poster để tiến hành hoàn thành Storyboard.





Hình 5.13: Minh họa Storyboard trong After Effect

Những lưu ý khi làm Storyboard:

Cần có sự cân bằng tỉ lệ của khung Storyboard và màn hình.

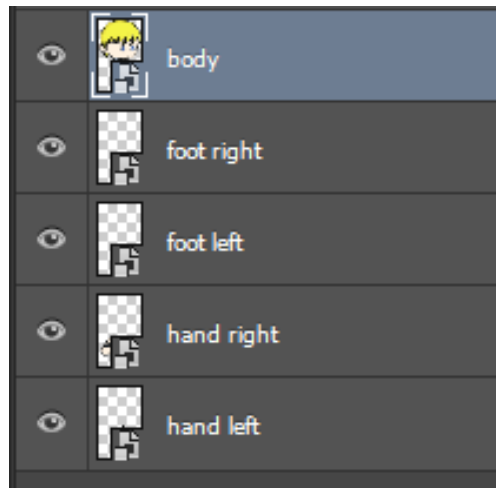
Tạo khung mô tả bên dưới khung Storyboard để đảm mỗi khung Storyboard chỉ có hình ảnh, nhưng người xem Storyboard vẫn có thể nắm được những thông tin quan trọng cần sự chính xác cao độ.

Có thể sử dụng phần mềm vẽ storyboard làm dữ liệu cho những thông tin trong kịch bản, các đạo cụ cần thiết, địa điểm, hướng máy và cuối cùng là danh sách các góc quay và cảnh quay.

5.5. Tạo hình đồ họa nhân vật: gắn xương, các chuyển động cơ bản

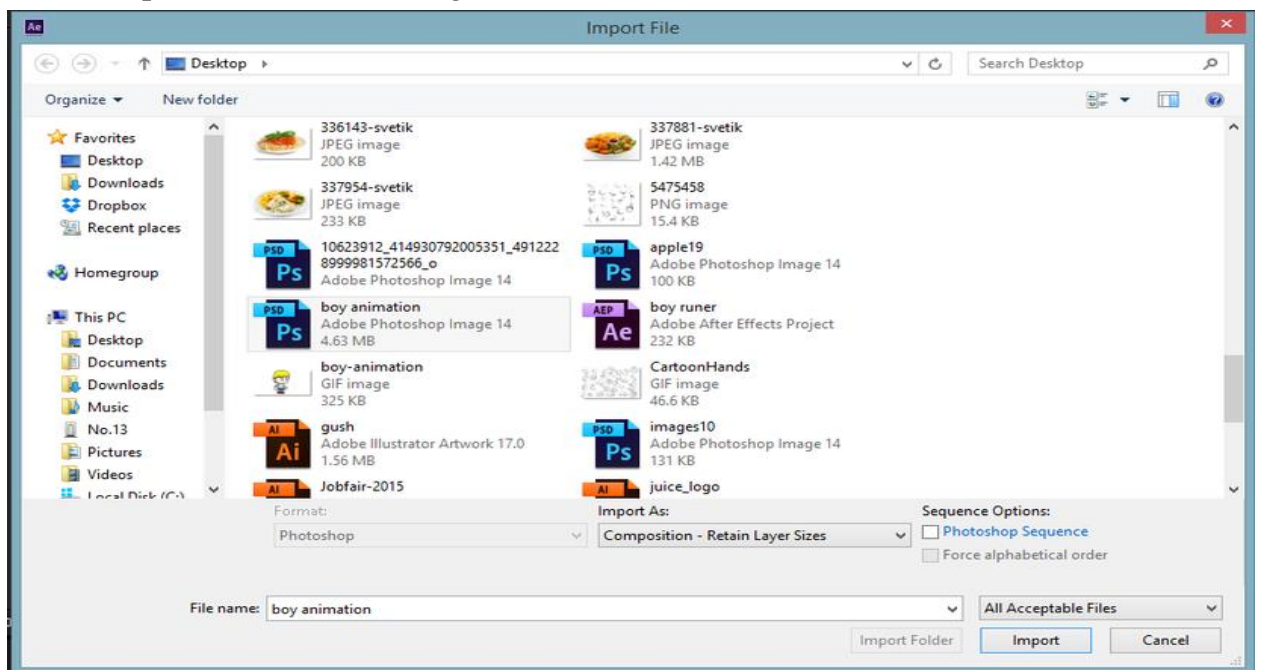
Tạo chuyển động nhân vật bằng After Effect

Chuẩn bị file PSD nhân vật (cần tách rời các yếu tố của nhân vật như body, hand, foot)



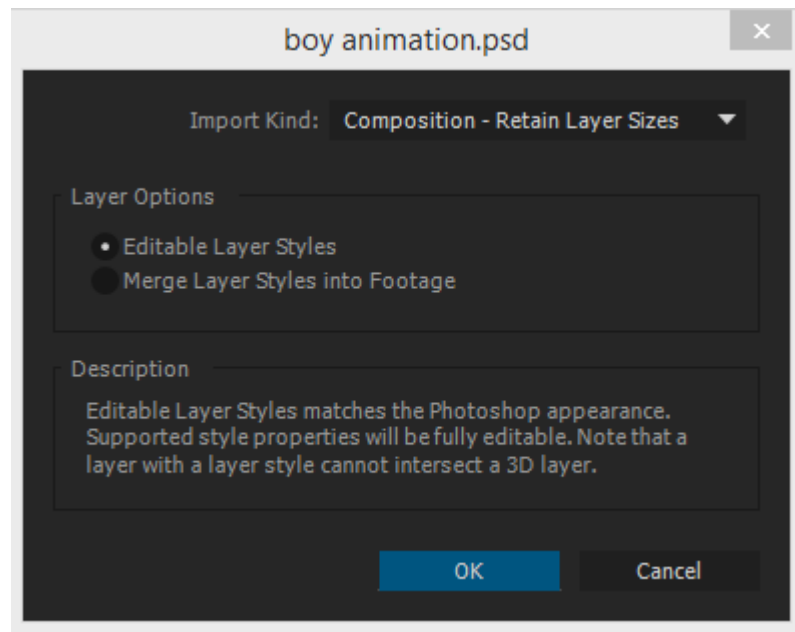
Hình 5.14: File PSD nhân vật

Import file PSD vào trong After Effect



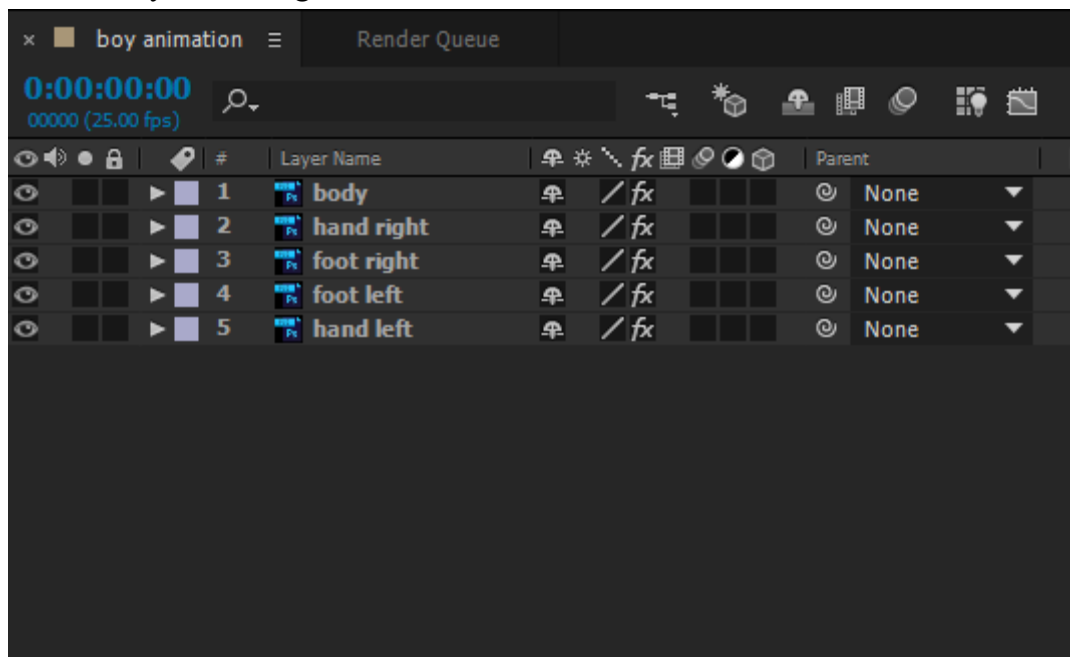
Hình 5.15: Import File PSD nhân vật

Chọn "Composition - Retain Layer Sizes"



Hình 5.16: Mô tả Import File PSD nhân vật

Chọn "Editable Layer Styles" Như vậy file được Import vào After Effect sẽ được chia thành các layer dễ dàng chỉnh sửa.



Hình 5.17: Layer nhân vật

Sử dụng công cụ "Puppet Pin Tool" để tạo các điểm neo cho các phần của nhân vật (Body, hand right, hand left, foot right, foot left)



Hình 5.18: Hiệu chỉnh các điểm neo nhân vật

Tạo các key frame theo chuyển động đã vẽ.

ở đây để tạo ra một chuyển động cần có hình ảnh mô tả chuyển động đó. Ví dụ:



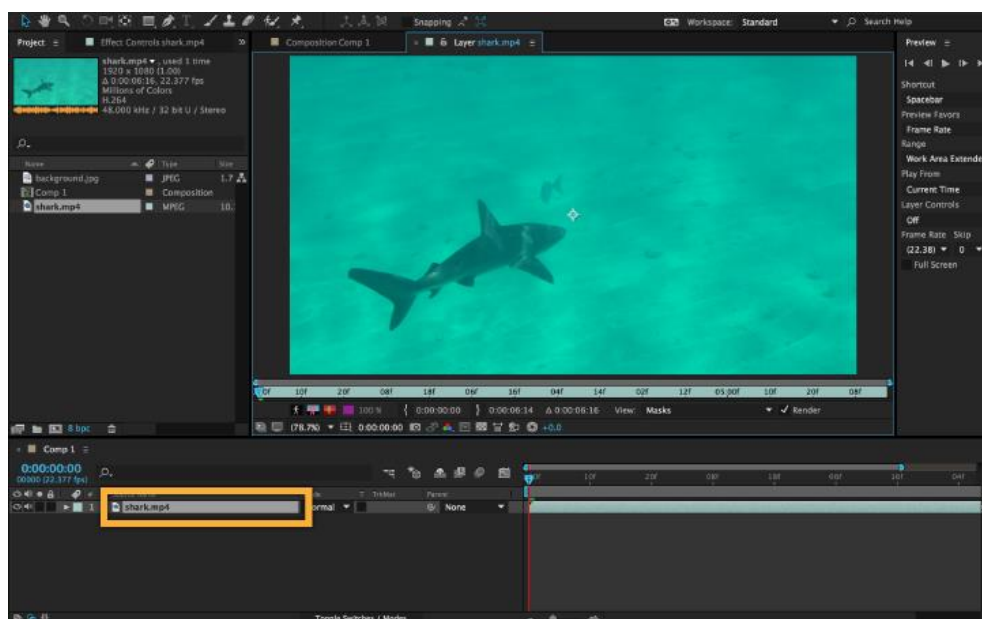
Hình 5.19: Tạo mô tả chuyển động nhân vật

5.6. Kỹ thuật tạo bối cảnh đồ họa

- **Mở file thiết kế cần ứng dụng kĩ xảo**

Sử dụng tài nguyên rotoscoping để thực hành theo các bước trong tutorial. Nhấn **Save to Creative Cloud** để sao chép tài nguyên vào tài khoản riêng.

Mở file thiết kế **rotoscoping.aep** trong **Adobe After Effect**. Click chuột hai lần vào layer có tên **shark.mp4** trong mục **Timeline**. Thao tác này sẽ mở chúng trong cửa **Layer** riêng.

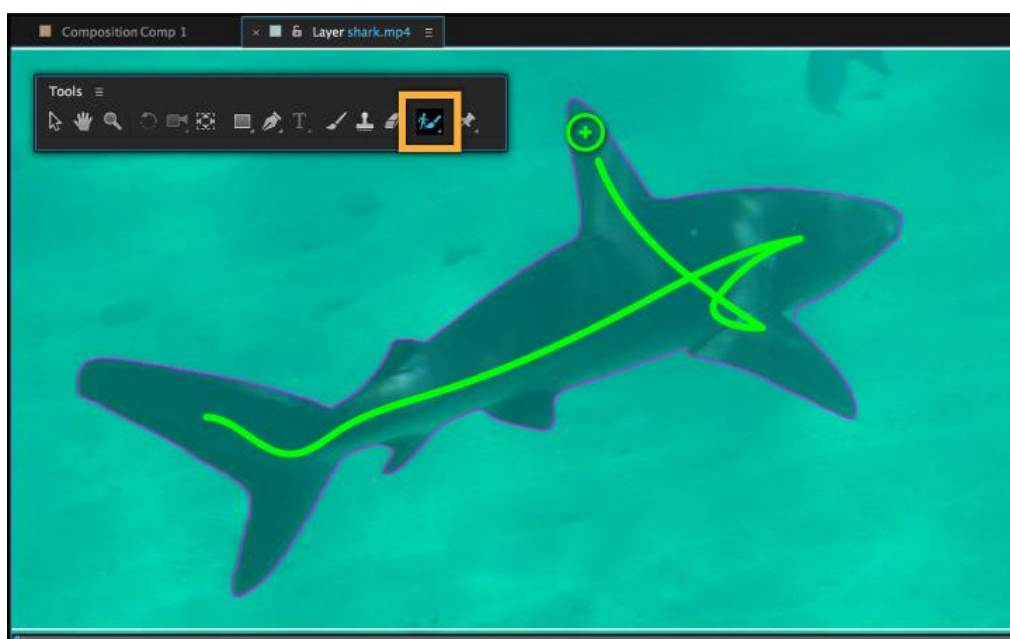


Hình 5.20: Bối cảnh trong After Effect

- **Sử dụng công cụ Roto Brush để thực hiện kỹ xảo**

Để phân tách hình ảnh con cá mập khỏi background, bạn cần xác định đường viền. Ở đây là đường viền hình cá mập trong khung hình đầu tiên của cảnh phim. Nhấn chọn công cụ **Roto Brush**.

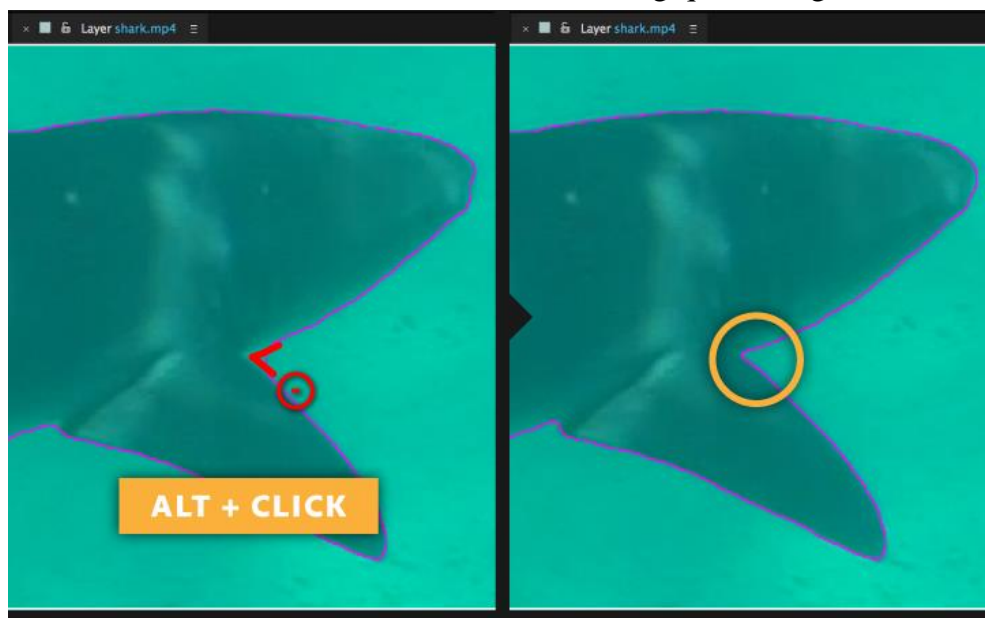
Click chọn công cụ **Roto Brush**. Xác định phần cảnh của khung hình bạn muốn giữ lại. Thay đổi mức phóng to để cả thân cá mập vừa với khung hình. Sử dụng nét vẽ để tô màu phân trong của cá mập. Đường viền chia cách màu đỏ sẫm sẽ xuất hiện dọc theo thân cá mập. Điều này cho thấy phần hình phía trước đã được tách khỏi nền.



Hình 5.21a: Công cụ Roto Brush

Sau khi xác định thân cá mập, hãy sử dụng công cụ **Roto Brush**. Chúng sẽ giúp bạn hoàn chỉnh các cạnh chưa hoàn hảo. Nhấn phím **Alt/Option** trên bàn phím. Chú ý rằng công cụ **Brush** sẽ chuyển đổi từ vòng tròn xanh thành đỏ.

Giữ **Alt/Option** trong lúc vẽ để xác định phần background muốn xóa. Hãy để ý kỹ phần ảnh xung quanh vây cá mập, chúng có các nét chưa hoàn hảo. Việc dành thời gian nhiều nhất có thể để hoàn chỉnh bước đầu tiên là vô cùng quan trọng.[6]



Hình 5.21b: Công cụ Roto Brush

Câu hỏi ôn tập chương:

1. Trình bày các thao tác về Composition
2. Nguyên tắc và phương pháp sử dụng KeyFrame.
3. Trình bày các phím tắt After Effects để tăng hiệu quả công việc.

Bài tập ôn tập:

1. Thực hành các dạng bài tập thiết kế hiệu ứng, kỹ xảo cơ bản và làm chủ giao diện phần mềm After Effects
2. Thực hành các dạng bài tập về kỹ thuật Track Motion, Track Camera và biểu thức lệnh với Expression trong After Effects
3. Thực hành các dạng bài tập kỹ xảo tách phông nền với Keylight trong After Effects và bài tập thiết kế hiệu ứng, kỹ xảo, Vfx sử dụng Plugin Keying Suite
4. Thực hành các dạng bài tập thiết kế hiệu ứng, kỹ xảo chuyên nghiệp sử dụng Mocha Tracking trong After Effects.
5. Thực hành các dạng bài tập về kỹ thuật ứng dụng Camera, Plugin Element 3D, Cinema 4D và Optical Flare trong After Effects.
6. Thực hành: Import files
Import một files *.PSD, chọn import kind, chọn layer, chọn merge layer, chọn footage dimension...
Import files *.TGA (24 bit)
Import files *.TGA (32 bit)
Import chuỗi hình *.TGA sequence.

Select từng chức năng của AE, kéo từng footage thả vào Composition panel và quan sát.

TÀI LIỆU THAM KHẢO

- [1] Trịnh Thị Vân Anh, *Giáo trình kỹ thuật đồ họa*, Nhà xuất bản thông tin và truyền thông 2010.
- [2] Dương Anh Đức, Lê Đình Duy, Hoàn Kiếm, *Giáo trình đồ họa máy tính*, Nhà xuất bản ĐHQG – HCM 2010.
- [3] Bùi Thế Duy, *Đồ họa máy tính*, NXB Đại học Quốc Gia HN 2009.
- [4] Lê Tấn Hùng, Huỳnh Quyết Thắng, *Kỹ thuật đồ họa máy tính*, NXB Khoa học và kỹ thuật, 2002
- [5] TS. Nguyễn Hữu Tài, *Giáo trình đồ họa máy tính*, Nhà xuất bản Đại học Huế, 2017
- [6] Jerrson Smith và nhóm AGI Creative, *Xử lý kỹ xảo với After Effects CS6 Digital Classroom*, Nhà xuất bản Bách khoa hà Nội, 2016.
- [7] David Salomon, *The Computer Graphics Manual*, Springer 2011.
- [8] Peter Shirley and Steve Marschner, *Fundamentals of Computer Graphics- Third Edition*, AK Peters 2011.