

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-----o0o-----



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG - 2011

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

TÌM HIỂU KỸ THUẬT 3D MORPHING

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công Nghệ Thông Tin

Sinh viên thực hiện: Nguyễn Quang Tuấn
Giáo viên hướng dẫn: PGS.TS Đỗ Năng Toàn
Mã số sinh viên: 090136

HẢI PHÒNG - 2011

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do – Hạnh phúc

-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: NGUYỄN QUANG TUẤN

Mã sinh viên: 090136

Lớp: CT902

Ngành: Công Nghệ Thông Tin

Tên đề tài: Tìm hiểu kỹ thuật 3D Morphing

NHIỆM VỤ CỦA ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung

 Tìm hiểu kỹ thuật 3D Morphing

b. Các yêu cầu cần giải quyết

- Tìm hiểu khái quát về xử lý ảnh và nội suy ảnh
- Tìm hiểu kỹ thuật nội suy lưới 3D Morphing
- Cài đặt thử nghiệm

2. Các số liệu cần thiết để thiết kế, tính toán

3. Địa điểm thực tập

Viện Khoa Học và Công Nghệ Việt Nam – Viện Công Nghệ Thông Tin

Địa chỉ: 18 Đường Hoàng Quốc Việt – Quận Cầu Giấy – Hà Nội

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn:

Họ và tên: PGS.TS Đỗ Năng Toàn

Học hàm và học vị: PGS.TS

Cơ quan công tác: Viện Khoa Học và Công Nghệ Việt Nam – Viện Công Nghệ thông Tin

Nội dung hướng dẫn: Tìm hiểu kỹ thuật 3D Morphing

Đề tài tốt nghiệp được giao ngày 11 tháng 4 năm 2011

Yêu cầu phải hoàn thành trước ngày 18 tháng 7 năm 2011

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N

Cán bộ hướng dẫn: Đ.T.T.N

Hải Phòng, Ngày tháng năm 2011

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

3. Cho điểm của cán bộ hướng dẫn:
(Điểm ghi bằng số và chữ)

Ngày..... tháng.....năm 2011

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế.....)

2. Cho điểm của cán bộ phản biện
(Điểm ghi bằng số và chữ)

Ngày.....tháng.....năm 2011

Cán bộ chấm phản biện

(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Em xin được bày tỏ lòng biết ơn sâu sắc tới thầy giáo, PGS.TS Đỗ Năng Toàn, người đã trực tiếp hướng dẫn, tận tình chỉ bảo em trong suốt quá trình làm đề án của mình.

Em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công nghệ thông tin - Trường ĐHDL Hải Phòng, những người đã nhiệt tình giảng dạy và truyền đạt những kiến thức cần thiết trong suốt thời gian em học tập tại trường.

Cuối cùng em xin cảm ơn tất cả các bạn đã góp ý, trao đổi hỗ trợ cho em trong suốt thời gian vừa qua.

Em chân thành cảm ơn!

Hải Phòng, tháng 07 năm 2011

MỤC LỤC

PHẦN MỞ ĐẦU	1
CHƯƠNG 1: KHÁI QUÁT VỀ XỬ LÝ ẢNH VÀ NỘI SUY ẢNH ...	3
1.1. Khái quát về xử lý ảnh.....	3
1.1.1. Những vấn đề cơ bản trong hệ thống xử lý ảnh	4
1.1.1.1. Điểm ảnh	4
1.1.1.2. Nắn chỉnh biến dạng.....	5
1.1.1.3. Mức xám của ảnh	5
1.1.1.4. Các kiểu ảnh	6
1.1.1.5. Một số thuộc tính cần quan tâm của ảnh số	8
1.1.1.6. Tọa độ ảnh	9
1.1.1.7. Trích đoạn đặc điểm	10
1.1.2. Nhận dạng	10
1.1.3. Nén ảnh	11
1.2. Nội suy ảnh	12
1.2.1. Khái niệm nội suy ảnh	12
1.2.2. Các vấn đề với nội suy ảnh.....	14
1.2.3. Nội suy ảnh 3 chiều (3D Morphing).....	15

CHƯƠNG 2: KỸ THUẬT 3D MORPHING LƯỚI.....	19
2.1. Morphing lưới đa diện (Polygonal mesh)	19
2.2. Rút trích các vector biến đổi.....	19
2.3. Nạp mô hình nguồn và biến đổi mô hình mục tiêu.....	21
CHƯƠNG 3: CHƯƠNG TRÌNH THỬ NGHIỆM	24
3.1. Giao diện của chương trình	24
3.2. Kết quả của chương trình	26
PHẦN KẾT LUẬN.....	28
TÀI LIỆU THAM KHẢO	29

DANH MỤC HÌNH VẼ

Hình 1.1: Quá trình xử lý ảnh.....	
Hình 1.2: Các bước cơ bản trong một hệ thống xử lý ảnh.....	
Hình 1.3: Ảnh thu nhận và ảnh mong muốn	
Hình 3.1: Giao diện chính của chương trình	
Hình 3.2: Giao diện tải mô hình 3D nguồn và mô hình 3D đích	
Hình 3.3: Mô hình nguồn được tải lên	
Hình 3.4: Mô hình nguồn khi chưa Morphing	
Hình 3.5: Mô hình nguồn sau khi được Morphing.....	
Hình 3.6: Mô hình đích sau khi kết thúc quá trình Morphing	

DANH MỤC BẢNG BIỂU

Bảng 1: Thuật toán rút trích vector biến đổi

Bảng 2: Thuật toán biến đổi mô hình

PHẦN MỞ ĐẦU

Thế kỷ XXI là thế kỷ đánh dấu bước phát triển nhảy vọt về công nghệ khoa học và kỹ thuật đặc biệt là lĩnh vực công nghệ thông tin, công nghệ thông tin đóng vai trò rất quan trọng trong đời sống của chúng ta, nó đã đạt được những thành tựu to lớn. Việc áp dụng các công nghệ khoa học kỹ thuật vào lĩnh vực đời sống của con người ngày càng tăng, máy tính điện tử không còn là phương tiện quý hiếm mà đang ngày một gần gũi với con người, đồng thời phần cứng máy tính và các thiết bị liên quan đã có sự tiến bộ vượt bậc về tốc độ tính toán, dung lượng chứa, khả năng xử lý và giá cả giảm đến mức máy tính và các thiết bị liên quan đến thực tại ảo không còn là thiết bị chuyên dụng nữa.

Cùng với sự phát triển mạnh mẽ của thực tại ảo, đã đưa con người tới những nhu cầu cao về các lĩnh vực giải trí, thiết kế kiến trúc và thiết bị công nghệ, trong việc đào tạo khoa học giáo dục, và cả trong lĩnh vực y tế. Ở đó con người được tận dụng hết những thế mạnh, cũng như những ứng dụng vô cùng to lớn của nó mang lại, đó là khả năng mô hình hóa cụ thể các đối tượng, mô tả được một cách trực quan nhất, để người dùng có thể có những đánh giá khách quan nhất về ý tưởng cũng như có thể dễ dàng thay đổi những ý tưởng của mình, đồng thời cũng tạo cho con người khả năng cảm nhận được sự hiện diện của mình, khả năng tự trị, và những phản hồi rất khách quan từ phía môi trường tới các giác quan của người sử dụng. Ngày nay cuốn theo những nhu cầu về giải trí, nghiên cứu khoa học, con người càng có nhiều ý tưởng sáng tạo dựa trên cái nền phát triển đã có của thực tại ảo, con người đã biết dựa vào đó để phát triển nên những trò chơi thực tại ảo thu hút một số lượng lớn người quan tâm, cũng như sử dụng nó để phát triển nên những bộ phim mang tính lịch sử, mang lại giá trị kinh tế vô cùng lớn. Trong những ứng dụng vô cùng mạnh mẽ đó của thực tại ảo, thì một kỹ thuật cũng mang lại những thành công nhất định trong các lĩnh vực như giải trí, điện ảnh, các trò chơi giải trí, hoặc các bộ phim hoạt hình ăn khách và cũng được ứng dụng trong tất cả các lĩnh vực khác nữa, như làm quảng cáo, nghiên cứu khoa học, tìm hiểu về sự tiến

hóa ..vv. Đó chính là kỹ thuật 3D Morphing. 3D Morping là kỹ thuật xây dựng lại một chuỗi các đối tượng ba chiều được biến hình từ đối tượng nguồn tới đối tượng mục tiêu.

Xuất phát trong hoàn cảnh đó, em lựa chọn đề tài đồ án “Tìm hiểu kỹ thuật 3D Morphing”, đồ án gồm Phần mở đầu, Phần kết luận và ba chương nội dung, cụ thể:

Chương 1: Khái quát về xử lý ảnh và nội suy ảnh

Chương này trình bày khái quát về xử lý ảnh, một số vấn đề cơ bản của xử lý ảnh, đề cập đến nội suy ảnh, một vấn đề quan trọng trong xử lý ảnh.

Chương 2: Kỹ thuật nội suy 3D Morphing lưới

Phần này trình bày kỹ thuật Morphing 3D dựa trên lưới đã giác và một số vấn đề liên quan.

Chương 3: Chương trình thử nghiệm

Phần này trình bày chương trình cài đặt thử nghiệm kỹ thuật 3D Morphing mà em tìm hiểu được trên nền tảng ngôn ngữ lập trình C++

Chương 1:

KHÁI QUÁT VỀ XỬ LÝ ẢNH VÀ NỘI SUY ẢNH

1.1. Khái quát về xử lý ảnh

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học còn tương đối mới mẻ so với nhiều ngành khoa học khác, nhưng tốc độ phát triển của nó rất nhanh, nhất là trên qui mô công nghiệp, điều này đã kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng cho nó. Xử lý ảnh có quan hệ mật thiết với nhận thức về ảnh của con người. Nói một cách khác, "thị giác máy" dựa trên phép xử lý ảnh bằng sự phân tích của máy, có thể nói "xử lý ảnh số và thị giác máy" được liên kết chặt chẽ với nhau.

Trong các dạng truyền thông cơ bản: lời nói, văn bản, hình ảnh, âm thanh thì hình ảnh là dạng truyền thông truyền tải thông tin mạnh mẽ nhất. Bằng thị giác, con người có thể nhận biết và hiểu về thế giới xung quanh chúng ta. Ví dụ: Những hình ảnh về trái đất, những hình ảnh trong dự báo thời tiết...

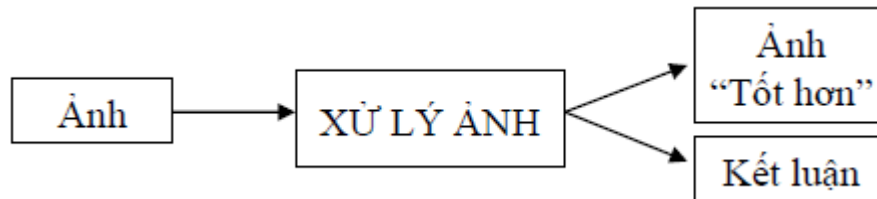
Việc trang bị cho máy tính có khả năng thị giác như con người không phải là việc dễ dàng. Chúng ta đang sống trong một không gian 3D, khi máy tính cố gắng phân tích đối tượng trong không gian 3D thì những bộ cảm biến có sẵn như camera,.. lại thường cho ảnh 2D. Như vậy, việc mất mát thông tin của hình ảnh sẽ xảy ra. Với những cảnh động thì sự di chuyển của đối tượng hay sự di chuyển của camera, tất cả những việc đó làm cho việc mất mát và sai lệch thông tin rất lớn.

Con người thu nhận thông tin qua các giác quan, trong đó thị giác đóng vai trò quan trọng nhất. Những năm trở lại đây với sự phát triển của phần cứng máy tính, xử lý ảnh và đồ họa đó phát triển một cách mạnh mẽ và có nhiều ứng dụng trong cuộc sống. Xử lý ảnh và đồ họa đóng một vai trò quan trọng trong tương tác người máy.

Việc xử lý ảnh trên máy tính là nhằm mục đích phân tích ảnh và phục hồi các thông tin bị sai lệch của ảnh trong quá trình chụp. Như vậy xử lý ảnh số là thực hiện các phép xử lý đối với ảnh số trên máy tính. Máy tính sử dụng các phần mềm xử lý

ảnh để phân tích, biến đổi ảnh nhằm làm cho ảnh đẹp hơn. Hầu như tất cả các phần mềm chỉnh sửa ảnh đều sử dụng 1 hoặc nhiều phương pháp nội suy. Hình ảnh sẽ mịn màng, không bị "vỡ hạt" khi phóng to tùy vào thuật toán được sử dụng trong giải thuật nội suy. Điều quan trọng cần ghi nhớ là giải thuật nội suy sẽ không thêm thông tin gì mới cho hình ảnh cả, nó chỉ thêm điểm ảnh và làm tăng dung lượng của tập tin mà thôi. Tuy nhiên nhờ những phần mềm xử lý này mà ảnh có thể được phóng to, thu nhỏ hay biến đổi tùy ý mà ảnh vẫn đẹp. Những biến đổi này đẹp hay xấu tùy theo mục đích của người sử dụng, nhưng muốn ảnh biến đổi theo đúng mục đích của mình thì điều quan trọng là người dùng cần phải hiểu ảnh.

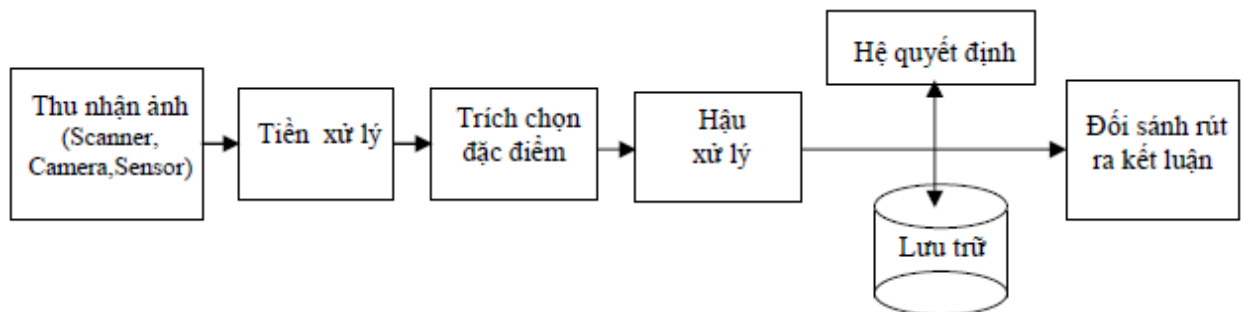
Quá trình xử lý ảnh được xem như là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn. Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh "tốt hơn" hoặc một kết luận.



Hình 1.1: Quá trình xử lý hình ảnh

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến $P(c_1, c_2, \dots, c_n)$. Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

Sơ đồ tổng quát của một hệ thống xử lý ảnh:



Hình 1.2: Các bước cơ bản trong một hệ thống xử lý ảnh

1.1. Những vấn đề cơ bản trong hệ thống xử lý ảnh

1.1.1.1. Điểm ảnh

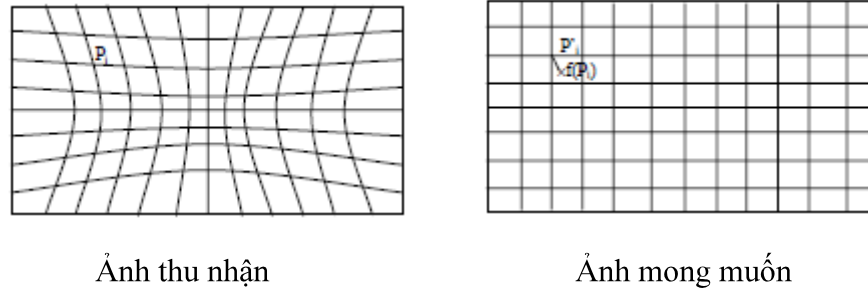
Ảnh trong thực tế (ảnh tự nhiên) là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính, cần thiết phải tiến hành số hoá ảnh nhằm biến đổi gần đúng một ảnh liên tục thành một tập điểm, phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức xám). Trong quá trình số hoá, người ta biến đổi tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hoá về không gian), và lượng hoá thành phân giá trị mà về nguyên tắc bằng mắt thường không phân biệt được 2 điểm kề nhau. Trong quá trình này người ta sử dụng khái niệm điểm ảnh (pixel). Như vậy một ảnh là một tập hợp các điểm ảnh. Khi sử dụng đến nội suy thì việc phân biệt hai điểm ảnh kề nhau là việc cần thiết.

Điểm ảnh (pixel) được xem như là dấu hiệu hay cường độ sáng tại một toạ độ trong không gian của đối tượng. Mỗi pixel gồm một cặp toạ độ x, y và màu.

Như vậy, một ảnh là một tập hợp các điểm ảnh. Khi được số hoá, nó thường được biểu diễn bởi mảng hai chiều hay ma trận hai chiều $I(n,p)$: mỗi phần tử có một giá trị nguyên hoặc là một véc tơ cấu trúc màu, n dòng và p cột. Ta nói ảnh gồm $n \times p$ pixels. Người ta thường kí hiệu $I(x,y)$ để chỉ một pixel. Thường giá trị của n chọn bằng p và bằng 256. Một pixel có thể lưu trữ trên 1, 4, 8 hay 24 bit. Mỗi điểm ảnh khi mã hoá sẽ được biểu diễn dưới dạng 8 bit. Cách mã hoá kinh điển thường dùng 16, 32 hay 64 mức. Mã hoá 256 mức là phổ dụng nhất do lý do kỹ thuật. Vì $2^8 = 256$ (0, 1, ..., 255), nên với 256 mức, mỗi pixel sẽ được mã hoá bởi 8 bit và từ đó có thể biểu diễn ảnh dưới nhiều dạng khác nhau. Số pixel tạo nên một ảnh gọi là độ phân giải (resolution).

1.1.1.2. Nắn chỉnh biến dạng

Ảnh thu nhận thường bị biến dạng do các thiết bị quang học và điện tử. Để khắc phục người ta sử dụng các phép chiếu, các phép chiếu thường được xây dựng trên tập các điểm điều khiển



Hình 1.3: Ảnh thu nhận và ảnh mong muốn

1.1.1.3. Mức xám của ảnh

Mức xám của điểm ảnh là cường độ sáng của nó được gán bằng giá trị số tại điểm đó. Trong biểu diễn số của các ảnh đa mức xám, một ảnh được biểu diễn dưới dạng một ma trận hai chiều. Mỗi phần tử của ma trận biểu diễn cho mức xám hay cường độ của ảnh tại vị trí đó. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh hoặc điểm ảnh (pixel). Một điểm ảnh có hai đặc trưng cơ bản là vị trí (x,y) của điểm ảnh và độ xám.

a. Các thang giá trị mức xám thông thường

Thông thường có các thang mức xám như : 16, 32, 64, 128, 256 (với lý do kỹ thuật máy tính dùng 1 byte (8 bit) để biểu diễn mức xám thì có thể biểu diễn: $2^8 = 256$ mức (0.....255) thì mức 256 là mức phổ dụng.

b. Mức xám ở ảnh đen trắng

Ảnh đen trắng là ảnh chỉ có hai màu đen trắng, mức xám ở các điểm ảnh có thể khác nhau. Nếu dùng 8 bit (1 byte) để biểu diễn mức xám, thì số các mức xám có thể biểu diễn được là 2^8 hay 256. Mỗi mức xám được biểu diễn dưới dạng là một số nguyên nằm trong khoảng từ 0 đến 255, với mức 0 biểu diễn cho mức cường độ đen nhất và 255 biểu diễn cho mức cường độ sáng nhất.

c. Mức xám ở ảnh màu nhị phân

Ảnh chỉ có hai mức đen, trắng phân biệt, tức dùng 1 bit mô tả 2^1 mức khác nhau. Nói cách khác mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1.

d. *Mức xám ở ảnh màu*

Ảnh màu được tạo nên từ ba màu cơ bản (Red, Blue, Green), người ta dùng 3 byte để mô tả mức màu, khi đó giá trị màu: $2^{8*3} = 2^{24} = 16,7$ triệu màu.

Với ảnh màu: Cách biểu diễn cũng tương tự như với ảnh đen trắng, chỉ khác là số tại mỗi phần tử của ma trận biểu diễn cho ba màu riêng rẽ gồm: đỏ (red), lục (green) và lam (blue). Để biểu diễn cho một điểm ảnh màu cần 24 bit, 24 bit này được chia thành ba khoảng 8 bit. Mỗi khoảng này biểu diễn cho cường độ sáng của một trong các màu chính.

1.1.1.4. Các kiểu ảnh

a. *Ảnh chỉ số (Index Images)*

Một ảnh chỉ số bao gồm một ma trận dữ liệu X và ma trận bản đồ màu (map). Ma trận dữ liệu có thể có kiểu thuộc lớp uint8, uint16 hoặc kiểu double. Ma trận bản đồ màu là một mảng $m \times 3$ kiểu double bao gồm các giá trị dấu phẩy động nằm giữa 0 và 1. Mỗi hàng của bản đồ chỉ ra các giá trị màu: red, green và blue của một màu đơn. Một ảnh chỉ số sử dụng ánh xạ trực tiếp giữa giá trị của pixel ảnh tới giá trị trong bản đồ màu. Màu sắc của mỗi pixel ảnh được tính toán bằng cách sử dụng giá trị tương ứng của X ánh xạ tới một giá trị chỉ số của bản đồ màu. Giá trị 1 chỉ ra hàng đầu tiên, giá trị 2 chỉ ra hàng thứ hai trong bản đồ màu.

Một bản đồ màu thường được chứa cùng với ảnh chỉ số và được tự động nạp cùng với ảnh. Tuy nhiên, ta không bị giới hạn khi sử dụng bản đồ màu mặc định, ta có thể sử dụng bất kì bản đồ màu nào. Các pixel trong ảnh được đại diện bởi một số nguyên ánh xạ tới một giá trị tương ứng trong bản đồ màu.

b. *Ảnh cường độ (Intensity Images)*

Một ảnh cường độ là một ma trận dữ liệu ảnh I mà giá trị của nó đại diện cho cường độ trong một số vùng nào đó của ảnh. Ma trận có thể thuộc lớp double, uint8 hay uint16. Trong khi ảnh cường độ hiếm khi được lưu với bản đồ màu. Những phần tử trong ma trận cường độ đại diện cho các cường độ khác nhau hoặc độ xám.

c. *Ảnh nhị phân (Binary Images)*

Trong một ảnh nhị phân, mỗi pixel chỉ có thể chứa một trong hai giá trị nhị phân 0 hoặc 1. Hai giá trị này tương ứng với bật hoặc tắt (on hoặc off). Một ảnh nhị phân được lưu trữ như một mảng logic của 0 và 1.

d. *Ảnh RGB (RGB Images)*

Một ảnh RGB được lưu trữ dưới dạng một mảng dữ liệu có kích thước 3 chiều $m \times n \times 3$, định nghĩa các giá trị màu red, green và blue cho mỗi pixel riêng biệt. Ảnh RGB không sử dụng bảng màu. Màu của mỗi pixel được quyết định bởi sự kết hợp giữa các giá trị R,G,B (Red, Green, Blue) được lưu trữ trong một mặt phẳng màu tại vị trí của pixel. Định dạng file đồ họa lưu trữ ảnh RGB giống như một ảnh 24 bit trong đó R,G,B chiếm tương ứng 8 bit 1. Điều này cho phép nhận được 16,7 triệu màu khác nhau.

Một mảng RGB có thể thuộc lớp double, uint8 hoặc uint16. Trong một mảng RGB thuộc lớp double, mỗi thành phần màu có giá trị giữa 0 và 1. Một pixel mà thành phần màu của nó là (0,0,0) được hiển thị với màu đen và một pixel mà thành phần màu là (1,1,1) được hiển thị với màu trắng.

Trong một ảnh RGB khoảng trắng tương ứng với giá trị cao nhất của mỗi màu riêng rẽ. Chẳng hạn trong ảnh mặt phẳng R, vùng trắng đại diện cho sự tập trung cao nhất của màu đỏ thuần khiết. Nếu R được trộn với G hoặc B ta sẽ có màu xám. Vùng màu đen trong ảnh chỉ ra giá trị của pixel mà không chứa màu đỏ $R=0$. Tương tự cho các mặt phẳng màu G và B.

1.1.1.5. Một số thuộc tính cần quan tâm của ảnh số

a. *Đường viền (Border):*

Đường viền của một vùng ảnh R là tập hợp các điểm ảnh trong vùng đó mà có một hay nhiều lân cận bên ngoài vùng R.

b. *Biên ảnh (Edge):*

Một điểm ảnh có thể coi là biên nếu ở đó có sự thay đổi đột ngột về mức xám. Tập hợp các điểm biên tạo thành đường bao của ảnh.

Thuộc tính biên gắn liền một điểm ảnh và lân cận của nó, đôi khi nó giúp cho việc xác định đặc tính giữa một cặp điểm lân cận

Ví dụ: Trong một ảnh nhị phân, một điểm có thể gọi là biên nếu đó là điểm đen và có ít nhất một điểm trắng lân cận.

c. Độ sắc nét của ảnh:

Độ sắc nét là khả năng phát hiện những chi tiết trong ảnh. Mắt người ít nhạy cảm với sự thay đổi nhanh hay chậm của độ sáng trong mặt phẳng ảnh nhưng nhạy cảm với sự thay đổi trung gian.

Độ phân giải trong ảnh được giới hạn bởi khả năng phân giải ở mắt người. Khi độ phân giải của ảnh cao hơn độ phân giải của mắt người thì con người không thể cảm nhận về ảnh được nữa.

Độ phân giải trong quang học được định nghĩa là khoảng cách giữa 2 điểm ảnh gần nhất mà con người không thể phân biệt được.

Mỗi một ảnh đều có độ sắc nét, độ phân giải riêng, việc xử lý ảnh nhằm mục đích làm cho ảnh sắc nét hơn, đẹp hơn hay gần với ảnh gốc hơn, và khi biến đổi bằng một trong các phương pháp nội suy là làm cho ảnh có khả năng zoom tốt, mà vẫn đảm bảo độ sắc nét, tránh được hiện tượng nhiễu hay răng cưa.

Để sử dụng một trong các phương pháp nội suy trong xử lý ảnh làm cho ảnh tốt hơn phải trải qua quá trình tìm được điểm ảnh thích hợp để chèn điểm ảnh mới vào, việc tìm kiếm này người ta dựa vào tọa độ của điểm ảnh, hay tọa độ pixel.

1.1.1.6. Tọa độ ảnh

a. Tọa độ pixel

Nhìn chung, phương pháp thuận tiện nhất cho việc biểu diễn vị trí trong một ảnh là sử dụng tọa độ pixel. Trong hệ tọa độ này, ảnh được xử lý như một lưới của các phần tử riêng biệt được đánh thứ tự từ đỉnh tới đáy và từ trái sang phải.

Với tọa độ pixel, thành phần đầu tiên r (hàng) được tăng khi đi từ trên xuống dưới trong khi c (cột) được tăng khi đi từ trái sang phải. Hệ tọa độ pixel là giá trị nguyên, có giá trị nằm trong khoảng giữa 1 và chiều dài của hàng hay cột.

b. Tọa độ không gian

Trong tọa độ không gian, vị trí trong một ảnh được định vị trên một mặt phẳng và chúng được mô tả bằng một cặp x và y (không phải r (hàng) và c (cột) như tọa độ pixel).

Hệ tọa độ không gian gần tương ứng với hệ tọa độ pixel trong một chừng mực nào đó. Chẳng hạn, tọa độ không gian của điểm giữa của bất kỳ pixel nào được phân biệt với tọa độ pixel của pixel đó. Cũng có một vài khác biệt, tuy nhiên, trong tọa độ pixel, góc trên trái của một ảnh là $(1,1)$ trong khi trong tọa độ không gian, vị trí này mặc định là $(0.5,0.5)$. Sự khác nhau này là do hệ tọa độ pixel là rời rạc trong khi tọa độ không gian là liên tục. Cũng vậy, góc trên trái luôn là $(1,1)$ trong hệ pixel, nhưng ta có thể chỉ ra một điểm gốc không chính quy cho hệ tọa độ không gian. Một sự khác biệt dễ gây nhầm lẫn nữa là quy ước: thứ tự của các thành phần nằm ngang và thẳng đứng được phục vụ cho kí hiệu của hai hệ thống. Như đã đề cập trước đây, tọa độ pixel được đại diện bởi một cặp (r,c) trong khi tọa độ không gian được biểu diễn bởi (x,y) . Khi cú pháp cho một hàm sử dụng r và c , nó tham chiếu đến hệ tọa độ pixel. Khi cú pháp sử dụng x , y nó đang ngầm định sử dụng hệ tọa độ không gian.

Khi sử dụng hệ tọa độ không gian không chính quy thì theo mặc định, tọa độ không gian của một ảnh tương ứng với tọa độ pixel. Chẳng hạn, điểm giữa của pixel tại $(5,3)$ có một tọa độ không gian là $x=3$, $y=5$ (nhớ rằng thứ tự của tọa độ bị đảo ngược).

Trong một số tình huống, ta có thể muốn sử dụng tọa độ không gian không chính quy (không mặc định). Chẳng hạn, ta có thể chỉ ra góc trên trái của một

ảnh tại điểm (19.0,7.5) thay cho (0.5,0,5). Nếu ta gọi một hàm mà trả về tọa độ cho ảnh này, tọa độ được trả lại sẽ là giá trị trong hệ tọa độ không chính quy.

1.1.1.7. Trích chọn đặc điểm

Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây:

- Đặc điểm không gian: Phân bố mức xám, phân bố mức xám, biên độ, điểm uốn...
- Đặc điểm biến đổi: Các đặc điểm loại này được trích chọn bằng việc thực hiện lọc vùng (zonal filtering). Các bộ vùng được gọi là “mặt nạ đặc điểm” (feature mask) thường là các khe hẹp với hình dạng khác nhau (chữ nhật, tam giác, cung tròn v.v...).
- Đặc điểm biên và đường biên: Đặc trưng cho đường biên của đối tượng, do vậy rất hữu ích trong việc trích chọn các thuộc tính bất biến được dùng khi nhận dạng đối tượng. Các đặc điểm này có thể được trích chọn nhờ toán tử gradient, toán tử Laplace, toán tử Laplace, toán tử “chéo không” (zero crossing) v.v...

Việc trích chọn hiệu quả các đặc điểm giúp cho việc nhận dạng các đối tượng ảnh chính xác, với tốc độ tính toán cao và dung lượng nhớ lưu trữ giảm xuống.

1.1.2. Nhận dạng

Nhận dạng tự động (automatic recognition), mô tả đối tượng, phân loại và phân nhóm các mẫu là những vấn đề quan trọng trong thị giác máy, được ứng dụng trong nhiều ngành khoa học khác nhau. Tuy nhiên, một câu hỏi đặt ra là: mẫu (pattern) là gì? Watanabe, một trong những người đi đầu trong lĩnh vực này đã định nghĩa: “Ngược lại với hỗn loạn (chaos), mẫu là một thực thể (entity), được xác định một cách ang áng (vaguely defined) và có thể gán cho nó một tên gọi nào đó”. Ví dụ mẫu có thể là ảnh của vân tay, ảnh của một vật nào đó được chụp, một chữ viết, khuôn mặt người hoặc một ký đồ tín hiệu tiếng nói. Khi biết một mẫu nào đó, để nhận dạng hoặc phân loại mẫu đó có thể:

Hoặc phân loại có mẫu (supervised classification), chẳng hạn phân tích phân biệt (discriminant analysis), trong đó mẫu đầu vào được định danh như một thành phần của một lớp đã xác định.

Hoặc phân loại không có mẫu (unsupervised classification hay clustering) trong đó các mẫu được gán vào các lớp khác nhau dựa trên một tiêu chuẩn đồng dạng nào đó. Các lớp này cho đến thời điểm phân loại vẫn chưa biết hay chưa được định danh.

Hệ thống nhận dạng tự động bao gồm ba khâu tương ứng với ba giai đoạn chủ yếu sau đây:

- 1) Thu nhận dữ liệu và tiền xử lý
- 2) Biểu diễn dữ liệu
- 3) Nhận dạng, ra quyết định

Bốn cách tiếp cận khác nhau trong lý thuyết nhận dạng:

- 1) Đối sánh mẫu dựa trên các đặc trưng được trích chọn
- 2) Phân loại thống kê
- 3) Đối sánh cấu trúc
- 4) Phân loại dựa trên mạng nơ-ron nhân tạo.

Trong các ứng dụng rõ ràng là không thể chỉ dùng có một cách tiếp cận đơn lẻ để phân loại “tối ưu” do vậy cần sử dụng cùng một lúc nhiều phương pháp và cách tiếp cận khác nhau. Do vậy, các phương thức phân loại tổ hợp hay được sử dụng khi nhận dạng và nay đã có những kết quả có triển vọng dựa trên thiết kế các hệ thống lai (hybrid system) bao gồm nhiều mô hình kết hợp.

Việc giải quyết bài toán nhận dạng trong những ứng dụng mới, nảy sinh trong cuộc sống không chỉ tạo ra những thách thức về thuật giải, mà còn đặt ra những yêu cầu về tốc độ tính toán. Đặc điểm chung của tất cả những ứng dụng đó là những đặc điểm đặc trưng cần thiết thường là nhiều, không thể do chuyên gia đề xuất, mà phải được trích chọn dựa trên các thủ tục phân tích dữ liệu.

1.1.3. Nén ảnh

Nhằm giảm thiểu không gian lưu trữ. Thường được tiến hành theo cả hai cách khuynh hướng là nén có bảo toàn và không bảo toàn thông tin. Nén không bảo

toàn thì thường có khả năng nén cao hơn nhưng khả năng phục hồi thì kém hơn. Trên cơ sở hai khuynh hướng, có 4 cách tiếp cận cơ bản trong nén ảnh:

- a. Nén ảnh thống kê: Kỹ thuật nén này dựa vào việc thống kê tần suất xuất hiện của giá trị các điểm ảnh, trên cơ sở đó mà có chiến lược mã hóa thích hợp. Một ví dụ điển hình cho kỹ thuật mã hóa này là *.TIF
- b. Nén ảnh không gian: Kỹ thuật này dựa vào vị trí không gian của các điểm ảnh để tiến hành mã hóa. Kỹ thuật lợi dụng sự giống nhau của các điểm ảnh trong các vùng gần nhau. Ví dụ cho kỹ thuật này là mã nén *.PCX
- c. Nén ảnh sử dụng phép biến đổi: Đây là kỹ thuật tiếp cận theo hướng nén không bảo toàn và do vậy, kỹ thuật thường nén hiệu quả hơn. *.JPG chính là tiếp cận theo kỹ thuật nén này.
- d. Nén ảnh Fractal: Sử dụng tính chất Fractal của các đối tượng ảnh, thể hiện sự lặp lại của các chi tiết. Kỹ thuật nén sẽ tính toán để chỉ cần lưu trữ phần gốc ảnh và quy luật sinh ra ảnh theo nguyên lý Fractal.

1.2 Nội suy ảnh

1.2.1. Khái niệm nội suy ảnh

Trong toán học giải tích số, phép nội suy là một phương pháp xây dựng mới các điểm dữ liệu trong phạm vi của một tập hợp rời rạc những điểm dữ liệu được biết.

Trong kỹ thuật và khoa học thường có một số điểm dữ liệu thu được bằng việc lấy mẫu hay thí nghiệm, và thử xây dựng một chức năng mà gần gũi phù hợp với những điểm dữ liệu đó.

Có thể nói nội suy là 1 giải thuật phần mềm dùng để thêm vào (hoặc bỏ bớt) số điểm ảnh trên ảnh kỹ thuật số. Tiến trình nội suy sẽ dựa trên màu sắc của những điểm ảnh cũ để xác định màu cho các điểm ảnh mới gần nó nhất. Một số máy ảnh số sử dụng giải thuật nội suy để tạo ra ảnh có dung lượng cao hơn khả năng thu nhận của bộ cảm biến ảnh hoặc tăng cường khả năng zoom kỹ thuật số của máy. Hầu như tất cả các phần mềm chỉnh sửa ảnh đều sử dụng 1 hoặc nhiều phương pháp nội suy. Hình ảnh sẽ mịn màng, không bị "vỡ hạt" khi phóng to hay biến đổi ảnh tùy vào thuật toán được sử dụng trong giải thuật nội suy.

Có nhiều phương pháp nội suy khác nhau, nhưng cần sử dụng phương pháp nội suy nào cho phù hợp cả về tốc độ và kinh tế. Một số phương pháp nội suy phổ biến như:

- Affine Interpolation (Nội suy tam giác)
- Nearest Neighbor Interpolation (Nội suy các pixel gần nhất)
- Bicubic Interpolation (Nội suy song khối)
- Bilinear Interpolation (Nội suy song tuyến tính)
- Trilinear Interpolation (Nội suy tam tuyến tính)
- Nội suy không gian
- Nội suy thời gian có bù chuyển động.

Ngoài ra còn nhiều phương pháp nội suy hình ảnh khác nhưng không được sử dụng phổ biến, thế nhưng điều mà ta quan tâm là giải thuật nội suy sẽ không thêm thông tin gì mới cho hình ảnh cả, nó chỉ thêm điểm ảnh và làm tăng dung lượng của tập tin.

Tuy nhiên các phương pháp nội suy làm việc theo một cách giống nhau. Trong mỗi trường hợp, để tính giá trị của một pixel đã được nội suy, chúng tìm điểm trong ảnh ra mà pixel nằm tại đó. Sau đó gán một giá trị tới các pixel ra bằng cách tính toán giá trị trung bình có trọng số của một số pixel lân cận. Trọng số dựa trên cơ sở khoảng cách tới điểm đang xét.

Trong xử lý ảnh người ta sử dụng rất nhiều đến kỹ thuật nội suy. Sau khi thu nhận ảnh người ta bắt đầu xử lý và các quá trình xử lý này đã có sử dụng đến kỹ thuật nội suy như:

- Xử lý điền đầy (Filling a region): Là quá trình tô màu một vùng nhất định bằng cách nội suy giá trị pixel từ viền của vùng.
- Thay đổi kích thước của ảnh như phóng đại ảnh, quay ảnh, bóp méo... đề có thể chỉ ra kỹ thuật nội suy cần sử dụng.
- Sinh ra hình ảnh trung gian khi thực hiện nội suy từ một khung ảnh nguồn và một khung ảnh đích.

Các hàm sử dụng tuyến tính yêu cầu một tham số chỉ ra phương pháp nội suy. Với hầu hết các hàm, phương pháp mặc định được sử dụng là nội suy các pixel gần nhất. Phương pháp này tạo ra một kết quả có thể chấp nhận được cho hầu hết các ảnh và là phương pháp duy nhất thích hợp với ảnh chỉ số. Với ảnh cường độ hay RGB thường chỉ ra kiểu song tuyến tính hoặc song khối bởi vì những phương pháp này cho kết quả tốt hơn.

Với ảnh RGB, nội suy thường được thực hiện trên mặt phẳng R,B,G một cách riêng biệt. Với ảnh nhị phân, nội suy gây ra những ảnh hưởng mà ta có thể nhận thấy được. Nếu sử dụng nội suy song tuyến tính hoặc song khối, giá trị tính toán được cho pixel trong ảnh ra sẽ không hoàn toàn là 0 hoặc 1. Ảnh hưởng trên ảnh kết quả phụ thuộc vào lớp của ảnh vào.

Việc giảm kích thước (hình học) của một ảnh có thể gây ra những ảnh hưởng nhất định lên ảnh, chẳng hạn như hiện tượng xuất hiện răng cưa tại biên của ảnh. Điều này là do thông tin luôn bị mất khi ta giảm kích thước một ảnh. Răng cưa xuất hiện như những gợn sóng trong ảnh sau cùng .

Vì vậy hầu như các phần mềm chỉnh sửa ảnh đều sử dụng 1 hoặc nhiều phương pháp nội suy. Hình ảnh sẽ mịn màng, không bị "vỡ hạt" khi phóng to tùy vào thuật toán được sử dụng trong giải thuật nội suy. Điều quan trọng là giải thuật nội suy sẽ không thêm thông tin gì mới cho hình ảnh cả, nó chỉ thêm điểm ảnh và làm tăng dung lượng của tập tin.

Tóm lại: Nội suy là quá trình sử dụng để ước lượng một giá trị ảnh ở một vị trí giữa các pixel.

1.2.2. Các vấn đề với nội suy ảnh

Làm thế nào để chúng ta nhận được bức ảnh với cỡ lớn mà chất lượng vẫn đảm bảo. Rõ ràng chúng ta cần tạo ra nhiều điểm ảnh (pixel). Quá trình tạo thêm điểm ảnh này gọi là nội suy ảnh. Điều này có thể thực hiện bởi phần mềm nội suy ảnh. Phần mềm nội suy phân tích các điểm ảnh trong một hình ảnh và cho biết thêm chi tiết dựa trên các điểm ảnh được phân tích. Ví dụ nội suy thay đổi kích thước của hình ảnh 200%, nói cách khác nó gấp đôi kích thước của hình ảnh. Kết quả là mỗi điểm ảnh sẽ trở thành 4 điểm ảnh (chủ yếu là 2x2). Những hình ảnh có kích thước gấp 4 và bây

giờ chúng ta có thể in hình ảnh đó với chất lượng tốt hơn. Còn khi sử dụng nội suy trong việc tô màu vùng thì giá trị các pixel được nội suy từ viền của vùng. Các pixel thêm vào này xác định giá trị màu nội suy từ các màu lân cận. Việc nội suy giá trị màu được ứng dụng nhiều trong công việc nắn chỉnh hình học, bóp méo, sinh ảnh.. hình ảnh sẽ trở lại trạng thái đẹp như ban đầu và có màu như màu gốc dựa vào việc nội suy. Tuy nhiên khi nội suy vẫn có một số vấn đề sau:

- Nội suy không tạo mới dữ liệu

Khi phóng to ảnh, nội suy chỉ chèn thêm điểm ảnh chứ không tạo mới dữ liệu. Tức là các thông tin của bức ảnh sẽ được phân bố trên một diện tích lớn hơn khi phóng to ảnh, như vậy bức ảnh sẽ không nhìn tốt hơn như ở kích thước ban đầu. Hình ảnh có thể bị xuống cấp. Như vậy một tập tin được nội suy sẽ trông không giống như một hình ảnh chưa nội suy. Nội suy chỉ khắc phục được hiện tượng răng cưa, làm hình ảnh mịn màng chứ không giữ nguyên trạng thái của bức ảnh ban đầu.

- Nội suy khắc phục hiện tượng răng cưa

Răng cưa là một thuật ngữ dùng để mô tả các đường thẳng hoặc đường cong không trơn mượt, trở thành răng cưa. Do bản chất của ảnh kỹ thuật số, mỗi ảnh kỹ thuật số được tạo ra bởi các điểm ảnh, bởi vậy các đường thẳng và các đường cong trong hình ảnh kỹ thuật số không thực sự là đường thẳng hay đường cong trơn mà là mô hình răng cưa của các điểm ảnh.

- Nội suy là nguyên nhân làm mất độ sắc nét

Cùng một quá trình làm trơn mượt các răng cưa, cũng giống như làm mượt các cạnh trong một hình ảnh. Một hình ảnh có cạnh sắc nét, đẹp (nó có một sự chuyển đổi ngẫu nhiên giữa màu đen hoặc trắng). Việc chuyển đổi màu đen/trắng thành màu đen/xám/trắng một cách mềm mại. Điều này đã làm giảm đi độ sắc nét của hình ảnh. Đây cũng là lý do tại sao hầu hết các công việc làm sắc nét nên được thực hiện sau khi nội suy.

- Nội suy không sinh ra hình ảnh thực sự tự nhiên

Kỹ thuật nội suy cho phép tạo ra các hình ảnh trung gian, rất hiệu quả, nhưng người ta không giám chắc rằng những trạng thái đó có thực sự tự nhiên hay không. Điều này hoàn toàn quyết định bởi khả năng đánh giá của người sử dụng đối với một trạng thái hình ảnh và thiết kế làm sao để đạt được kết quả tốt nhất. Nhược điểm của

các phương pháp nội suy hiện tại là không tính toán được sự thay đổi về điểm nhìn hoặc tư thế của vật thể. Điều này dẫn đến kết quả là những biến đổi hình ảnh ba chiều đơn giản (như: chuyển dịch hoặc xoay chiều) sẽ trở nên vô cùng khó khăn.

1.2.3. Nội suy ảnh 3 chiều (3D Morphing)

Morphing là viết tắt của Metamorphosing, là một kỹ xảo biến hình hoạt hình bằng máy tính mang tính cách mạng, được dùng để tạo ra các đối tượng trung gian từ hai đối tượng nguồn và đối tượng đích, nhìn như cái này biến thành cái kia một cách hợp lý.

Morphing là một kỹ thuật tạo hiệu ứng đặc biệt trong công nghiệp điện ảnh, giải trí, làm Game, nó cũng liên quan tới rất nhiều kỹ xảo khác, tạo cho máy khả năng tính toán làm biến đổi đối tượng, hoặc tạo ra các đối tượng trung gian từ các đối tượng được tạo ra thông thường.

Morphing là một khái niệm được đặt ra từ hơn một thập kỷ trước, cho chuỗi biến hình hoạt hình hiển thị bằng cách biến đổi từ từ giữa các đối tượng. Khái niệm này đã được sử dụng để biểu diễn các biến đổi của hình ảnh, các khối đa giác, bề mặt, và các phần cơ thể được chia nhỏ tới các thành phần nhỏ nhất của một khối thể tích được gọi là điểm ảnh thể tích (Voxel). Morphing là quá trình xử lý gồm hai giai đoạn, thứ nhất chúng ta phải Warping (phép kéo) đối tượng, sau đó là việc nội suy giữa hai đối tượng (Blending). Sở dĩ có hai bước trên là do khi ta Morphing hai đối tượng, thì có thể hai đối tượng sẽ không khớp nhau, hình thành nên những chuỗi kết quả không mong muốn, nên việc đầu tiên chúng ta phải Warping đối tượng đó trước cho tương thích với nhau, rồi sau đó chúng ta mới tiến hành nội suy bình thường. Warping (Phép kéo) là một khái niệm mà bản chất của nó là làm biến đổi cấu trúc hình học của đối tượng, từ hệ tọa độ này tới hệ tọa độ khác, mà nó không làm ảnh hưởng tới thuộc tính của đối tượng đồ họa như: Màu sắc, Texture Map, Chuẩn tắc..vv. Ta có thể ký hiệu phép Warping là : $O' = f(O)$, $f: R^n \rightarrow R^n$.

3D Morphing là phép biến hình dần dần giữa các đối tượng 3D. Các đối tượng 3D Morphing có thể được mô tả bằng hình học nguyên thủy hoặc bằng khối thể tích (Tập hợp dữ liệu thể tích). Mỗi đại diện yêu cầu các thuật toán Morphing khác nhau. Cách thức đại diện cho một đối tượng 3D hay dùng nhất trong đồ họa máy tính và đặc biệt trong các mô hình hoạt hình 3D, đó là các đối tượng 3D được đại diện bởi lưới đa

diện bằng đỉnh, các đỉnh này được phân bố theo một trật tự nhất định, các cạnh nối giữa các đỉnh, và các bề mặt, mỗi bề mặt bao gồm, một mặt chứa tập đỉnh với các cạnh liên kết. Mỗi mô hình bao gồm phần hình học tạo ra hình dạng đối tượng và một kết cấu bề mặt (texture) phủ lên bề mặt ngoài của dạng hình học đó.

Đầu vào là đối tượng nguồn và đối tượng mục tiêu, một chuỗi đối tượng trung gian được hình thành yêu cầu phải nội suy các yếu tố đầu vào, như vậy chuỗi đối tượng kết quả được tạo ra, đối tượng gần với đối tượng nguồn sẽ giống đối tượng nguồn hơn, và đối tượng gần với đối tượng mục tiêu thì giống đối tượng mục tiêu hơn, trong trình tự của chuỗi các đối tượng.

Đầu vào: Hình ảnh (2D) hoặc đối tượng 3D gốc.

Quá trình xử lý: Dùng các kỹ thuật Morphing

Đầu ra: Hình ảnh (2D) hoặc đối tượng 3D mục tiêu.

Có rất nhiều kỹ thuật được dùng trong việc biến đổi đối tượng, trong đó 3D Morphing bao gồm các phép biến đổi ba chiều đơn giản, như các phép quay, phép tỉ lệ, phép tịnh tiến..vv. Ngoài ra, còn là những phép nội suy phức tạp như Affine, Bilinear được sử dụng trong những bài toán yêu cầu nội suy tập điểm giữa đối tượng này tới đối tượng khác, hoặc những thuật toán phức tạp khác. Đối với mỗi bài toán, mỗi yêu cầu khác nhau, mà chúng ta có thể ứng dụng từng kỹ thuật khác nhau vào quá trình xử lý Morphing để đạt được kết quả hiệu quả nhất, theo đúng yêu cầu. Một số kỹ thuật 3D Morphing.

1. Morphing lưới đa diện (Polygonal mesh): Morphing đa giác là quá trình thực thi biến hình đối tượng nguồn và đối tượng mục tiêu được biểu diễn bằng đa giác hoặc lưới đa giác. Quá trình biến đổi sử dụng đa diện hoặc lưới đa diện làm đầu vào, lưới đa diện được định nghĩa như một bề mặt tuyến tính bao gồm các đa giác được mô tả như là tập cấu trúc của Vertex/Face/Edge/Graph hay một tập Vertex được phân bố theo một trật tự nhất định. Một đặc trưng quan trọng của một mô hình mặt lưới chính là mô tả tính topo của mô hình và đây là cơ sở nghiên cứu mạnh với phần lớn các bài báo liên quan tới lĩnh vực Morphing.

Ưu điểm: Định dạng lưới đa giác là định dạng phổ biến, dễ tạo ra, hỗ trợ nhiều gói mô hình, trong đó cũng đi kèm với rất nhiều công cụ kỹ thuật đồ họa để tạo ra mô hình như 3DSMax, Maya..vv, và đã được phát triển khá nhiều trong các ngành công

nghiệp giải trí. Phần lớn các phương pháp tiếp cận theo dạng này, và chúng cũng hỗ trợ rất tốt trong việc tính toán xử lý và tạo đối tượng nhanh hơn

Hạn chế: Bộ dữ liệu từ các lĩnh vực y tế, địa chất không hỗ trợ dạng dữ liệu này. Tính topo phức tạp của mô hình dẫn đến khó trong việc xử lý quá trình Morphing đặc biệt nếu chúng yêu cầu người sử dụng điều khiển hay kiểm soát. Thực thi phức tạp trong quá trình xử lý và hiển thị.

Hai phần của vấn đề về Morphing bằng dữ liệu kiểu này: Đó là vấn đề tương thích, và vấn đề nội suy.

2. Morphing thể tích (Volume Morphing): Sử dụng khối thể tích 3 chiều như thể tích đầu vào cho quá trình xử lý Morphing. Các mô hình thể tích được mô tả dưới dạng các Voxel. Voxel viết tắt của Volume Pixel là đơn vị khối hình hộp(Box) nhỏ nhất của thể tích, gọi là điểm ảnh thể tích.

Volume được định nghĩa như là một tập các Voxel phân tán, với mỗi Voxel được liên kết với một tập giá trị với kích thước S, nghĩa là thể tích V được cho bởi: $V = \{(x_i, v_i) \mid x_i \in \mathbb{R}^3, v_i \in \mathbb{R}^S, i = 1 \dots n\}$ [Chen1995]

Phần lớn định dạng thường gặp cho biểu diễn dữ liệu thể tích đó là trong định dạng của một mạng lưới tọa độ ba chiều hay gọi là lưới ô vuông ba chiều (Grid). Mỗi ô lưới được biểu diễn bởi vị trí tọa độ x, y, z của chúng. Mỗi vị trí (i, j, k) được biểu diễn như một vị trí Voxel trong đó có một giá trị liên kết tới nó.

Ưu điểm: Cách tiếp cận này không yêu cầu hạn chế về mặt topo và hình học đối với mô hình. Morphing thể tích có thể dễ dàng áp dụng đối với mô hình dạng mắt lưới(Mesh) bằng cách chuyển đổi chúng sang dữ liệu thể tích, trong khi điều ngược lại có thể dẫn đến tính topo gây khó khăn cho việc biến hình Morphing. Một số lượng lớn dữ liệu trong ngành y tế, địa chất, và các lĩnh vực năng lượng được tạo ra trong định dạng dữ liệu thể tích và có thể Morphing trực tiếp trên tập dữ liệu đó. Phần lớn các thuật toán Morphing thể tích không cần một ánh xạ song ánh, giữa tập đỉnh từ đối tượng nguồn tới đối tượng đích như trong kỹ thuật Morphing Mesh. Hơn nữa, định dạng thể tích là định dạng đơn giản cho phép thực thi biến hình một cách dễ dàng.

Hạn chế: Phương pháp thể tích này không phải là định dạng phổ biến như các định dạng đa giác trong ngành công nghiệp giải trí và các mô hình thử nghiệm có sẵn cũng rất hạn chế. Phương pháp này, yêu cầu các tính toán chuyên sâu để xử lý và tạo đối tượng. [Cohen-Or1998] sử dụng phương pháp biến đổi trường khoảng cách. Sử dụng

các điểm đặc trưng do người dùng định nghĩa. Quá trình Morphing sử dụng phép kéo (Warping) và cross-dissolving hay phép trộn Blending, nhược điểm là chất lượng Morphing phụ thuộc vào tập điểm đặc trưng.

Vấn đề với công nghệ 3D Morphing hiện tại: Đó là vấn đề về tương thích, và yêu cầu người sử dụng định nghĩa các điểm đặc trưng.

Ngày nay với sự phát triển mạnh mẽ của khoa học kỹ thuật, cùng với sự phát triển mạnh mẽ của Thực tại ảo, đã đưa con người tới những nhu cầu cao về các lĩnh vực giải trí, thiết kế kiến trúc, trong việc đào tạo khoa học giáo dục, và cả trong lĩnh vực y tế. Ở đó con người được tận dụng hết những thế mạnh, cũng như những ứng dụng vô cùng to lớn của nó mang lại, đó là khả năng mô hình hóa cụ thể các đối tượng, mô tả được một cách trực quan nhất, để người dùng có thể có những đánh giá khách quan nhất về ý tưởng cũng như có thể dễ dàng thay đổi những ý tưởng của mình, đồng thời cũng tạo cho con người khả năng cảm nhận được sự hiện diện của mình, khả năng tự trị, và những phản hồi rất khác quan từ phía môi trường tới các giác quan của người sử dụng. Ngày nay cuốn theo những nhu cầu về giải trí, con người càng có nhiều ý tưởng sáng tạo dựa trên cái nền phát triển đã có của Thực tại ảo, con người đã biết dựa vào đó để phát triển nên những trò chơi thực tại ảo thu hút một số lượng lớn người quan tâm, cũng như sử dụng nó để phát triển nên những bộ phim mang tính chất lịch sử, mang lại giá trị kinh tế vô cùng lớn. Trong những ứng dụng vô cùng mạnh mẽ đó của thực tại ảo, thì kỹ thuật 3D Morphing cũng mang lại những thành công nhất định trong các lĩnh vực như giải trí, điện ảnh, các trò chơi giải trí, hoặc các bộ phim hoạt hình ăn khách, làm quảng cáo. Ngoài ra, cũng được ứng dụng trong tất cả các lĩnh vực khác nữa, như giáo dục, nghiên cứu quá trình phát triển tiến hóa của người hoặc sự phát triển của các loài cây, và dùng để tái tạo hay biến đổi khuôn mặt để tạo ra những kết quả mong muốn trong quá trình nghiên cứu.

Chương 2:

KỸ THUẬT 3D MORPHING LƯỚI

2.1. Morphing lưới đa diện (Polygonal mesh)

Bài Toán:

Cho hai mô hình S và T, mỗi mô hình được biểu diễn dưới dạng lưới đa diện, một lưới được xem như là một phần tuyến tính của bề mặt, bao gồm tập hợp đa diện, mỗi đa diện được tạo nên từ các tam giác. Đặc trưng cơ bản của mô hình này đó là tính chất topo và tính hình học. Tính chất topo thể hiện cấu trúc của Vertex/ Edge/ Face. Và tính chất hình học được thể hiện là tập hợp các tọa độ đỉnh $V = \{v_1, v_2, \dots, v_n\}$ thuộc R^3 được phân bố theo một trật tự nhất định trên lưới đa diện. Mỗi mô hình bao gồm một kết cấu (texture) phủ lên bề mặt ngoài của dạng hình học.

Thuật toán 3D Morphing cho lớp đối tượng này thường yêu cầu hai bước:

- Bước thứ nhất: Thiết lập một tương ứng từ mỗi điểm trên S tới mỗi điểm trong T
- Bước thứ hai: Sử dụng tương ứng trên, bước này là tạo ra một chuỗi các đối tượng trung gian, bằng cách nội suy tương ứng các điểm từ vị trí gốc trên S tới vị trí mục tiêu trên T.

Hai bước trên được gọi tương ứng là vấn đề tương thích và vấn đề nội suy. Sau đây chúng ta sẽ xem xét cụ thể từng bước của thuật toán, để có thể tạo ra một mô hình mục tiêu hợp lý.

2.2. Rút trích các vector biến đổi

Việc thiết lập một tương ứng từ mỗi điểm trên mô hình nguồn S tới mô hình đích T thực tế là biến đổi các điểm trên mô hình theo vector tương ứng để tạo ra các biến đổi từ trạng thái nguồn tới trạng thái mục tiêu. Việc tính toán vector tương ứng được tính toán như sau. Với mỗi một điểm trên mô hình sẽ có một vector biến đổi tương ứng. Vector này có gốc tại điểm trên mô hình nguồn và ngọn là điểm tương ứng với mô hình mục tiêu. Bên cạnh các tọa độ x, y, z của vector, cần lưu trữ thêm chỉ số của điểm trong mô hình ứng với vector đó để có thể xác định điểm cần dịch chuyển khi thực hiện biến đổi các mô hình sau này.

Giả sử có N là tập điểm của mô hình nguồn và T là tập điểm của mô hình mục tiêu.

$$N = (x_{11}, y_{11}, z_{11}, \dots, x_{1n_1}, y_{1n_1}, z_{1n_1}, \dots, x_{mn_m}, y_{mn_m}, z_{mn_m}) \in \mathcal{R}^{3 \times m \times n}$$

$$T = (x_{11}, y_{11}, z_{11}, \dots, x_{1n_1}, y_{1n_1}, z_{1n_1}, \dots, x_{mn_m}, y_{mn_m}, z_{mn_m}) \in \mathcal{R}^{3 \times m \times n}$$

Ta cần tính ra tập vector V để biến đổi từ N sang T , cùng với tập chỉ số I tương ứng. Công thức tính như sau:

$$\begin{cases} V_{ik} = T_{ij} - N_{ij} \\ I_{ik} = j \end{cases} \quad \text{với} \quad \begin{cases} j \in [1, n_i] / N_{ij} \neq T_{ij} \\ k \in [1, p_i] \end{cases}$$

Trong đó:

m : số bộ phận của mô hình chịu sự biến đổi.

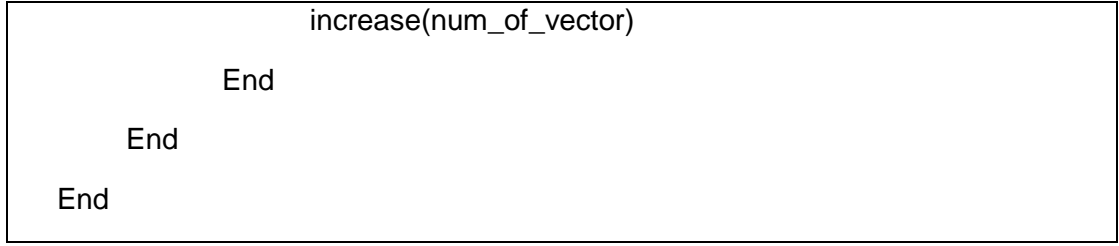
n_i : số điểm trên bộ phận thứ i .

p_i : số vector biến đổi của bộ phận thứ i .

Dưới đây là thuật toán để tính vector biến đổi cho một phép biến đổi từ mô hình nguồn sang một mô hình mục tiêu tương ứng.

Bảng 1: Thuật toán rút trích vector biến đổi

Function Set_Of_Deforming_Vector
<p><u>Input:</u></p> <p><i>ScrModel</i>: Mô hình cân bằng.</p> <p><i>TargetModel</i>: Mô hình được biến đổi theo âm tiết.</p> <p><u>Output:</u></p> <p><i>vector</i>: Tập vector biến đổi tương ứng từ mô hình nguồn tới mô hình mục tiêu</p> <p><i>vec_index</i>: Tập các chỉ số của điểm trên mô hình có vector biến đổi tương ứng trong tập vector.</p>
<pre> For each part ∈ (Parts of Model) Begin {Tính tập vector biến đổi cho bộ phận part} num_of_vector_{part} = 0 For i = 1 → num_of_point(part) Begin {Tính vector biến đổi tại điểm i } If ($X_{i(ScrModel)} \neq X_{i(TargetModel)}$ and $Y_{i(ScrModel)} \neq Y_{i(TargetModel)}$ and $Z_{i(ScrModel)} \neq Z_{i(TargetModel)}$) Begin $vector_i = (X_{i(TargetModel)} - X_{i(ScrModel)} ,$ $Y_{i(TargetModel)} - Y_{i(ScrModel)} ,$ $Z_{i(TargetModel)} - Z_{i(ScrModel)})$ $vec_index_{num_of_vector} = i$ End End End End End </pre>



Sau khi hoàn thành bước trên, ta sẽ có được một tập các vector tương ứng cho tất cả các điểm trên mô hình. Mỗi điểm được biến đổi dựa trên vector biến đổi của chính nó.

2.3. Nạp mô hình nguồn và biến đổi mô hình mục tiêu

Tập các vector biến đổi đã được tạo ra ở trên, sẽ cho phép chúng ta biến đổi giữa mô hình nguồn và mô hình mục tiêu. Quá trình biến đổi mô hình được thực hiện như sau:

Đầu tiên, ta có mô hình nguồn cần biến đổi, sau đó dựa vào các vector đã được tính toán ở trên ta tiến hành biến đổi mô hình nguồn, bằng cách dịch chuyển các điểm trên mô hình nguồn theo các vector biến đổi tương ứng. Bây giờ giả sử ta có N tập điểm của mô hình nguồn, V tập vector biến đổi, và I là tập chỉ số vector.

$$N = (x_{11}, y_{11}, z_{11}, \dots, x_{1n_1}, y_{1n_1}, z_{1n_1}, \dots, x_{mn_m}, y_{mn_m}, z_{mn_m}) \in \mathfrak{R}^{3 \times m \times n}$$

$$V = (x_{11}, y_{11}, z_{11}, \dots, x_{1n_1}, y_{1n_1}, z_{1n_1}, \dots, x_{mp_m}, y_{mp_m}, z_{mp_m}) \in \mathfrak{R}^{3 \times m \times p}$$

$$I = \{k \in [1, n_i] / T_{ik} \neq N_{ik}\}$$

Trong đó:

m : số bộ phận của mô hình chịu sự biến đổi.

n_i : Số điểm trên bộ phận thứ i .

Ta cần tính ra tập điểm T của mô hình mục tiêu. Công thức tính như sau:

$$\begin{cases} j = I_{ik} \\ T_{ij} = N_{ij} - N_{ik} \end{cases} \quad \text{với} \quad \begin{cases} i \in [1, m] \\ k \in [1, n_i] / N_{ik} \neq T_{ik} \\ j \in [1, n_i] \end{cases}$$

Bảng 2: Thuật toán biến đổi mô hình

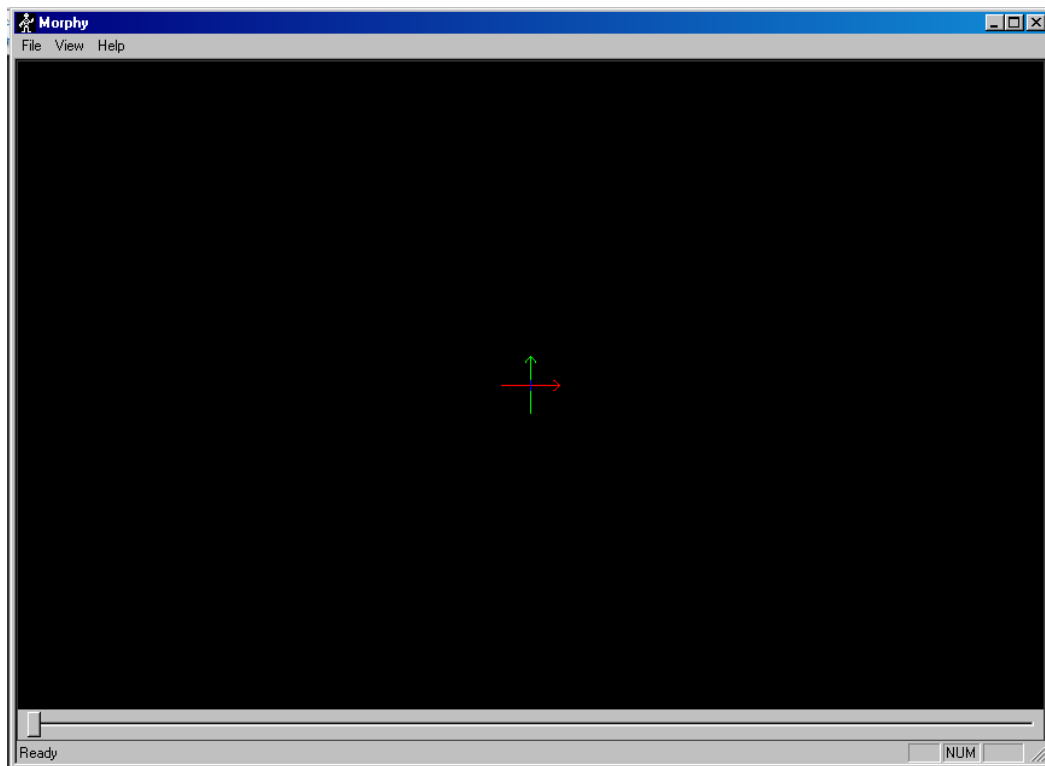
Function Set_Of_Deforming_Vector
<p><u>Input:</u></p> <p><i>ScrModel</i>: Mô hình nguồn.</p> <p><i>vector</i>: Tập vector biến đổi tương ứng từ mô hình nguồn tới mục tiêu</p> <p><i>vec_index</i>: Tập các chỉ số của điểm trên mô hình có vector biến đổi tương ứng trong tập vector.</p> <p><u>Output:</u></p> <p><i>TargetModel</i>: Mô hình mục tiêu.</p>
<pre> For each part ∈ (Parts of Model) Begin {Biến đổi bộ phận part} For i = 1 → num_of_vector(part) Begin {Dịch chuyển điểm trên mô hình ứng với vector biến đổi thứ i} point_{vec_index (i)} (TargetModel) ← point_{vec_index (i)} (ScrModel)+vector_i End End End End </pre>

Chương 3:

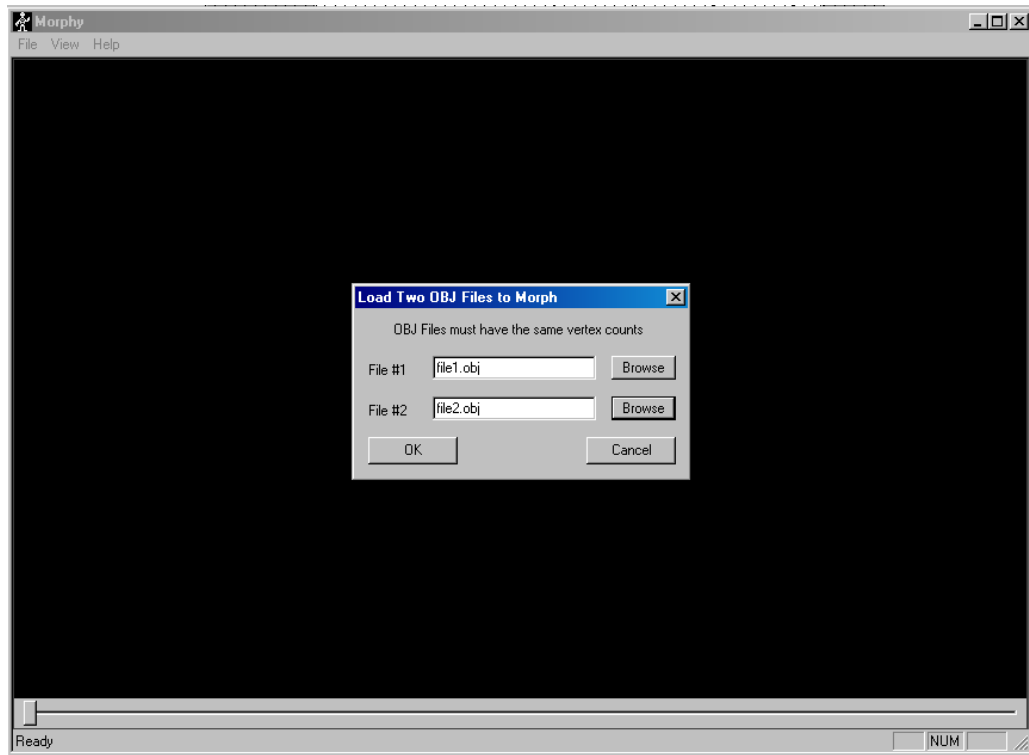
CHƯƠNG TRÌNH THỬ NGHIỆM

Em đã cài đặt thử nghiệm chương trình Morph thể hiện kỹ thuật 3D Morphing bằng ngôn ngữ Visual Studio C++ 2005. Đầu vào của chương trình là một mô hình 3D gốc và một mô hình 3D đích. Các mô hình 3D này được tạo ra từ trước bởi các chương trình tạo mô hình chuyên dụng trong thiết kế đồ họa như là: 3D Studio Max, Maya v.v..

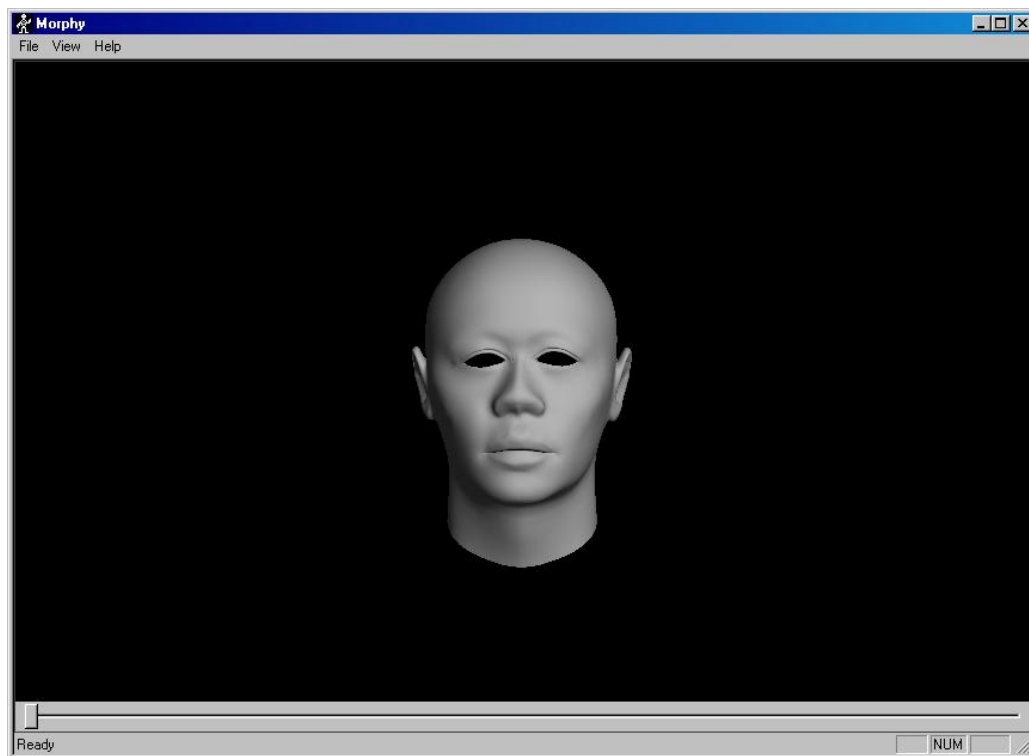
3.1. Giao diện của chương trình



Hình 3.1: Giao diện chính của chương trình



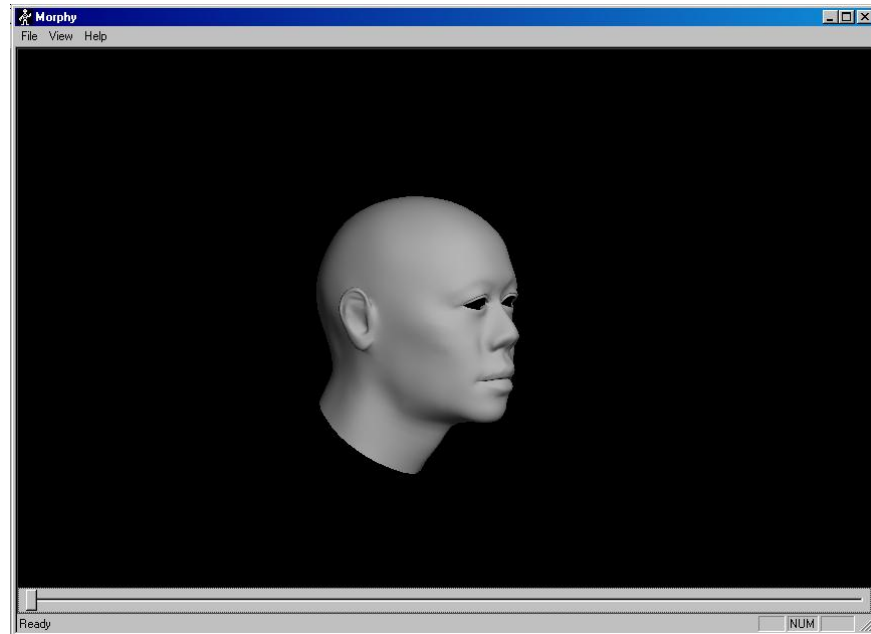
Hình 3.2: Giao diện tải mô hình 3D nguồn và mô hình 3D đích



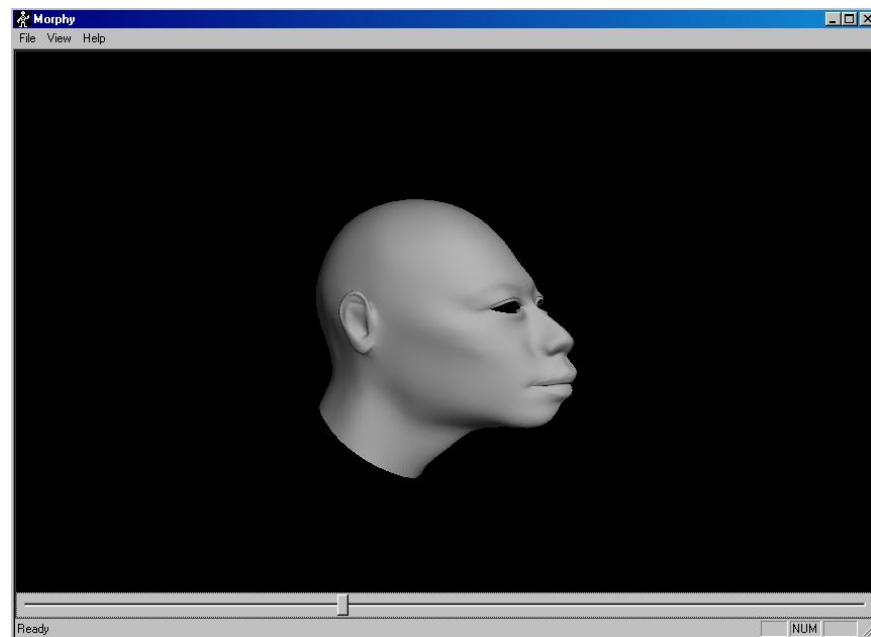
Hình 3.3: Mô hình 3D nguồn được tải lên

3.2. Kết quả của chương trình

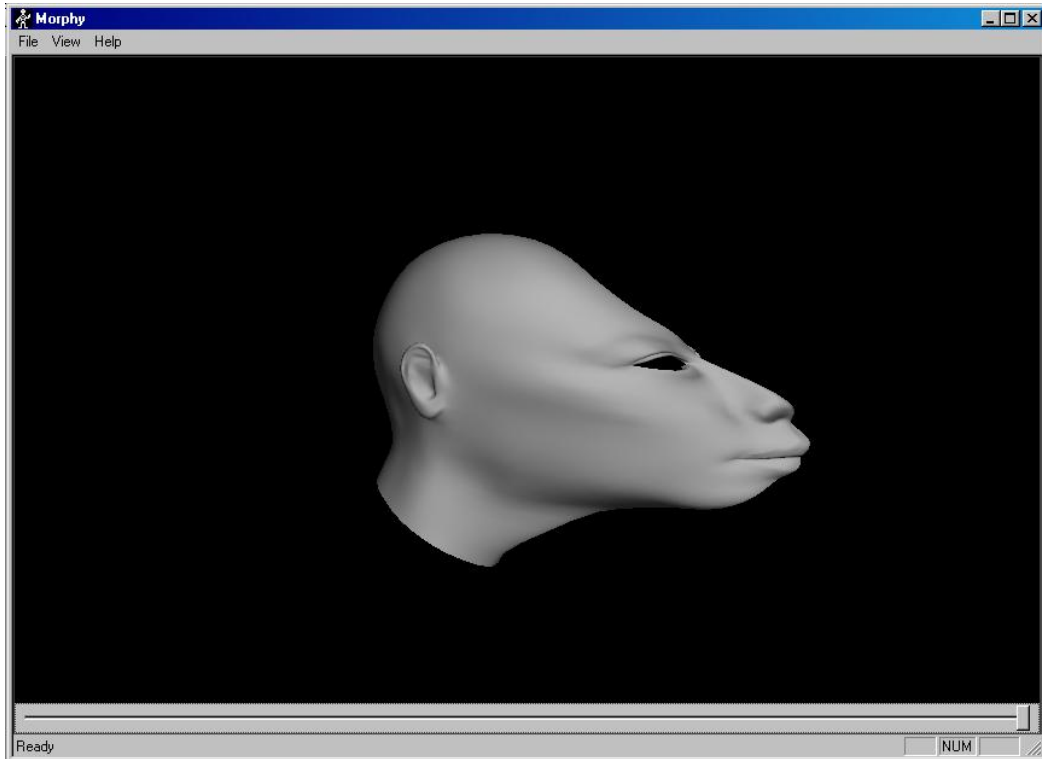
Sau khi hoàn thành quá trình tải mô hình nguồn và mô hình đích, ta tiến hành morphing bằng cách kéo thanh chạy ở bên dưới.



Hình 3.4: Mô hình nguồn khi chưa morphing



Hình 3.5: Mô hình nguồn sau khi được morphing



Hình 3.6: Mô hình đích sau khi kết thúc quá trình morphing

PHẦN KẾT LUẬN

Trong những năm gần đây công nghệ thông tin phát triển với tốc độ nhanh chóng về cả phần cứng và phần mềm. Sự phát triển của công nghệ thông tin đã thúc đẩy sự phát triển của nhiều lĩnh vực xã hội khác như: y học, giáo dục, giải trí, kinh tế v.v. Sự phát triển của phần cứng cả về phương diện thu nhận, hiển thị, cùng với tốc độ xử lý đã mở ra nhiều hướng mới cho sự phát triển phần mềm, đặc biệt là lĩnh vực xử lý ảnh cũng như công nghệ thực tại ảo đã ra đời và thâm nhập mạnh mẽ vào đời sống của con người.

Ảnh thu được sau quá trình thu nhận ảnh hoặc các phép biến đổi không tránh khỏi nhiều hoặc khuyết thiếu. Sự sai sót này một phần bởi các thiết bị quang học và điện tử, phần khác bởi bản thân các phép biến đổi không phải là toàn ánh, nên có sự ánh xạ thiếu hụt đến những điểm trên ảnh kết quả. Việc khắc phục những nhược điểm này luôn là vấn đề đặt ra cho các hệ thống xử lý ảnh, đặc biệt đối với các hệ thống xử lý ảnh 3D.

Đồ án đã giải quyết được mục tiêu ban đầu là tìm hiểu kỹ thuật 3D Morphing. Các kết quả chính:

- 1) Tìm hiểu khái quát về xử lý ảnh và nội suy ảnh
- 2) Tìm hiểu về 3D Morphing và kỹ thuật nội suy lưới 3D morphing
- 3) Cài đặt chương trình thử nghiệm

Tuy nhiên, vẫn còn một số vấn đề mà đồ án chưa đề cập đến, một số hướng phát triển khác nữa có thể mở rộng như: xử lý điều kiện mô phỏng theo nguồn sáng, xử lý các bề mặt có kết cấu khác nhau, tính toán độ phức tạp hình học.

Mặc dù em đã có nhiều nỗ lực trong tìm hiểu và thực hiện đề tài, nhưng vì thời gian và trình độ có hạn, chắc chắn rằng đề án không tránh khỏi nhiều thiếu sót. Em hy vọng rằng kỹ thuật 3D Morphign trong việc sinh tạo hình ảnh ba chiều sẽ là đề tài tiếp tục nhận được nhiều sự quan tâm nghiên cứu và phát triển mạnh mẽ trong thời gian tới.

TÀI LIỆU THAM KHẢO

- [1]. **Giáo trình xử lý ảnh** ; PGS.TS Đỗ Năng Toàn, TS Phạm Việt Bình.
- [2]. **Kỹ thuật đồ họa** ; Ths Trịnh Thị Vân Anh.
- [3]. **Image Warping and Morphing** ; Leow Wee Kheng, CS5245 Vision & Graphics for Special Effects.
- [4]. **A study on face morphing algorithms.**
- [5]. **3D Morphing using strain field interpolation** ; J. Comput. Sci. & Technol, Month 200X, Vol.21, No.X, pp.XX–XX

PHỤ LỤC

1. Các phép biến đổi 3 chiều

Trong không gian ba chiều ta cũng có các phép biến đổi giống như trong không gian hai chiều. Các phép biến đổi sẽ được minh họa qua các ma trận 4×4 , tọa độ các điểm được biểu diễn theo tọa độ đồng nhất nghĩa là thay cho tọa độ (x, y, z) ta sẽ dùng $(x, y, z, 1)$.

1.1 Hệ tọa độ

1.1.1 Hệ tọa độ bàn tay trái

Hệ tọa độ theo qui ước bàn tay trái: để bàn tay trái sao cho ngón cái hướng theo trục z , khi nắm tay lại, các ngón tay chuyển động theo hướng từ trục x đến trục y .

1.1.2 Hệ tọa độ bàn tay phải

Hệ tọa độ theo qui ước bàn tay phải: để bàn tay phải sao cho ngón cái hướng theo trục z , khi nắm tay lại, các tay chuyển động theo hướng từ trục x đến trục y .

1.1.3 Hệ tọa độ đồng nhất

Hệ tọa độ thuần nhất (Homogeneous Coordinates): Mỗi điểm (x, y, z) trong không gian Descartes được biểu diễn bởi một bộ bốn tọa độ trong không gian 4 chiều thu gọn (hx, hy, hz, h) . Người ta thường chọn $h=1$.

Các phép biến đổi tuyến tính là tổ hợp của các phép biến đổi sau: tỉ lệ, quay, biến dạng và đối xứng. Các phép biến đổi tuyến tính có các tính chất sau:

- Gốc tọa độ là điểm bất động
- Ảnh của đường thẳng là đường thẳng
- Ảnh của các đường thẳng song song là các đường thẳng song song
- Bảo toàn tỉ lệ khoảng cách

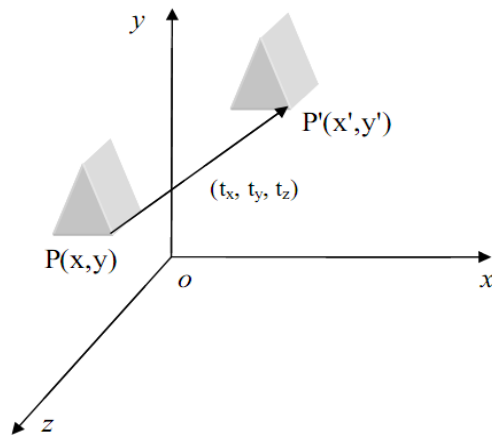
- Tổ hợp các phép biến đổi có tính phân phối

1.2 Các phép biến đổi Affine cơ sở

1.2.1 Phép tịnh tiến

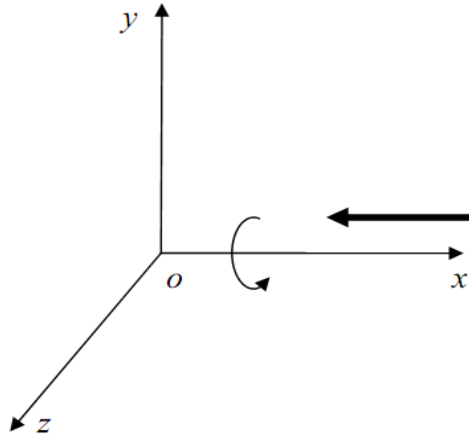
Giả sử véc tơ tịnh tiến là (t_x, t_y, t_z) khi đó phương trình phép tịnh tiến như sau (T là ma trận của phép tịnh tiến):

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = R \cdot P = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



1.2.2 Phép quay

Khi thực hiện phép quay trong không gian ba chiều ta cần phải biết trục quay và góc quay. Chiều của góc quay được xác định theo chiều cùng chiều kim đồng hồ (chiều âm) và ngược chiều kim đồng hồ (chiều dương) khi mắt nhìn dọc theo trục quay (ta sẽ gọi đó là hướng nhìn). Ví dụ phép quay minh họa trên hình 3.6 là phép quay theo chiều dương, trục quay ox và hướng nhìn là theo hướng âm của trục ox .



Phép quay chiều dương, trục quay ox, hướng nhìn là hướng âm trục ox

Cách xác định chiều dương trong các phép quay

Định nghĩa về chiều quay được dùng chung cho cả hệ tọa độ theo qui ước bàn tay phải và bàn tay trái. Cụ thể chiều dương được định nghĩa như sau:

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

1.3 Các phép quay quanh trục tọa độ

Trước hết ta xét các phép quay một góc θ theo các trục Ox, Oy, Oz khi hướng nhìn là hướng âm của trục đó.

Phép quay quanh trục Oz: Ta có thể có công thức biến đổi tọa độ bằng cách mở rộng công thức phép quay trong không gian hai chiều

$$\begin{cases} x' = x \cos\theta + y \sin\theta \\ y' = -x \sin\theta + y \cos\theta \\ z' = z \end{cases}$$

Biểu diễn tương đương dưới dạng ma trận như sau:

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = R_{oz}(\theta) \cdot P = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Phép quay quanh trục Ox:

$$\begin{cases} x' = x \\ y' = y \cos \theta - z \sin \theta \\ z' = y \sin \theta + z \cos \theta \end{cases}$$

Biểu diễn tương đương dưới dạng ma trận như sau:

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = R_{ox}(\theta) \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Phép quay quanh trục Oy:

$$\begin{cases} x' = x \cos \theta + z \sin \theta \\ y' = y \\ z' = -x \sin \theta + z \cos \theta \end{cases}$$

Biểu diễn tương đương dưới dạng ma trận như sau:

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = R_{oy}(\theta) \cdot P = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Phép quay quanh một trục song song với trục tọa độ:

Giả sử trục quay song song với trục Ox (các trường hợp còn lại tương tự), ta thực hiện lần lượt các lệnh biến đổi sau:

(1) Áp dụng phép tịnh tiến T để đưa trục quay về trục Ox

(2) Áp dụng phép quay $R(\theta)$ quay đối tượng quanh trục ox một góc θ

(3) Áp dụng phép tịnh tiến T^{-1} để đưa trục quay về vị trí ban đầu

Phép quay quanh một trục bất kì:

Giả sử trục quay là một đường thẳng d đi qua hai điểm $P1(x1, y1, z1)$ và $P2(x2, y2, z2)$. Phép quay $R(\theta)$ quanh đường thẳng d một góc θ theo hướng nhìn từ điểm $P2$ tới $P1$. Ta thực hiện lần lượt các phép biến đổi sau:

(1) Áp dụng phép tịnh tiến đưa trục quay về vị trí đi qua gốc tọa độ

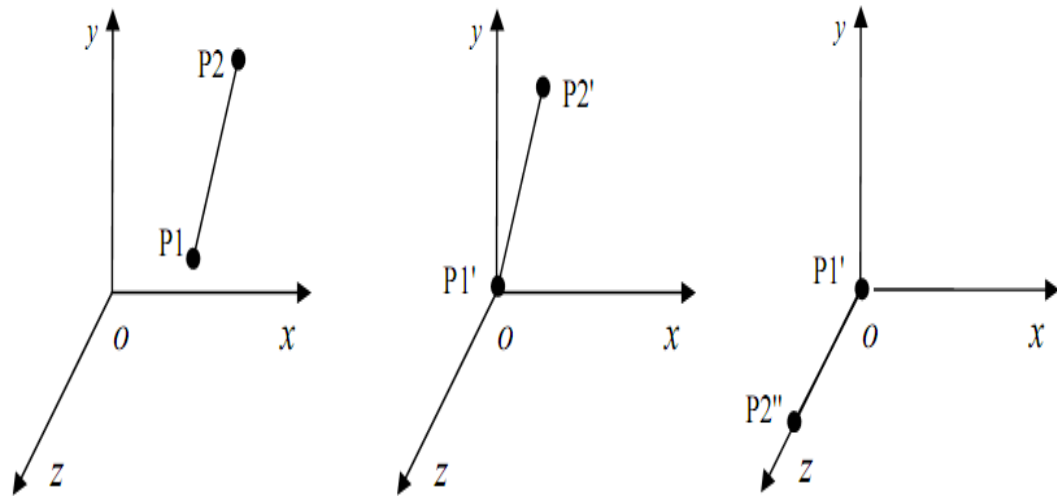
(2) Áp dụng phép quay đưa trục quay về vị trí trùng với một trục tọa độ

(3) Áp dụng phép quay vật thể quanh trục quay (trục tọa độ)

(4) Áp dụng phép quay đưa trục quay về vị trí tại bước 2

(5) Áp dụng phép quay đưa trục quay về vị trí ban đầu

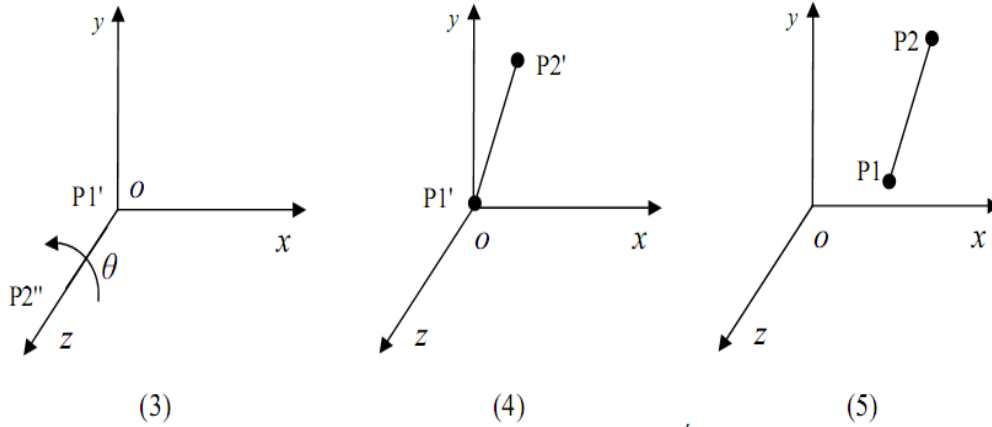
Quá trình được minh họa trong hình:



Vị trí ban đầu

(1)

(2)



Phép quay quanh một trục bất kì

1.4 Phép tỉ lệ

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = S \cdot P = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

1.5 Phép đối xứng

Trong không gian ta có thể thực hiện phép đối xứng qua một đường thẳng hoặc qua một mặt phẳng. Trong mục này ta chỉ xét khi trục đối xứng là các trục tọa độ và mặt phẳng đối xứng là mặt phẳng xy , yz hoặc xz .

Nếu trục tọa độ là trục đối xứng thì phép đối xứng tương đương với phép quay 180^0 quanh trục đó. Ví dụ phép đối xứng qua trục ox có ma trận là:

$$Re_{ox} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 180^0 & -\sin 180^0 & 0 \\ 0 & \sin 180^0 & \cos 180^0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Phép đối xứng qua mặt phẳng xy có ma trận biến đổi là:

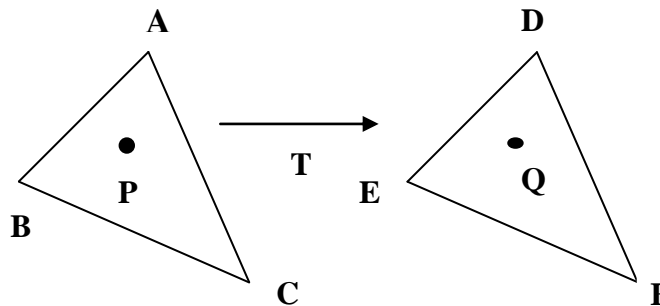
$$R_{e_{xy}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tương tự khi mặt phẳng đối xứng là xz hoặc yz

2. Các phép nội suy 3 chiều

2.1 Phép nội suy Affine

Đây là phép nội suy hai tam giác trong hệ tọa độ Euclide. Giả sử chúng ta có hai tam giác và muốn nội suy hai tam giác này cho nhau. Một cách đơn giản nhất là sử dụng kỹ thuật ánh xạ dựa trên hệ tọa độ Barycentric được minh họa như sau:



Hình 4.1: Phép nội suy Affine

Trước tiên chúng ta định nghĩa một ánh xạ T cho các đỉnh của tam giác: $T(A)=D$, $T(B)=E$, $T(C)=F$. Với các điểm còn lại chúng ta sẽ ánh xạ chúng theo tọa độ Barycentric $(\alpha_1, \alpha_2, \alpha_3)$ nghĩa là:

$$P = \alpha_1 * A + \alpha_2 * B + \alpha_3 * C$$

Trong đó:

$$\alpha_i \geq 0 \text{ và } \alpha_1 + \alpha_2 + \alpha_3 = 1$$

Một điểm Q là ánh xạ của điểm P qua phép T được tính như sau:

$$\begin{aligned} Q = T(P) &= T(\alpha_1 * A + \alpha_2 * B + \alpha_3 * C) \\ &= \alpha_1 * T(A) + \alpha_2 * T(B) + \alpha_3 * T(C) \end{aligned}$$

$$= \alpha_1 * D + \alpha_2 * E + \alpha_3 * F$$

Như vậy, để sử dụng phép nội suy Affine thì ta phải chuyển từ hệ tọa độ Euclide sang hệ tọa độ Barycentric. Cách chuyển được thực hiện như sau:

Với mỗi điểm $M(x_m, y_m)$ nằm trong tam giác ABC thì chúng ta đều có thể biểu diễn tọa độ của nó theo tọa độ các đỉnh của tam giác như sau:

$$\begin{cases} X_m = u * x_a + v * x_b + w * x_c \\ Y_m = u * y_a + v * y_b + w * y_c \\ U + v + w = 1 \\ U, v, w \geq 0 \end{cases}$$

Giải hệ phương trình này ta được nghiệm duy nhất.

2.2 Phép nội suy Bilinear

Đây là kỹ thuật xác định một hàm biến đổi từ một hình vuông kích thước $[0,1] \times [0,1]$ tới một tứ giác trong không gian (tứ giác này không nhất thiết phải đồng phẳng). Nếu chúng ta giả sử tọa độ của khối hình vuông là u và v thì phép biến đổi B được thực hiện như sau:

$$M(u, v) = (1-u) \begin{matrix} A & D \\ B & C \end{matrix} + v \begin{matrix} 1-v \\ u \end{matrix}$$

Phép biến đổi được thực hiện tương đương với hai việc. Việc thứ nhất là nội suy trên các cạnh AD và BC thu được điểm P và Q.

$$P = (1-v) * A + v * D$$

$$Q = (1-v) * B + v * C$$

Việc tiếp theo là nội suy trên đoạn PQ sử dụng thông số u :

$$\begin{aligned} M(u, v) &= (1-u) * P + u * Q \\ &= (1-u)[(1-v) * A + v * D] + u[(1-v) * B + v * C] \\ &= (1-u)(1-v) * A + u(1-v) * B + uv * C + v(1-u) * D \end{aligned}$$

Vậy :

$$M(u, v) = (1-u)(1-v)*A + u(1-v)*B + uv*C + v(1-u)*D$$

