

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**MÃ VĂN THU**

**MÔ HÌNH 3D VÀ TỐI ƯU HÓA MÔ HÌNH TRONG  
THỰC TẠI ẢO**

**LUẬN VĂN THẠC SĨ NGÀNH CÔNG NGHỆ THÔNG TIN**

Hà Nội - 2016

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**MÃ VĂN THU**

**MÔ HÌNH 3D VÀ TỐI ƯU HÓA MÔ HÌNH TRONG  
THỰC TẠI ẢO**

**Ngành: Hệ thống thông tin**

**Chuyên ngành: Hệ thống thông tin**

**Mã số: 60480104**

**LUẬN VĂN THẠC SĨ NGÀNH CÔNG NGHỆ THÔNG TIN  
NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS.TS. ĐỖ NĂNG TOÀN**

**Hà Nội - 2016**

## **LỜI CAM ĐOAN**

Tôi Mã Văn Thu xin cam đoan những nội dung trình bày luận văn này là kết quả tìm hiểu, nghiên cứu của bản thân dưới sự hướng dẫn của PGS.TS. Đỗ Năng Toàn và các nhà nghiên cứu đi trước. Nội dung tham khảo, kế thừa, phát triển từ các công trình đã được công bố được trích dẫn, ghi rõ nguồn gốc. Kết quả mô phỏng, thí nghiệm được lấy từ chương trình của bản thân.

Nếu có gì sai phạm tôi xin hoàn toàn chịu trách nhiệm.

**Người cam đoan**

**Mã Văn Thu**

## LỜI CẢM ƠN

Trong quá trình thực hiện luận văn mặc dù gặp rất nhiều khó khăn nhưng tôi luôn nhận được sự quan tâm, giúp đỡ từ thầy cô, đồng nghiệp bạn bè và người thân. Đây là nguồn động lực giúp tôi hoàn thành luận văn này.

Tôi xin gửi lời chân thành cảm ơn tới PGS.TS. Đỗ Năng Toàn đã tận tình giúp đỡ, hướng dẫn và chỉ bảo trong quá trình thực hiện luận văn.

Tôi xin chân thành cảm ơn tới quý thầy, cô trường Đại học Công nghệ - Đại học Quốc gia Hà Nội đã tận tình chỉ bảo, truyền đạt những kiến thức quý báu giúp tôi hoàn thành nhiệm vụ học tập trong suốt thời gian theo học tại trường. Quý thầy cô đã giúp tôi có được những kiến thức quan trọng trong lĩnh vực Công nghệ thông tin, là nền tảng vững chắc cho những nghiên cứu của bản thân trong thời gian tới.

Tôi xin gửi lời chân thành cảm ơn tới anh, chị phòng Thực tại ảo - Viện Công nghệ Thông tin - Viện Hàn lâm Khoa học Việt Nam đã giúp đỡ tôi rất nhiều trong lĩnh vực mô phỏng, thực tại ảo.

Tôi xin cảm ơn anh em, đồng nghiệp đã giúp đỡ, ủng hộ tinh thần trong thời gian tôi tham gia học tập.

Cuối cùng, tôi xin cảm ơn tất cả những người đã luôn luôn quan tâm, sẻ chia và động viên tôi.

*Hà Nội, ngày 10 tháng 10 năm 2016*

*Mã Văn Thu*

## MỤC LỤC

LỜI NÓI ĐẦU .....	1
CHƯƠNG 1. THỰC TẠI ẢO VÀ BÀI TOÁN TỐI ƯU MÔ HÌNH .....	2
1.1. Khái quát về thực tại ảo và mô hình 3D trong thực tại ảo .....	2
1.1.1. Thực tại ảo .....	2
1.1.2. Cấu tạo mô hình 3D .....	7
1.1.3. Các phương pháp tạo mô hình phổ biến hiện nay .....	11
1.1.3.1. Phương pháp tạo mô hình bằng thiết kế dựa trên phần mềm 3D .....	11
1.1.3.2. Tạo mô hình bằng máy quét 3D .....	15
1.2. Bài toán tối ưu hóa mô hình 3D .....	18
1.2.1. Một số phương pháp tạo mô hình 3D .....	18
1.2.2. Đầu vào , đầu ra bài toán tối ưu hóa mô hình.....	19
1.2.3. Nguyên lý tối ưu mô hình 3D .....	20
CHƯƠNG 2. MỘT SỐ KỸ THUẬT TỐI ƯU HÓA MÔ HÌNH.....	22
2.1. Kỹ thuật tối ưu mô hình dựa trên lưới tam giác .....	22
2.1.1. Giới thiệu về tối ưu và các phương pháp tối ưu phổ biến .....	22
2.1.2. Phương pháp Incremental Decimation .....	23
2.1.3. Thuật toán đề xuất .....	28
2.2. Kỹ thuật tối ưu mô hình dựa trên lưới tứ giác.....	32
2.2.1. Chuyển mô hình bề mặt lưới tam giác của về mô hình bề mặt lưới tứ giác.....	33
2.2.2. Làm mềm lưới tứ giác .....	38
2.2.3. Tối ưu hóa lưới tứ giác .....	42
CHƯƠNG 3. THỰC NGHIỆM VÀ ỨNG DỤNG TỐI ƯU MÔ HÌNH 3D .....	49
3.1. Yêu cầu thực nghiệm, ứng dụng .....	49
3.1.1. Yêu cầu với thực nghiệm.....	49
3.1.2. Kiểm tra các mô hình đầu vào .....	50
3.2. Phân tích, lựa chọn công cụ .....	50
3.3. Một số kết quả thực nghiệm tối ưu mô hình .....	51
3.3.1. Hướng dẫn sử dụng chương trình thực nghiệm.....	51
3.3.2. Một số kết quả tối ưu mô hình trên chương trình thực nghiệm.....	53
KẾT LUẬN .....	60
TÀI LIỆU THAM KHẢO .....	61

## DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT

3D	Three Dimentional	Ba chiều
VR	Virtual Reality	Thực tại ảo
	Virtual Environment	Môi trường ảo
	Surgical Simulation	Giải phẫu giả
	Error	Lỗi

## DANH MỤC CÁC HÌNH VẼ

Hình 1.1. Ứng dụng Thực tại ảo trong thiết kế nội thất .....	2
Hình 1.2. Ứng dụng Thực tại ảo trong thiết kế ô tô .....	3
Hình 1.3. Ứng dụng Thực tại ảo trong game giải trí .....	4
Hình 1.4. Ứng dụng Thực tại ảo trong phim Avatar .....	4
Hình 1.5. Mô phỏng lái tàu ảo của công ty mô phỏng việt nam .....	5
Hình 1.6. Mô phỏng cơ thể ảo của phòng thực tại ảo viện.....	6
Hình 1.7: Mô hình 3D tìm người.....	8
Hình 1.8. Mô hình 3D hình cầu cắt vát .....	8
Hình 1.9. Một mặt của hình hộp được tạo bởi 2 mặt tam giác.....	10
Hình 1.10. Mô hình 3D thu được từ tập đỉnh .....	11
Hình 1.11. Một số hình khối 3D cơ bản .....	12
Hình 1.12. Công cụ Select and Move trong thiết kế .....	12
Hình 1.13. Chế độ Editable Poly .....	13
Hình 1.14. Hai tấm plane chiều đứng và cạnh (cách 1).....	14
Hình 1.15. Hai tấm plane chiều đứng và cạnh (cách 2).....	15
Hình 1.16. Máy quét Artec Eva và mô hình thu được khi sử dụng máy quét này .....	16
Hình 1.17. Tổng thống Mỹ Obama và hình ảnh 3D của ông thu được từ máy quét.....	16
Hình 1.18. Máy quét TTO - Sense 3D.....	17
Hình 1.19. Máy quét Digitizer.....	18
Hình 1.20. Các mô hình được tạo ra từ máy quét có số lượng lưới cực lớn .....	19
Hình 1.21. Thu thập và làm mịn dữ liệu.....	20
Hình 1.22. Mô hình và xử lý từ máy quét 3D .....	20
Hình 1.23. Tối ưu hóa lưới .....	21
Hình 2.1 Kỹ thuật loại bỏ điểm .....	23
Hình 2.2 Loại bỏ và phục hồi bề mặt .....	24
Hình 2.3 Ví dụ về xóa điểm.....	25
Hình 2.4 Tối ưu lưới theo William J. Schroeder .....	25
Hình 2.5. Khoảng cách từ điểm tới mặt phẳng.....	26
Hình 2.6. Bề mặt cong .....	26
Hình 2.7. Tối ưu lưới theo The Gaussian Curvature .....	27
Hình 2.8. Các kết quả đơn giản hóa của một mô hình Igea .....	28
Hình 2.9. Góc tại đỉnh O.....	28
Hình 2.10. Góc tại đỉnh O và góc giữa 2 mặt phẳng kề nhau .....	29
Hình 2.11. Đỉnh O với nhiều cạnh kết nối.....	29
Hình 2.12. Mô hình trước và sau tối ưu .....	30
Hình 2.13. Sơ đồ khối việc xóa điểm .....	31
Hình 2.14. Mô hình lưới cho animation .....	33
Hình 2.15. Hàng đầu tiên của các yếu tố được đặt sử dụng thuật toán mở.....	34
Hình 2.16.: Các trạng thái của cạnh mặt trước.....	35
Hình 2.17. Các bước của quá trình xử lý tạo ra một tứ giác từ mặt trước $N_A - N_B$ .....	35

Hình 2.18. Lựa chọn cạnh bên.....	37
Hình 2.19. Tạo ra cạnh bên .....	38
Hình2.20. Bề mặt lưới 3D của mô hình .....	39
Hình2.21. Bề mặt lưới 3D của mô hình .....	39
Hình 2.22. Xử lý một trường hợp làm mịn.....	40
Hình 2.23. Lưới nhiều lớp .....	40
Hình 2.24. Lưới nhiều lớp đã được làm mịn .....	42
Hình 2.25. Mô hình lưới chuẩn để tạo chuyển động cho cánh tay .....	43
Hình 2.26. Mỗi đỉnh thường xuyên gây ra một hệ trục tọa độ .....	43
Hình 2.27. Sự khác nhau của bề mặt đối tượng với số lưới bằng nhau.....	44
Hình 2.28 Thể hiện sự chuyển cạnh .....	44
Hình 2.29. Ba hoạt động của việc gộp và tách kết nối của các điểm .....	45
Hình 2.30. Tập mô tả hữu hạn hợp lệ các khả năng để kết hợp ba họa động.....	45
Hình 2.31. Ví dụ về việc kết hợp 3 hoạt động trên và làm mềm kết quả.....	46
Hình 2.32. Đường nét đứt kiểm soát hướng đi bằng cách điều.....	46
Hình 2.33 Đường đi một chu trình .....	47
Hình 2.34. Sử dụng Quad- loop toàn cục .....	48
Hình 2.35. Các mô hình trước và sau khi được tối ưu .....	48
Hình 3.1. Mô hình con thỏ trong các bài thử nghiệm tối ưu lưới.....	50
Hình 3.2: Các mô hình khác sử dụng trong chương trình .....	50
Hình 3.3. Ảnh chụp chương trình thực nghiệm.....	51
Hình 3.4. Chạy file chương trình.....	52
Hình 3.5. Cửa sổ lựa chọn màn hình chạy chương trình.....	52
Hình 3.6. Tổng quan về chương trình.....	53
Hình 3.7 Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 75%.....	54
Hình 3.8. Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 50%.....	55
Hình 3.9. Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 25%.....	57
Hình 3.10. Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 10%.....	58
Hình 3.11. Lưới ghé tựa trước khi tối ưu với số lưới là 49314 mặt .....	59
Hình 3.12. Lưới ghé tựa sau khi tối ưu với số lưới là 18.494 mặt tam giác.....	59



## LỜI NÓI ĐẦU

Ngày nay, thực tại ảo (thực tế ảo) đã ngày càng chứng tỏ vai trò quan trọng trong đời sống cũng như trong khoa học, kỹ thuật. Thực tại ảo hiện diện ở hầu như mọi lĩnh vực giải trí, văn hóa, kinh tế, chính trị, quốc phòng, khoa học, đời sống v.v.. Trong thực tại ảo việc xây dựng các đối tượng 3D (3 chiều) là vô cùng quan trọng, vì các đối tượng 3D giúp cho thế giới trong thực tại ảo giống với thực tế hơn đáp ứng được các nhu cầu khắt khe của con người.

Các mô hình đối tượng 3D trong thực tại ảo được tạo ra chủ yếu bằng ba phương pháp đó là tạo ra từ các lệnh trong ngôn ngữ lập trình, từ các nhà thiết kế sử dụng phần mềm 3D và từ các máy quét 3D. Cùng với sự phát triển của khoa học công nghệ những chiếc máy quét 3D đang dần trở thành công cụ đắc lực cho việc tạo ra các mô hình 3D từ thế giới thực. Các sản phẩm được tạo từ máy quét 3D có tỷ lệ chính xác so với mẫu ban đầu khá cao, đồng thời cũng giảm khá nhiều thời gian và chi phí để tạo ra đối tượng 3D. Tuy nhiên, đồng nghĩa với độ chính xác cao, các mô hình này cũng có số lượng lưới khá lớn, khó có thể dùng cho nhiều ứng dụng khác nhau của thực tại ảo.

Bài toán tối ưu hóa bề mặt lưới trên mô hình 3D có nhiều ý nghĩa khi mà số lượng mô hình cần đưa vào thực tại ảo ngày một nhiều, ví dụ như chúng ta mô phỏng lại một thành phố, hay tái tạo lại một khu bảo tàng. Bên cạnh đó, sản phẩm về thực tại ảo xuất hiện nhiều trên các điện thoại thông minh hay máy tính bảng, các thiết bị này hiện tại thì cấu hình phần cứng đang còn khá khiêm tốn. Với khả năng ứng dụng cao, chi phí thấp thì cần phải có những nghiên cứu chuyên sâu về tối ưu lưới mô hình để các mô hình 3D sau được tạo từ máy quét có thể ứng dụng được rộng rãi hơn.

Chính vì tầm quan trọng của bài toán tối ưu hóa mô hình với số lượng lưới lớn, đặc biệt là các mô hình tạo ra từ máy quét 3D, tôi thực hiện đề tài “**Mô hình 3D và tối ưu hóa mô hình trong thực tại ảo**” nhằm đáp ứng các yêu tố về kích thước dữ liệu mô hình đối tượng đồng thời tạo lưới mới cho bề mặt mô hình để thỏa mãn cho các chuyển động của mô hình.

Nội dung luận văn được chia làm 3 phần chính: Chương 1 là trình bày tổng quan về thực tại ảo và các cách khác nhau tạo ra các đối tượng 3D. Chương 2 là hệ thống hóa một số giải pháp tối ưu hóa mô hình sao cho các mô hình thu được đáp ứng đủ tiêu chuẩn của người sử dụng. Chương 3 là trình bày thực nghiệm của bài toán tối ưu hóa lưới mô hình. Phần cuối cùng là kết luận và hướng phát triển tiếp theo của luận văn.

## CHƯƠNG 1. THỰC TẠI ẢO VÀ BÀI TOÁN TỐI ƯU MÔ HÌNH

### 1.1. Khái quát về thực tại ảo và mô hình 3D trong thực tại ảo[2]

Theo cách thức thông thường, người sử dụng tương tác với máy tính thông qua các thiết bị đầu vào như bàn phím, chuột, v.v. và các thiết bị đầu ra như màn hình, loa v.v.. Hệ thống Thực tại ảo (Virtual Reality-VR) ra đời cho phép người sử dụng tương tác với máy tính theo một phương thức tích cực hơn, cao hơn.

#### 1.1.1. Thực tại ảo

Thực tại ảo [2] là công nghệ sử dụng các kỹ thuật mô hình hoá không gian ba chiều, đưa thế giới ba chiều vào trong máy tính để tạo ra một môi trường ảo (Virtual Environment) bằng 3D. Trong môi trường ảo, người sử dụng đã thực sự trở thành một phần của hệ thống. Một trong các ứng dụng, con người có thể được nhập vai để có thể tự do chuyển động trong không gian ảo, tương tác với các vật thể ảo. Ngược lại, môi trường ảo tác động lại hay có những phản hồi tương ứng với các hành động của người sử dụng, các tác động này tuân theo những quy tắc toán học, vật lý, .. tự nhiên, làm con người có cảm giác như đang tồn tại trong một thế giới thực.

#### Các lĩnh vực ứng dụng của Thực tại ảo

Công nghệ Thực tại ảo đang ngày một phát triển rộng rãi và đã có mặt trong hầu hết các lĩnh vực quan trọng của cuộc sống.

##### ❖ Kiến trúc, xây dựng và công nghiệp chế tạo

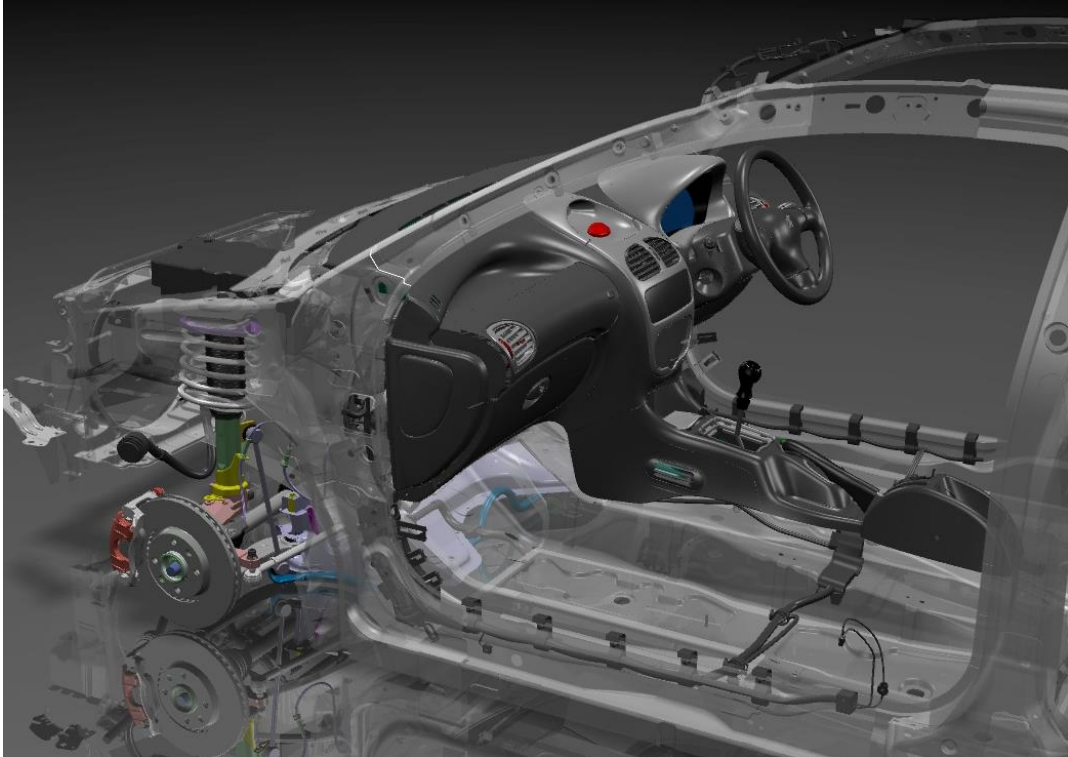


*Hình 1.1. Ứng dụng Thực tại ảo trong thiết kế nội thất*

Thiết kế kiến trúc là một trong những lĩnh vực ứng dụng công nghệ thực tại ảo nhiều nhất. Trước đây, khi chưa có thực tại ảo, các ý tưởng công trình kiến trúc chỉ được thể hiện trên các khổ giấy. Và chi tiết bằng cách thêm thông số và bản vẽ các

mặt của công trình. Ngày nay, khả năng mô hình hoá thể giới thực của công nghệ thực tại ảo dường như đáp ứng một cách đầy đủ, trực quan các công trình của ngành thiết kế kiến trúc từ không gian 3D, kết cấu công trình, vật liệu, ánh sáng,... cho phép khách hàng, nhà đầu tư tự do tham quan, khảo sát công trình cần xây dựng của họ theo nhiều góc độ và vị trí khác nhau.

Bên cạnh kiến trúc, xây dựng công nghệ thực tại ảo hỗ trợ đắc lực cho ngành sản xuất thiết bị cơ khí, mà công đoạn thiết kế mô hình có vai trò quan trọng khi thiết kế động cơ, thiết kế ô tô, tàu biển, máy bay,..



*Hình 1.2. Ứng dụng Thực tại ảo trong thiết kế xe hơi*

Khả năng mô hình hoá cho phép nhà thiết kế thể hiện được một cách trực quan nhất ý tưởng thiết kế của mình, qua đó có thể đánh giá cơ bản về hiệu năng của thiết bị dựa trên những thử nghiệm mô phỏng trên thiết bị ảo, và có những hiệu chỉnh cần thiết trước khi thiết bị thực sự được sản xuất. Điều này rõ ràng góp phần không nhỏ trong thành công của thiết bị công nghệ, giảm bớt những chi phí phát sinh.

#### ❖ **Giải trí**

Khi công nghệ thực tại ảo ra đời, con người luôn luôn nghĩ ra những thứ mới để đầu tư cho lĩnh vực giải trí. Việc áp dụng công nghệ 3D khiến chi phí đầu tư vào lĩnh vực phim, game, ... khá thấp mà lợi nhuận thu vào là vô cùng to lớn.

Số lượng người bị cuốn hút theo các trò chơi game, đặc biệt là giới trẻ, tăng theo cấp số nhân,



*Hình 1.3. Ứng dụng Thực tại ảo trong game giải trí*

số lượng vé bán ra trong các rạp chiếu phim 3D làm vô cùng lớn từ năm 2007 trở về đây, ví dụ như phim Avatar, Transfomer,..



*Hình 1.4. Ứng dụng Thực tại ảo trong phim Avatar*

Hơn thế, sự phát triển mạnh mẽ trong lĩnh vực công nghệ, giúp phần cứng máy tính càng ngày càng đáp ứng được các nhu cầu của của các nhà sản xuất cũng như các tín đồ giải trí. Nếu như chúng ta trở lại khoảng hơn 15 năm về trước, thật khó có thể

tìm thấy một máy tính có cấu hình đủ mạnh để cho phép tạo ra các ứng dụng thực tại ảo thời gian thực.

#### ❖ **Giáo dục và Đào tạo**

Ngay từ khi công nghệ 3D ra đời, thì hầu hết các ứng dụng thực tại ảo đều được phát triển trong quân đội. Sự đầu tư vô cùng lớn từ phía các nhà lãnh đạo của Mỹ, Nga, .. là việc tập luyện bắn ảo, các bài toán mô phỏng cháy nổ của thuốc súng, hay mô phỏng đường đi của tên lửa, ...

Ngày nay, sự phát triển trên nền công nghệ và kỹ thuật cao, thực tại ảo tích hợp những đặc tính làm cho bản thân nó có những tiềm năng vượt trội so với các công nghệ đa phương tiện truyền thống khác. Ví dụ: lái xe đạp tương tác ảo cho trẻ em, tương tác lái container, lái máy bay ảo. Qua các thiết bị phần cứng, như màn hình, joytick, kính, người học có thể nhập vai để tương tác với thực tế ảo thông qua các hành động của mình, đồng thời thế giới ảo cũng tác động lại với người học, khiến người học tăng thêm các kỹ năng nghiệp vụ mà không cần phải chịu chi phí, hay hậu quả do mình gây ra. Ví dụ một vụ lái xe container mà đâm vào một tòa nhà ven đường,...



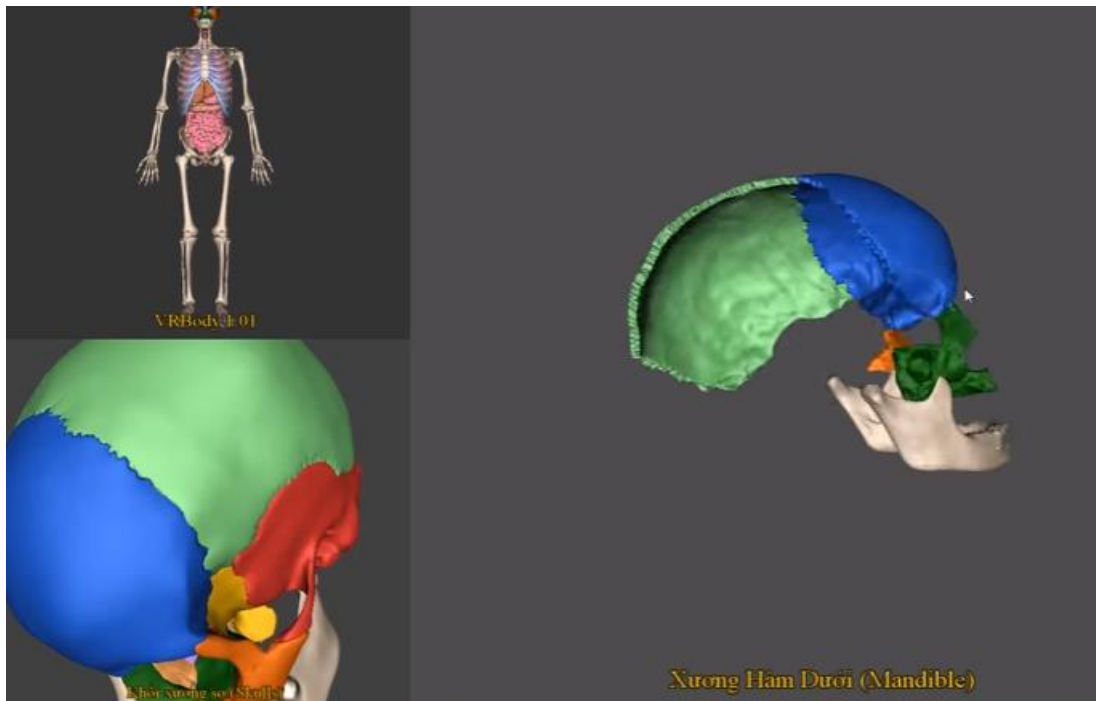
*Hình 1.5. Mô phỏng lái tàu ảo của công ty mô phỏng Việt Nam*

Khi công nghệ thực tại ảo tính toán chính xác những vụ nổ, vụ va chạm, tính toán khác trong xã hội thì ngày càng nhiều các thí nghiệm được thực hiện ngay trên môi trường ảo.

Tính chất trực quan của bài giảng thực tại ảo được nâng cao một bước làm tăng sự hứng thú trong học tập cũng như khả năng ghi nhớ các khái niệm quan trọng trong bài giảng. Ví dụ chúng ta hoàn toàn có thể tạo ra một môi trường ảo của các trận chiến trong môn học lịch sử mà người học có thể nhìn cuộc chiến từ nhiều góc độ khác nhau, có thể là người trung gian, có thể là hóa thân một nhân vật để tham gia trận chiến đó. Từ đó, học viên nắm bắt được nhanh chóng và có ý thức hơn với những tính huống được học. Và cũng không phải là viễn tưởng khi ta có thể nói rằng một ngày nào đó bài học của học viên sẽ là những kỹ năng sống được đào tạo trong môi trường ảo.

#### ❖ Y học

Trong y học, công nghệ thực tại ảo giúp cho con người có thể thao tác giải phẫu trực tiếp với các thể ảo. Giúp cho việc đào tạo các bác sỹ đa khoa được hoàn thiện hơn và tự tin hơn trong các ca mổ của mình. Giúp cho công nghệ y tế được phát triển hơn, qua mô phỏng giúp cho con người hiểu hơn về quá trình truyền máu, tiêu hóa thức ăn,.. từ các bài toán mô phỏng.



Hình 1.6. Mô phỏng cơ thể ảo của phòng thực tại ảo viện khoa học công nghệ Việt Nam

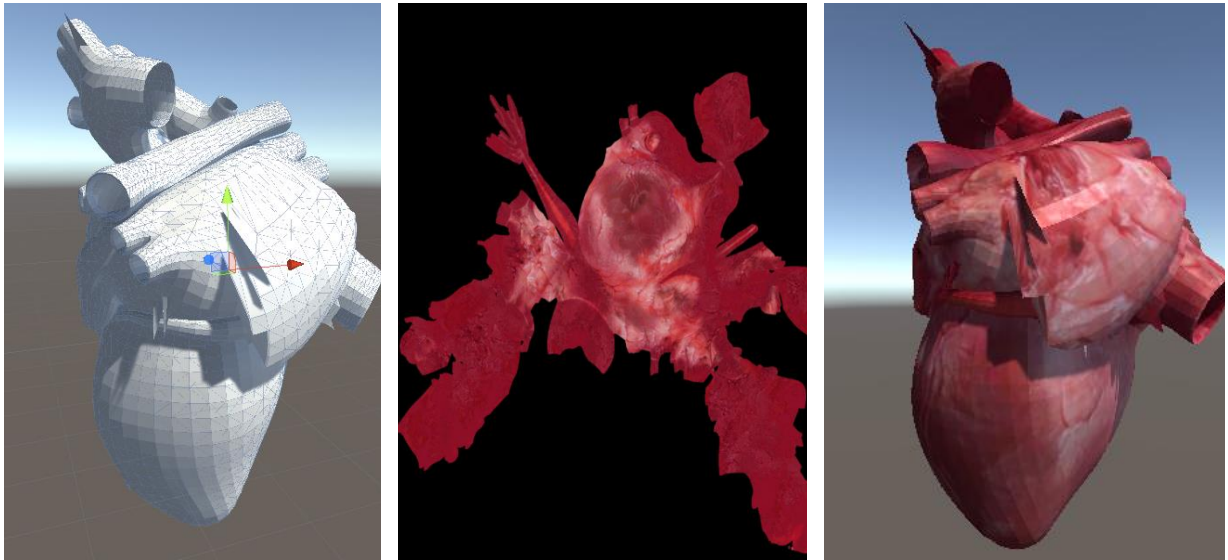
Y học là một trong những lĩnh vực ứng dụng tiềm năng trong công nghệ Thực tại ảo. Cho đến nay, lĩnh vực nổi bật trong y học áp dụng thành công công nghệ Thực tại ảo là giả lập giải phẫu (Surgical Simulation).

Trên cơ sở các kỹ thuật đồ hoạ máy tính và Thực tại ảo, hệ thống đào tạo y học này bao gồm hai bộ phận cơ bản: Khối tương tác ba chiều là mô hình sinh thể ảo cho phép người sử dụng thực hiện các thao tác giải phẫu thông qua các dụng cụ giải phẫu ảo; Khối giao diện người dùng hai chiều cung cấp những thông tin phản hồi trực quan từ mô hình trong quá trình giải phẫu cũng như những thông tin hướng dẫn trong phiên đào tạo.

Phương pháp đào tạo có tính tương tác cao này mang nhiều ưu điểm so với các phương pháp truyền thống như thực hành trên mô hình plastic hay trên bệnh nhân thực. Thứ nhất, khác với phương pháp dùng mô hình plastic, sinh thể giải phẫu ảo có khả năng cung cấp những thông tin phản hồi sinh học một cách tự nhiên như một sinh thể sống thực, chẳng hạn như sự thay đổi về nhịp tim, huyết áp v.v. Điều này tạo cho học viên có cảm giác đang trải qua một ca mổ trong một tình huống thực. Thứ hai, khác với thực hành trên bệnh nhân thật, những sai lầm của học viên trong quá trình thực tập không phải trả giá bằng những thương tổn thực trên cơ thể người bệnh. Điều này cũng làm giảm áp lực lên học viên khi thực hiện phẫu thuật ảo. Từ đó, giúp họ tự tin và chủ động hơn trong học tập. Phương pháp này còn cho phép các bác sĩ không ngừng nâng cao trình độ tay nghề, kỹ năng phối hợp làm việc bằng cách liên tục đặt ra những giả định tình huống bệnh, cập nhật những dữ liệu bệnh lý mới để thực hiện những phương pháp mới, kỹ thuật mới trong điều trị. Bác sĩ cũng có thể tự lập kế hoạch mổ thử trên bệnh nhân ảo trước khi mổ trên bệnh nhân thật do đó làm tăng mức độ an toàn và hiệu quả điều trị, giảm thiểu sai lầm rủi ro đáng tiếc xảy ra.

### **1.1.2. Cấu tạo mô hình 3D**

Mô hình 3D [4] là một cấu trúc dữ liệu trong đó mô tả hình thái 3D của một đối tượng. Hiện nay để tạo ra một mô hình 3D có nhiều cách khác nhau, chúng có thể được tạo ra nhờ các phần mềm thiết kế 3D như 3Ds max, maya v.v.. thông qua các nhà thiết kế 3D, hoặc từ các máy quét 3D (khi đó một đối tượng ngoài thế giới thực sẽ tạo ra một được một mô hình 3D trên máy tính thông qua máy quét), hoặc được tạo ra bằng một vài cách đặc thù nào đó. Ví dụ như được tạo ra từ việc tối ưu một mô hình khác như trong luận văn đang trình bày, hoặc lập trình để tạo ra mô hình v.v.. Để có thể tạo ra một mô hình 3D đầu tiên chúng ta phải hiểu về cấu trúc của một mô hình 3D. Theo những tài liệu tôi tìm hiểu được, một mô hình gồm có 3 thành phần cơ bản là tập các đỉnh, tập các mặt và tập UV. Trong đó, tập UV thường kết hợp với một ảnh chất liệu bên ngoài để tạo ra hình ảnh của mô hình với bề mặt giống với thực tế.



(a)

(b)

(c)

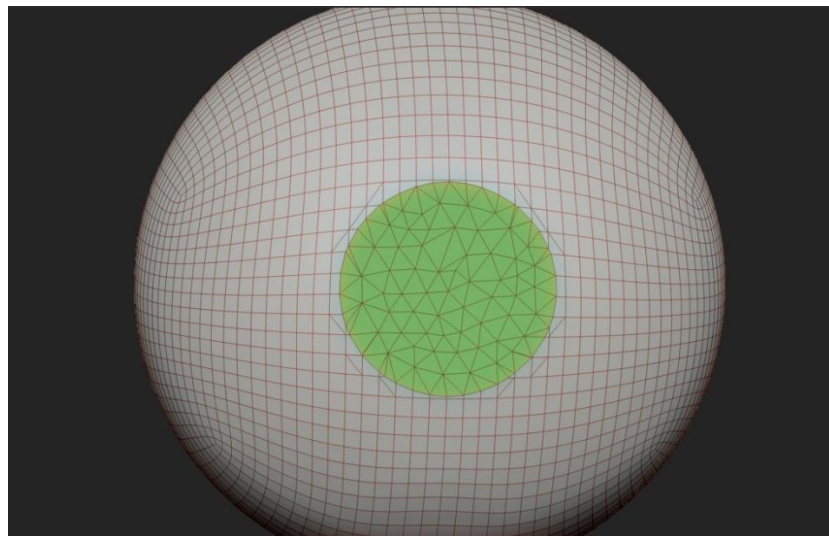
*Hình 1.7: Mô hình 3D tim người*

*(a) Mô hình 3D chỉ gồm tập các đỉnh và tập các mặt*

*(b) Texture đã trải UV cho mô hình quả tim*

*(c) Hình ảnh quả tim 3D khi có đầy đủ các thành phần*

Trong mô hình 3D, tập đỉnh là tập các vector 3 chiều mà mỗi vector là một điểm trong không gian 3 chiều. Tập đỉnh này sẽ quy định hình dạng 3D của đối tượng, tiếp đó chúng ta cần tập các mặt để kết nối các đỉnh với nhau từ đó tạo ra bề mặt của đối tượng. Tập các đỉnh và tập các mặt về cơ bản tạo ra một mô hình 3D giống với một bức tượng được đan bởi lưới sắt rỗng bên trong. Chúng tạo ra một hình dạng giống một lưới dựa trên quan hệ giữa các đỉnh và các mặt.



*Hình 1.8. Mô hình 3D hình cầu cắt vát*



Để mô hình giống thật hơn ta xác định một texture và một tập UV để quy định việc sử dụng texture trên mỗi bề mặt của đối tượng. Như vậy để xác định một mô hình 3D thường chúng ta phải xác định 3 thành phần của nó là tập các đỉnh, tập các mặt và tập UV. Để dễ hình dung, tôi lấy ví dụ về một đối tượng 3D cơ bản là một khối hộp được tạo ra dựa trên dạng lưới tam giác như sau:

### **-Tập các đỉnh**

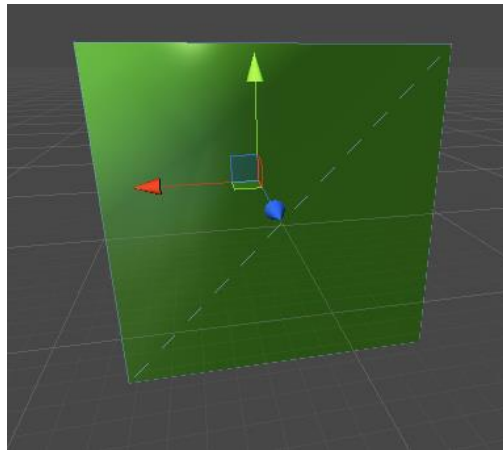
```
var size = 100;
Vector3 [] VertexList = new Vector3 []{
    new Vector3(-size, -size, -size),
    new Vector3(-size, size, -size),
    new Vector3( size, size, -size),
    new Vector3( size, -size, -size),
    new Vector3( size, -size, size),
    new Vector3( size, size, size),
    new Vector3(-size, size, size),
    new Vector3(-size, -size, size)
};
```

Ở đây tôi xác định một khối hộp do đó tôi cần tối thiểu 8 đỉnh, vị trí các đỉnh được sắp xếp trong không gian tương ứng với tập vector 3 chiều VertexList được tạo ở trên.

### **-Tập các mặt**

```
int []FaceList = new int []{
    0, 1, 3, // 1: Mặt sau
    1, 2, 3,
    3, 2, 5, // 2: Mặt trước
    3, 5, 4,
    5, 2, 1, // 3: Mặt trên
    5, 1, 6,
    3, 4, 7, // 4: Mặt dưới
    3, 7, 0,
    0, 7, 6, // 5: Mặt trái
    0, 6, 1,
    4, 5, 6, // 6: Mặt phải
    4, 6, 7
};
```

Với một khối hộp tôi cần xác định 6 mặt tương ứng là trước, sau, trái, phải, trên, dưới. Ở đây tôi sử dụng các mặt ở dạng tam giác, tức là một mặt được tạo ra từ 3 đỉnh. Trong mô hình 3D có 2 dạng mặt cơ bản là mặt được tạo ra từ 3 đỉnh (mặt tam giác) và mặt được tạo ra từ 4 đỉnh (mặt tứ giác) trong nội dung luận văn do mô hình 3D được tái cấu trúc chỉ được sử dụng để quan sát do đó tôi lựa chọn việc tái cấu trúc mô hình sử dụng mặt tam giác. Chi tiết việc xây dựng các mặt được trình bày ở phần tiếp theo của luận văn. Như vậy với 6 mặt của hình hộp tôi cần 12 mặt tam giác, giá trị và quan hệ của các mặt được mô tả trong tập mặt FaceList ở trên.



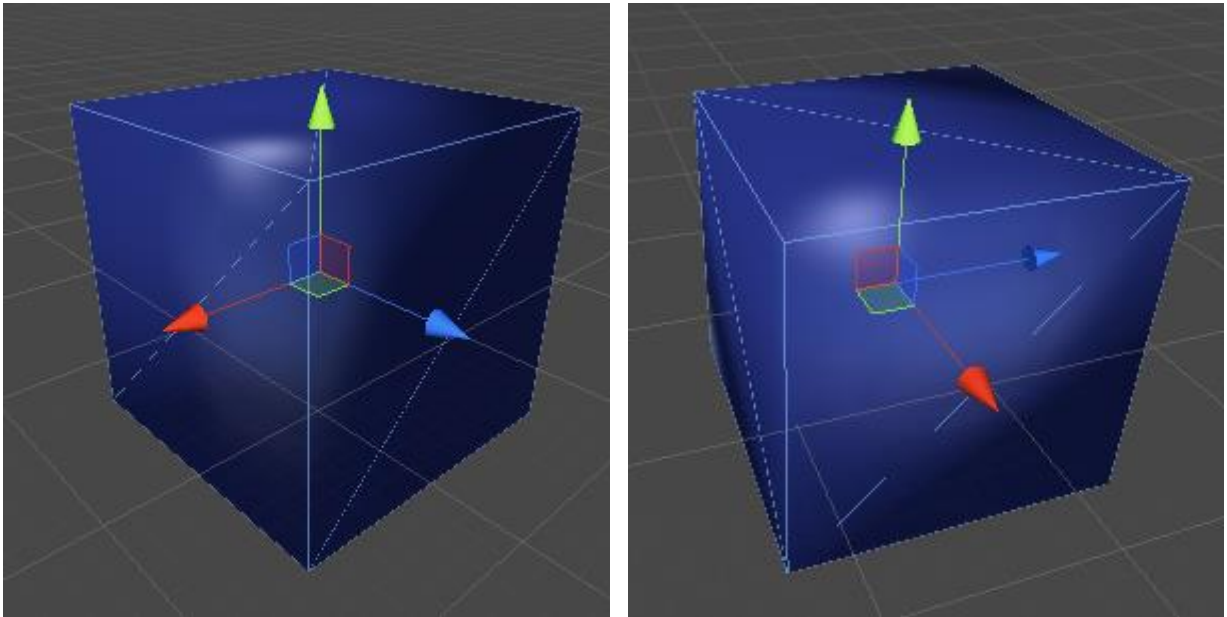
*Hình 1.9. Một mặt của hình hộp được tạo bởi 2 mặt tam giác*

**-Tập UV:**

```
Vector2 [] UVs = new Vector2[] {
    new Vector2(0,0),
    new Vector2(0,0),
    new Vector2(0,0),
    new Vector2(0,0),
    new Vector2(0,0),
    new Vector2(0,0),
    new Vector2(0,0),
    new Vector2(0,0),
};
```

Tập UV xác định vị trí của các texture, đó là một tập của các vector 2 chiều và số lượng phần tử của tập này tương ứng với số đỉnh của đồ thị. Trong nội dung luận văn vì mô hình tái cấu trúc không thu được texture từ ảnh cắt lớp nên tôi không xây dựng tập UV cho các mô hình 3D và khởi tạo mặc định là một vector 2 chiều có 2 giá trị tương ứng bằng 0. Sau khi xác định được 3 thành phần cơ bản của một mô hình 3D

mà cụ thể ở đây là một khối hộp 3D ta thu được kết quả là một mô hình 3 chiều như hình bên dưới



Hình 1.10. Mô hình 3D thu được từ tập đỉnh: *VertexList* , tập mặt: *FaceList* ,  
tập UV: *UVs*

### 1.1.3. Các phương pháp tạo mô hình phổ biến hiện nay

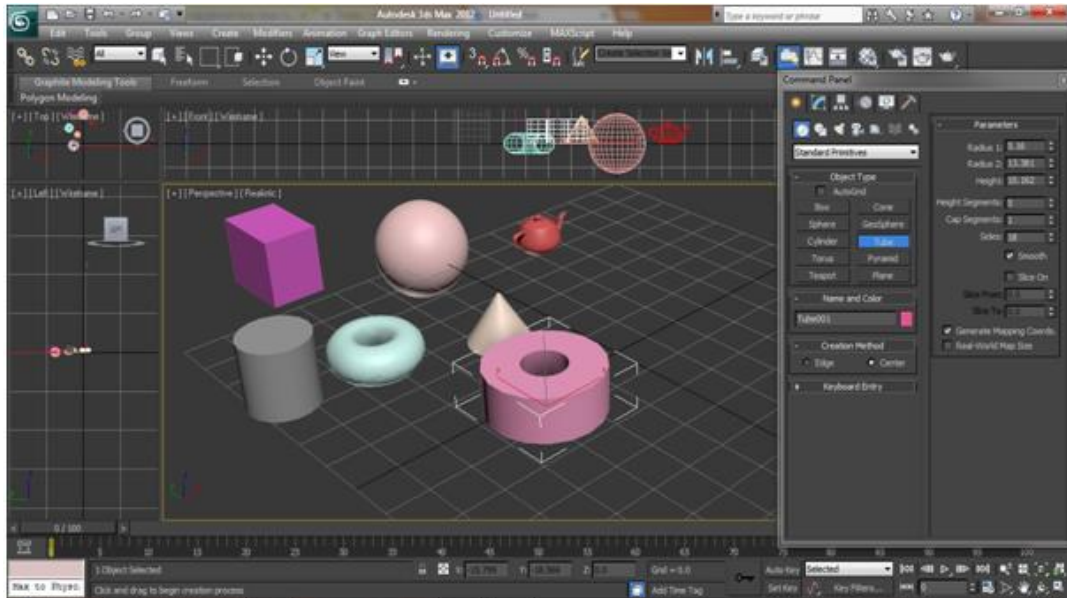
#### 1.1.3.1. Phương pháp tạo mô hình bằng thiết kế dựa trên phần mềm 3D

Tùy theo kĩ thuật và kinh nghiệm làm việc mỗi cá nhân có thể đưa cho mình những phương pháp khác nhau để đi vào thiết kế, hay xây dựng một sản phẩm. Cách mà người ta làm ra nó sẽ ảnh hưởng bởi nhân tố thời gian và lượng chất xám phải đầu tư vào. Để tạo ra cùng 1 sản phẩm sẽ có nhiều con đường cho ta lựa chọn, đi bằng cách nào cho hợp lí, tối ưu nhất, tiết kiệm thời gian và công sức nhất. Tôi xin trình bày 3 phương pháp dưới đây:

##### **A- Phương pháp thiết kế đi từ tổng thể đến chi tiết**

Các vật thể hữu hình trong cuộc sống của chúng ta hầu hết được cấu tạo nên từ những hình khối cơ bản, cũng như vậy trong phần mềm mô phỏng 3DS Max đã cung cấp cho chúng ta các hình khối đó để thể hiện các đối tượng trong không gian 3 chiều: khối cầu, khối trụ, khối hộp...

Để vẽ được các dạng khối cơ bản này trên bảng lệnh **Command Panel** chọn **Creat** sau đó chọn **Geometry** (dạng hình học) phần mềm sẽ xổ xuống cho ta một danh sách các Object Type : Box, Sphere, Cylinder, Torus, Teapot, Cone, Tube, Plane v.v...



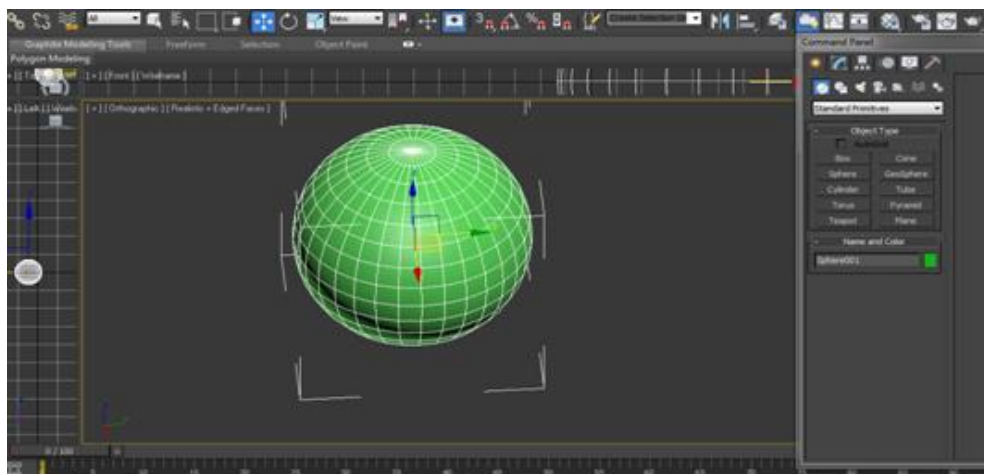
Hình 1.11. Một số hình khối 3D cơ bản

Bằng việc quan sát vật thể cần mô phỏng hay thiết kế, người thực hiện rút ra cho mình những nhận xét, cấu tạo của vật thể, như mô phỏng chiếc nón lá Việt Nam- nó có dạng hình chóp nhọn thì tại sao ta không dùng luôn chức năng Cone(chóp nhọn) để vẽ, sẽ rất nhanh chóng, tuy nhiên nhiều vật thể khác có thể không đơn giản như vậy, người ta có thể phải sử dụng kết hợp nhiều hình khối, lấy phần giao, phần bù để tạo được vật thể như ý muốn.

Từ các hình khối cơ bản đã tạo được như trên, 3DS Max cũng cho chúng ta công cụ để hiệu chỉnh nó. Có 3 công cụ rất hữu ích sau:

#### **Select and Move** (công cụ di chuyển)

Dùng để chọn và di chuyển đối tượng theo trục x, y hoặc z. Khi đã chọn được đối tượng thì giữ trái chuột và rê chuột tới vị trí mới theo trục x, y, z hoặc theo cả 3 hướng. Nếu muốn chính xác thì cần phải gọi hộp thoại Move Transform Type - In sau đó nhập các giá trị tương ứng. Để hiện hộp thoại này thì click chuột phải ngay trên nút Select and Move.



Hình 1.12. Công cụ Select and Move trong thiết kế

### Select and Rotate (công cụ xoay)

Dùng để chọn và xoay đối tượng, thao tác tương tự như với Select and Move. Nếu muốn quay đối tượng với độ chính xác cao thì cần phải gọi hộp thoại Rotate Transform Type - In bằng cách click phải chuột ngay trên nút Rotate and Move sau đó nhập vào giá trị tương ứng.

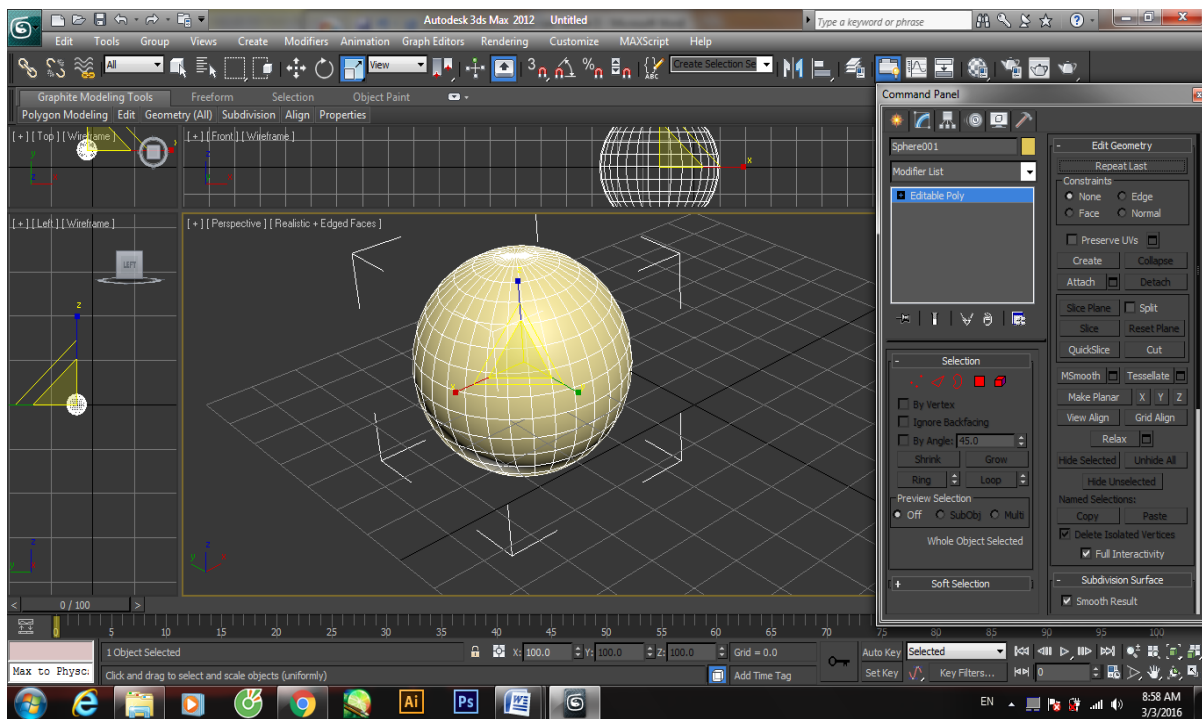
### Select and Uniform Scale (công cụ thu, phóng)

Dùng để chọn và thay đổi kích thước đều trên bề mặt của đối tượng hoặc thay đổi kích thước không đều trên bề mặt của đối tượng.

### Ngoài ra còn có chế độ hiệu chỉnh nâng cao **Editable Poly**

Đây là cách thường dùng để tạo hình trong 3ds max, ngoài Editable Poly bạn còn gặp Editable Mesh, Editable Patch, NURBS. Để sử dụng được chế độ này bạn cần phải chuyển đổi những hình khối cơ bản trở thành những khối có khả năng chỉnh sửa được.

Cách làm: bạn chọn vật thể cần chuyển đổi, click phải chuột chọn **Convert To** -> chọn chế độ cần chuyển đổi, ở đây tôi nói đến chế độ **Editable Poly**, phần mềm sẽ hiện giao diện như sau:



Hình 1.13. Chế độ Editable Poly

Ở đây có các chế độ hiệu chỉnh cạnh, đường, điểm, mặt, và khối. Kích chọn từng chế độ sẽ xuất hiện một bảng chức năng tương ứng nữa.

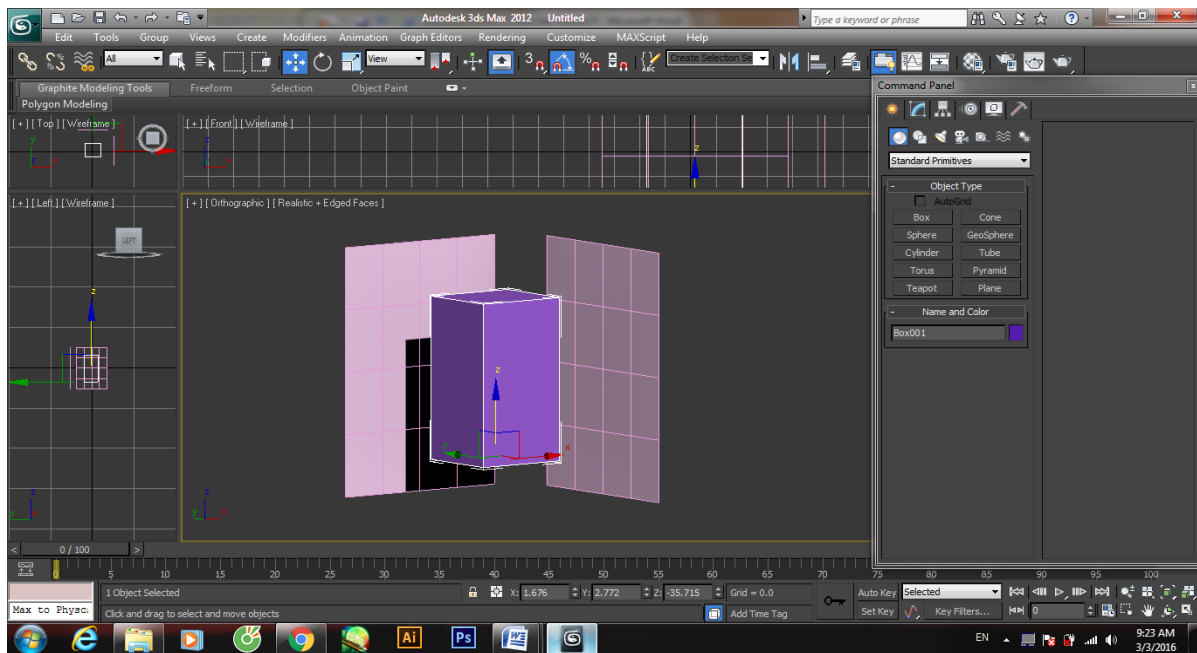
Bằng cách này bạn có thể co kéo, bóp dãn. Khối cơ bản của chúng ta sẽ như một cục đất nặn mà thành hình hay không là do kỹ năng nhào lặn của mỗi người. Tuy nhiên sẽ vô cùng khó khăn trong việc xác định chiều và quan sát trong không gian 3D.

## ***B- Phương pháp thiết kế đi từ chi tiết đến tổng thể***

Ngược lại với phương pháp trên, để bắt tay vào xây dựng vật thể theo phương pháp này ta sẽ phải chuẩn bị ảnh hình chiếu của vật thể thường là hình chiếu mặt trước và hình chiếu mặt trái hoặc hình chiếu mặt phải. Ưu điểm của phương pháp này là độ chính xác cao hơn phương pháp trên, ở trên việc tạo ra vật thể hoàn toàn do khả năng quan sát trong không gian và cảm tính, ước chừng của người thực hiện. Còn ở phương pháp này ta có thể tạo chính xác hoàn toàn chi tiết khi có cơ sở dữ liệu ảnh đầy đủ, tuy nhiên làm theo phương pháp này cần nhiều thời gian và độ tỉ mỉ hơn. Vì thế tùy theo yêu cầu của sản phẩm mà ta chọn cho mình một phương pháp phù hợp.

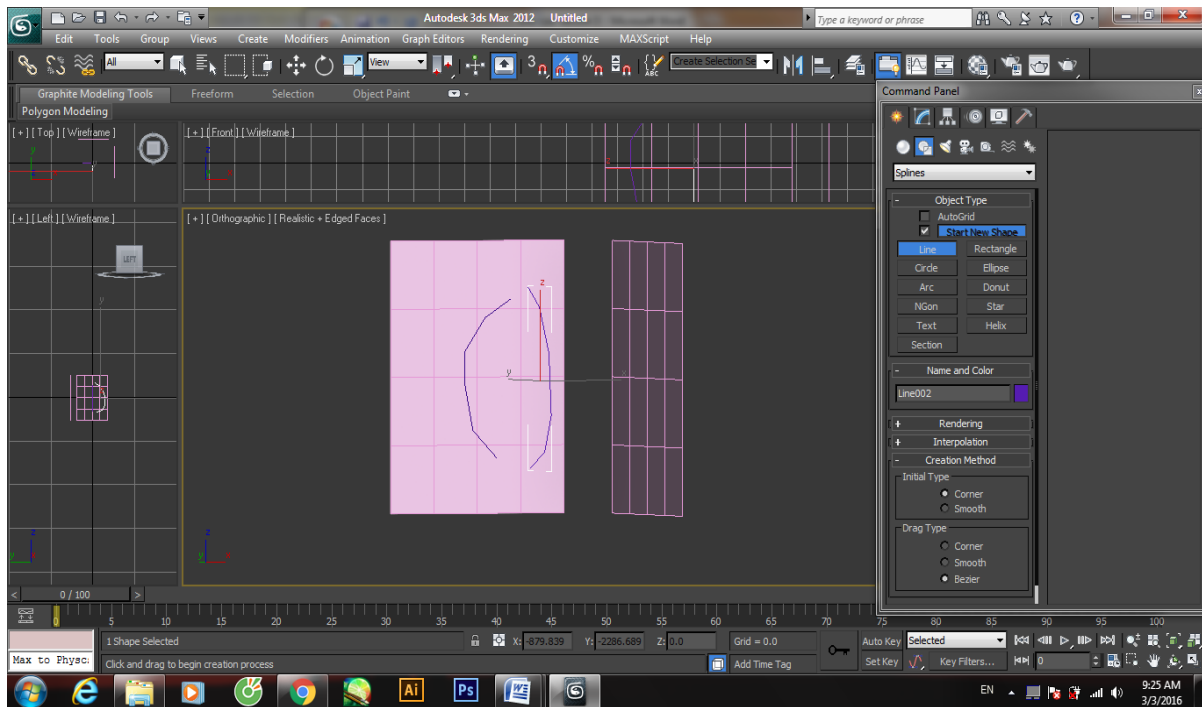
Ở đây cũng chia làm 2 cách như sau:

**Cách 1:** Tạo hai tấm Plane mặt chiếu đứng và chiếu cạnh, tạo hình khối cơ bản để đối chiếu hiệu chỉnh theo.



*Hình 1.14. Hai tấm plane chiếu đứng và cạnh (cách 1)*

**Cách 2:** Cũng tạo hai tấm Plane mặt chiếu đứng và chiếu cạnh, nhưng thay vào đó không tạo hình khối đối chiếu mà dùng đường Line đồ theo 2 mặt chiếu.



Hình 1.15. Hai tấm plance chiều đứng và cạnh (cách 2)

### C- Phương pháp thiết kế phối hợp

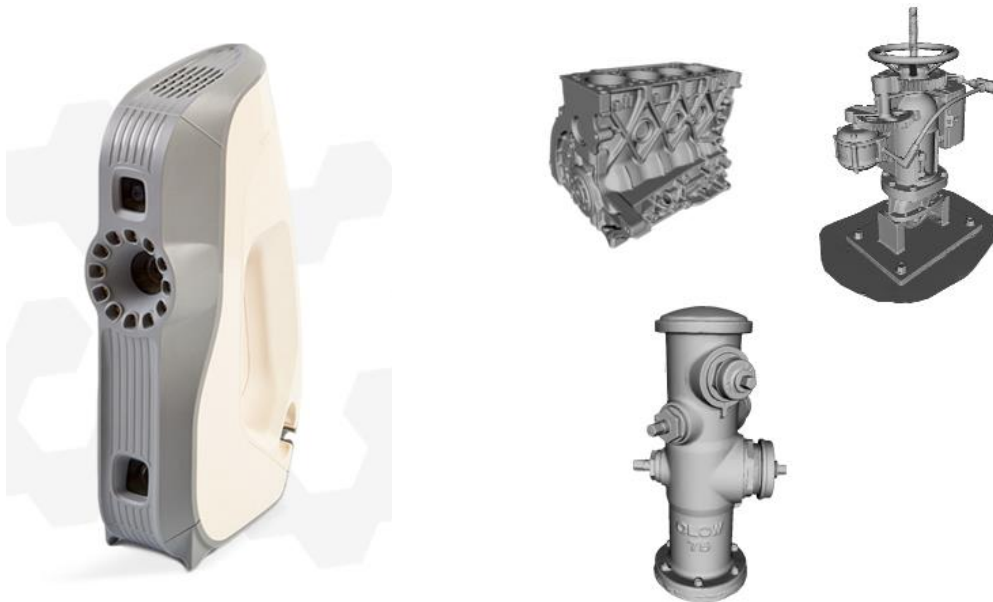
Tận dụng hiệu quả của từng phương pháp, cách tốt nhất cho công việc của chúng ta là sử dụng linh hoạt kết hợp các phương pháp. Vì khách quan mà nói mọi vật thể trong cuộc sống đều tương đối phức tạp, bằng khả năng phân tích tốt cùng với sự linh hoạt công việc thiết kế xây dựng của các nhà mô phỏng sẽ được thực hiện dễ dàng và nhanh chóng, hiệu quả hơn. Một số thành phần ta dùng phương pháp hiệu chỉnh từ khối, một số thành phần thì dùng **Line** vẽ thêm và hiệu chỉnh.

#### 1.1.3.2. Tạo mô hình bằng máy quét 3D

Bên cạnh việc tạo mô hình bằng phương pháp thiết kế sử dụng con người, hiện nay chúng ta có thể tạo ra các mô hình từ các thiết bị phần cứng là máy quét 3D. Có nhiều các thiết bị phần cứng khác nhau hiện đang lưu hành trên thị trường. Dưới đây là một số thiết bị được sử dụng để quét và tạo ra mô hình 3D.

Artec Eva là lựa chọn lý tưởng để tạo một mô hình 3D nhanh, kết cấu chính xác của các đối tượng có kích thước trung bình như một bức tượng người bán thân, một bánh xe hợp kim hoặc một hệ thống ống xả xe máy. Nó quét nhanh với độ phân giải cao, màu sắc sống động cho phép các ứng dụng gần như không giới hạn.

Ánh sáng, sự nhanh nhạy và linh hoạt, đó là máy quét Artec phổ biến nhất và là một dòng máy quét 3D cầm tay dẫn đầu trên thị trường.



(a) Máy quét Artec Eva

(b) Mô hình thu được từ máy quét Artec Eva

Hình 1.16. Máy quét Artec Eva và mô hình thu được khi sử dụng máy quét này

Với việc dễ dàng sử dụng, tốc độ và sự chính xác cao, Artec Eva là sản phẩm thiết yếu cho hàng loạt các ngành công nghiệp. Từ tạo mẫu nhanh cho đến kiểm tra chất lượng, CGI cho đến bảo tồn di sản, ngành công nghiệp ô tô cho đến pháp y, y học và các bộ phận giả cho đến hàng không vũ trụ, Artec Eva được sử dụng để tùy chỉnh, cải tiến và sắp xếp vô số các ngành công nghiệp tư tưởng tiên tiến. Artec Eva thậm chí được sử dụng để quét tổng thống Barack Obama và tạo nên bức chân dung 3D đầu tiên của một tổng thống Mỹ.



Hình 1.17. Tổng thống Mỹ Obama và hình ảnh 3D của ông thu được từ máy quét.

TTO - Sense 3D Scanner là loại máy quét vật thể 3 chiều cầm tay được công ty 3D System giới thiệu ngày 8-11 với mức giá bình dân 399 USD.





*Hình 1.18. Máy quét TTO - Sense 3D*

Sense 3D Scanner nhỏ gọn, vận hành tự động kèm theo phần mềm hiệu chỉnh, giúp người dùng tạo ra các hình mẫu 3 chiều (3D) của vật thể bằng cách quét bất cứ cái gì kể cả con người mà không tốn nhiều thời gian lẫn kỹ thuật đồ họa trên máy tính.

Khi muốn quét một vật thể, dù là nhỏ, vừa hay lớn, người dùng đi vòng quanh quét vật thể qua Sense. Ảnh 3 chiều vật thể sẽ được thiết bị tạo dựng, có thể dùng để in 3D.

Máy còn có chức năng nhận diện vật thể, các đối tượng trong một khung cảnh tổng thể được chọn lọc ra và nhận dạng. Sense 3D Scanner nhỏ và rẻ hơn sản phẩm Digitizer của MakerBot ra mắt trước đó. Tuy nhiên, Sense 3D Scanner không thực sự di động, vì nó vẫn cần kết nối đến máy tính để truyền tải dữ liệu, và chất lượng ảnh quét vật thể không được công bố cụ thể.

Trước đó trong tháng 10, công ty hàng đầu trong lĩnh vực công nghệ in 3D MakerBot đã phát hành ra thị trường máy quét 3D (scanner) Digitizer với giá tham khảo 1.500 USD. Theo MakerBot, Digitizer có đối tượng khách hàng rất rộng, từ kiến trúc sư, nhà thiết kế, giáo dục...



*Hình 1.19. Máy quét Digitizer*

"Digitizer có thể chụp đủ các điểm để tạo ra khoảng 200.000 tam giác cho mỗi mô hình 3D mới, chụp các chi tiết nhỏ 0,5mm, các bề mặt có độ sâu 0,5mm..."

## **1.2. Bài toán tối ưu hóa mô hình 3D**

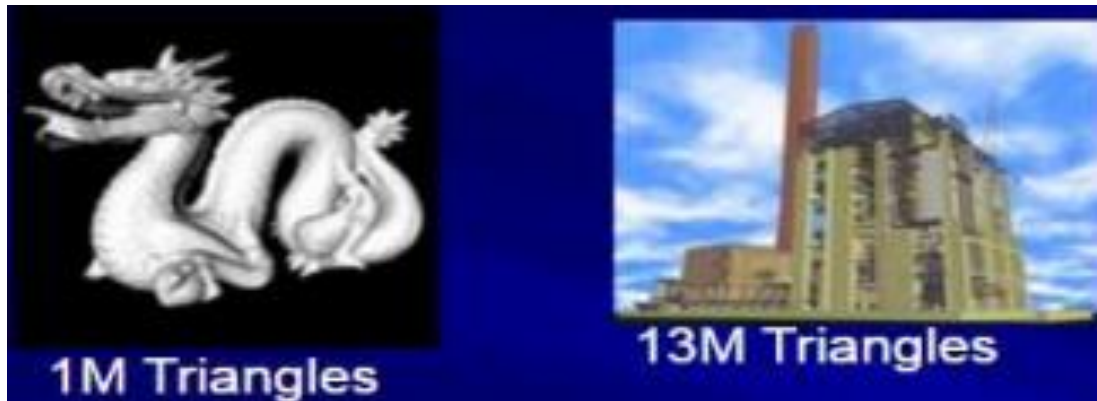
### **1.2.1. Một số phương pháp tạo mô hình 3D**

Như đã trình bày ở trên, có ba phương pháp chính để tạo ra mô hình 3D. Một là sử dụng các lệnh trong ngôn ngữ lập trình để vẽ ra các mô hình đơn giản. Hai là sử dụng các phần mềm thiết kế 3D tạo ra mô hình từ các nhà thiết kế. Ba là sử dụng các thiết bị máy quét 3D tạo mô hình từ vật thể thực. Với mỗi phương pháp tạo mô hình đều mang những ưu, nhược điểm khác nhau. Phương pháp đầu khá hạn chế về chất lượng mô hình và đòi hỏi phải tính toán nhiều khi lập trình để tạo ra các đối tượng 3D.

Với phương pháp thủ công thứ hai là sử dụng con người để thiết kế có khả năng tạo ra mô hình 3D với khả năng tối ưu về lưới, tuy nhiên độ tối ưu của mô hình phụ thuộc vào kiến thức và kinh nghiệm của từng nhà thiết kế. Phương pháp này cũng đòi hỏi chi phí về con người, trong đó thời gian để tạo ra một sản phẩm là tương đối dài, độ đồng đều và ổn định không cao do phụ thuộc vào con người, chi phí cho mỗi một mô hình là lớn. Bên cạnh đó khả năng chính xác là một vấn đề khi sử dụng phương pháp này.

Phương pháp sử dụng các thiết bị phần cứng là máy quét để tạo mô hình 3D mang nhiều ưu điểm: thời gian tạo ra một mô hình ngắn, độ chính xác cao, tính ổn định, chi phí rẻ v.v.. Tuy nhiên, mô hình tạo ra từ máy quét có một nhược điểm lớn chính là số lượng lưới lớn (số các đỉnh và số các mặt lớn), điều đó ảnh hưởng rất nhiều tới việc sử dụng mô hình từ máy quét trong thực tế. Với một chương trình mô phỏng trong thực tại ảo cần sử dụng hàng trăm, hàng nghìn mô hình thậm chí lớn hơn thì việc

một mô hình không tối ưu sẽ ảnh hưởng tới quá trình lưu trữ, tính toán để vẽ mô hình ra các thiết bị hiển thị, tính toán va chạm v.v.. Do đó, trên thực tế đa phần các chương trình mô phỏng và thực tại ảo cũng như lưu trữ không thể sử dụng mô hình từ máy quét 3D.



Hình 1.20. Các mô hình được tạo ra từ máy quét có số lượng lưới cực lớn

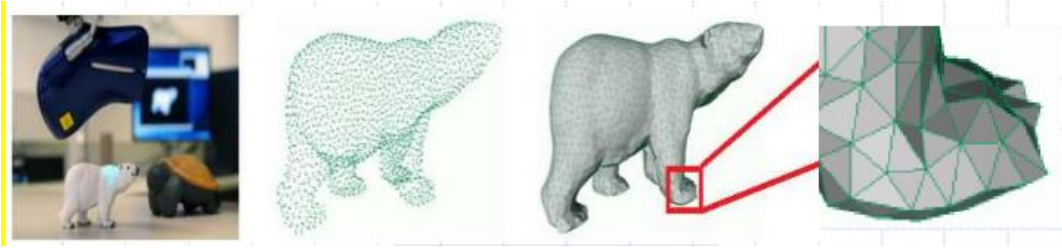
### 1.2.2. Đầu vào , đầu ra bài toán tối ưu hóa mô hình

Có hai dạng bài toán tối ưu mô hình 3D thường được nhắc đến trong lĩnh vực mô phỏng 3D với đầu vào và đầu ra cùng là mô hình 3D nhưng mang những đặc điểm khác nhau giữa mô hình trước tối ưu và sau tối ưu.

Thứ nhất, là tối ưu về mặt hình ảnh. Ở đó, với đầu vào là một mô hình 3D đã được thiết kế hoặc thu từ máy quét người xử lý cần nâng cao chất lượng hình ảnh của mô hình. Khi đó chúng ta cần chú ý tới việc tối ưu chất lượng hình ảnh hoặc lưới của mô hình, điều này dẫn tới các bài toán xử lý về ánh sáng, góc cạnh để khi render thu được hình ảnh chân thực nhất có thể. Trên thực tế quá trình tối ưu này dẫn tới một trường phái thiết kế siêu thực. Ở đó những nhà thiết kế có thể thay thế nhân vật thực bằng nhân vật thiết kế ảo.

Thứ hai, là tối số lượng lưới (mặt và đỉnh trong mô hình) với bài toán này đầu vào là một mô hình 3D (thường là mô hình thu được từ máy quét) và đầu ra là mô hình đó với số lượng lưới giảm đi nhưng vẫn đảm bảo hình dạng và hình ảnh của đối tượng không thay đổi nhiều giữa trước và sau tối ưu.

Trong nội dung luận văn tập trung vào giải quyết bài toán thứ hai. Tức là nghiên cứu các kỹ thuật làm giảm lưới mô hình 3D nhưng vẫn đảm bảo giữ được hình dạng và hình ảnh của đối tượng sau khi render.



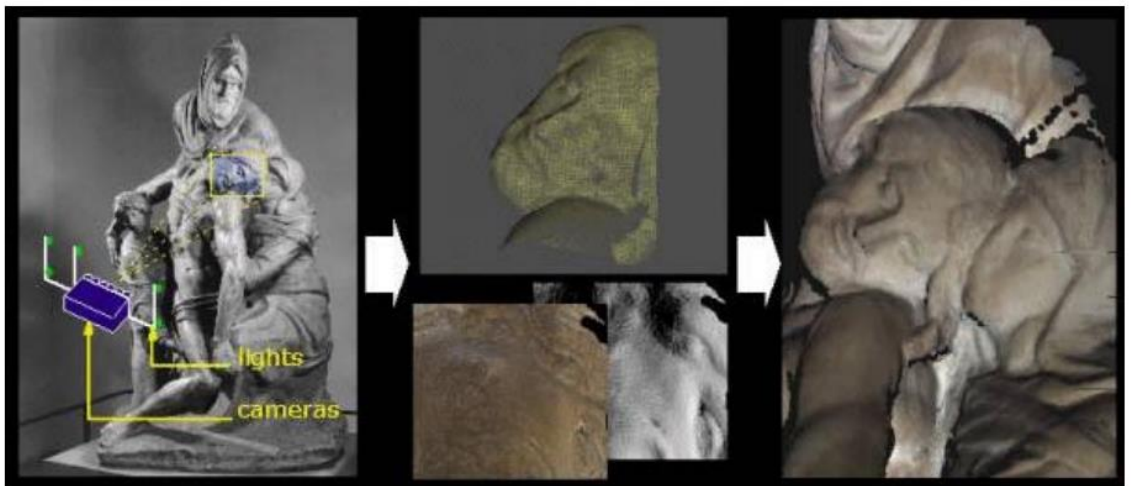
*Thu thập dữ liệu bằng 3D scan*      *Tái tạo bề mặt lưới tam giác*      *Làm mịn bề mặt lưới tam giác*

*Hình 1.21. Thu thập và làm mịn dữ liệu*

Có nhiều nghiên cứu của các nhà khoa học trong và ngoài nước về vấn đề này, đồng thời đã có nhiều công trình được công bố cũng như các kỹ thuật được đề xuất để giải quyết bài toán này. Trong nội dung của luận văn tập trung tìm hiểu, nghiên cứu về các phương pháp tối ưu mô hình. Chi tiết các kỹ thuật sẽ được trình bày chi tiết trong các phần sau của luận văn.

### 1.2.3. Nguyên lý tối ưu mô hình 3D

Việc tái tạo lại mô hình từ các vật thể từ các đối tượng thực bằng công nghệ tái tạo mô hình vật thể trên máy tính. Việc thu thập dữ liệu của đối tượng thực dưới dạng các điểm, liên kết các dữ liệu và tái tạo lại mô hình của đối tượng thực trong không gian 3D. Từ các dữ liệu điểm, một lưới tam giác được xây dựng để liên kết các điểm trong không gian 3 chiều. Mô hình đối tượng được số hoá sau khi thực hiện một số bước tinh chỉnh, làm mịn, vá lỗ thủng lưới tam giác để tăng cường chi tiết hoá mô hình.



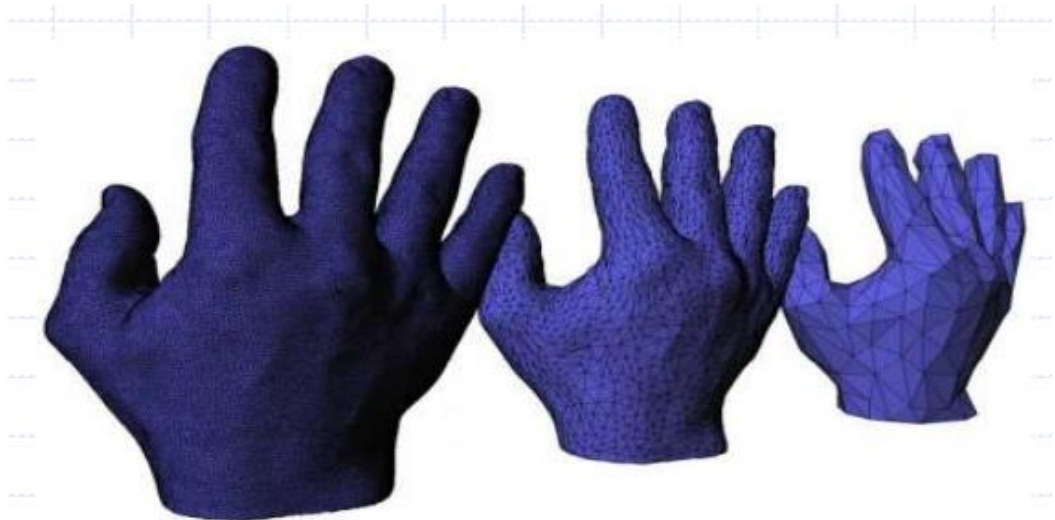
*Hình 1.22. Mô hình và xử lý từ máy quét 3D*

Ngày nay, kỹ thuật mô hình hoá các đối tượng 3D đã có nhiều ứng dụng rộng rãi, đặc biệt trong thiết kế và biểu diễn các đối tượng ba chiều. Việc tái lập lại các đối tượng 3D thực bằng cách sử dụng cơ chế tái tạo ngược bằng các thu thập dữ liệu các đối tượng thực dưới dạng điểm để liên kết các dữ liệu và tái tạo lại mô hình.

Mặt lưới đa giác thường được sử dụng để đại diện cho các bề mặt 3D. Cho một tập đỉnh mẫu lấy từ một bề mặt nhẵn, lưới tam giác với các đỉnh như là một đại diện (xấp xỉ) của bề mặt mẫu. Điều này còn phụ thuộc vào sự lựa chọn của các kết nối giữa các đỉnh gọi là cạnh. Chúng ta quan niệm tập rời rạc các điểm, các cạnh nối đại diện cho các bề mặt (thể hiện rằng các đặc tính nổi bật nhất của nó về các độ sâu sù, độ cong, vv) là đại diện cho các dữ liệu của mô hình 3D.

Nội dung chính của việc tối ưu hóa lưới là gì? Các mô hình sau khi thu thập được từ máy scan 3D có thể có nhiều chi tiết. Mật độ lưới càng dày thì dẫn đến tốn kém bộ nhớ, việc xử lý khi tính toán là vô cùng khó khăn. Mô hình chứa nhiều thông tin hình học dư thừa. Ý tưởng căn bản là:

- Loại bỏ hình học dư thừa.
- Giảm kích thước mô hình.
- Cải thiện hiệu suất thời gian chạy.



*Hình 1.23. Tối ưu hóa lưới*

Trong luận văn chúng ta sẽ tìm hiểu về hai trường hợp để xử lý tối giản lưới đó là:

- Tối giản hóa lưới mô hình theo bề mặt lưới tam giác: để tối ưu hóa cho những mô hình đối tượng tĩnh.
- Tối giản mô hình theo bề mặt lưới tứ giác: để tối ưu hóa cho những mô hình đối tượng chuyển động và biến dạng.

## CHƯƠNG 2. MỘT SỐ KỸ THUẬT TỐI ƯU HÓA MÔ HÌNH

### 2.1. Kỹ thuật tối ưu mô hình dựa trên lưới tam giác

#### 2.1.1. Giới thiệu về tối ưu và các phương pháp tối ưu phổ biến

Với các các mô hình thu được từ máy quét 3D thì mặt lưới tam giác thường được sử dụng để đại diện cho các bề mặt 3D. Do trong không gian 3D, qua 3 điểm bất kỳ luôn tồn tại một mặt phẳng chứa chúng. Vì vậy, các mô hình lưới sinh ra và tính toán thông thường là các bề mặt lưới tam giác. Bắt đầu với một tam giác ban đầu tùy ý với các đỉnh được đưa ra, các thuật toán chính ở đây là hoán đổi các cạnh, loại bỏ các đỉnh, các cạnh một cách tham lam để giảm tối đa các bề mặt mà vẫn giữ đảm bảo được hình dáng của đối tượng.

Trong nhiều năm qua, vấn đề của làm giảm (hoặc làm mịn) bề mặt lưới nhận được rất nhiều sự chú ý. Vấn đề này đã được đi sâu nghiên cứu trong suốt thế kỷ 19 và bây giờ có thể được coi là một công nghệ trưởng thành. Một số công trình đã có kết quả thực nghiệm và đã tích hợp sẵn trong gói mô hình phổ biến trong các phần mềm 3D như Maya, Blender hoặc MeshLab. Mục tiêu chính của việc tối ưu hóa lưới là có được một mô hình gần giống nhất với hình dạng mô hình ban đầu mà có một số lượng hình tam giác tạo ra mô hình là nhỏ nhất. Vì các hình dáng của mô hình cần được bảo đảm không bị biến dạng quá nhiều trong quá trình tối ưu lưới.

Các hướng tiếp cận việc tối ưu hóa bề mặt lưới tam giác đã được nghiên cứu, có 3 hướng chính đó là:

- ❖ Multi-resolution:[Eck at al, 1995,P.173-182 ]
  - Remeshing
  - Parametric Surfaces
  - Subdivision Surfaces
- ❖ Polygonal Simplification:
  - Theo toán tử địa phương: Vertex Clustering [Rossignac & Borrel ,93]; **Incremental Decimation**; Triangle Collapse.
  - Theo toán tử toàn cầu: Low-Pass Filtering[He et al, 96]; Morphological Operators[Nooruddin, 99]; Alpha-Hull[El-Sana and Varshney, 98].
- ❖ Image Imposters: Warping [Rafferty98]; Texture Depth Meshes [Aliaga99]

Tùy vào từng trường hợp mà người ta dựa vào các hướng giải quyết khác nhau. Để tối ưu hóa bề mặt lưới truyền thống được ưu tiên nhất. Chúng ta sẽ tiếp cận theo hướng thứ 2 là: Polygonal Simplification, và trong hướng này chúng ta sẽ quan tâm đặc biệt đến phương pháp **Incremental Decimation**, vì phương pháp này được phát triển rộng rãi trong các gói chương trình đã nghiên cứu thành công.

### 2.1.2. Phương pháp Incremental Decimation

❖ Phương pháp Decimation Incremental thực hiện dựa trên một số bước sau:

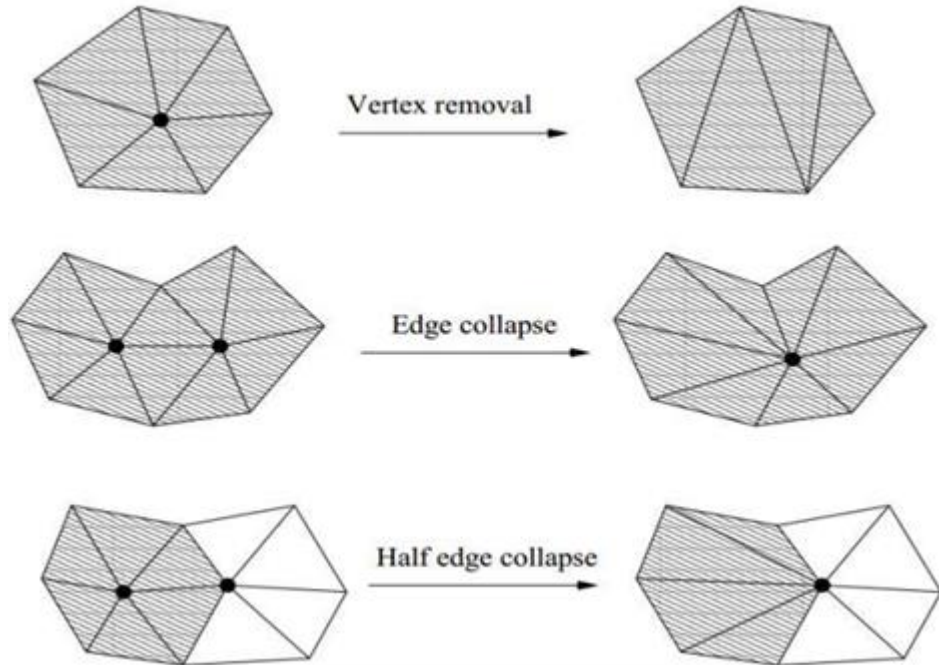
- Loại bỏ một đỉnh tại một thời điểm và sửa chữa lại bề mặt mô hình bởi việc loại bỏ.

- Ba toán tử chính để loại bỏ điểm là:

+ Vertex Removal // Xóa điểm.

+ Edge Collapse// Gộp 2 điểm trên cạnh thành một điểm.

+ Half Edge Collapse // Gộp 2 điểm thành 1 điểm, tuy nhiên 1 điểm sẽ được sát nhập vào điểm kia (điểm này giữ nguyên vị trí).



Hình 2.1 Kỹ thuật loại bỏ điểm

- Thứ tự Decimation dựa trên một số hàm chi phí (ưu tiên).

❖ Quy trình cho việc tối ưu hóa bề mặt lưới tam giác như sau:

- Quy trình chung:

*Repeat:*

{

- Chọn đối tượng 3D;

- Áp dụng toán tử decimation để giảm lưới đối tượng;

}

*Until* Mục tiêu tối giản được đáp ứng.

- Tối ưu hóa tham lam:

*For* mỗi khu vực của đối tượng:

{

Đánh giá chất lượng

*Enqueue* (chất lượng, khu vực);

}

Repeat :

{

Get vùng lưới tốt nhất từ queue;

Áp dụng toán tử decimation;

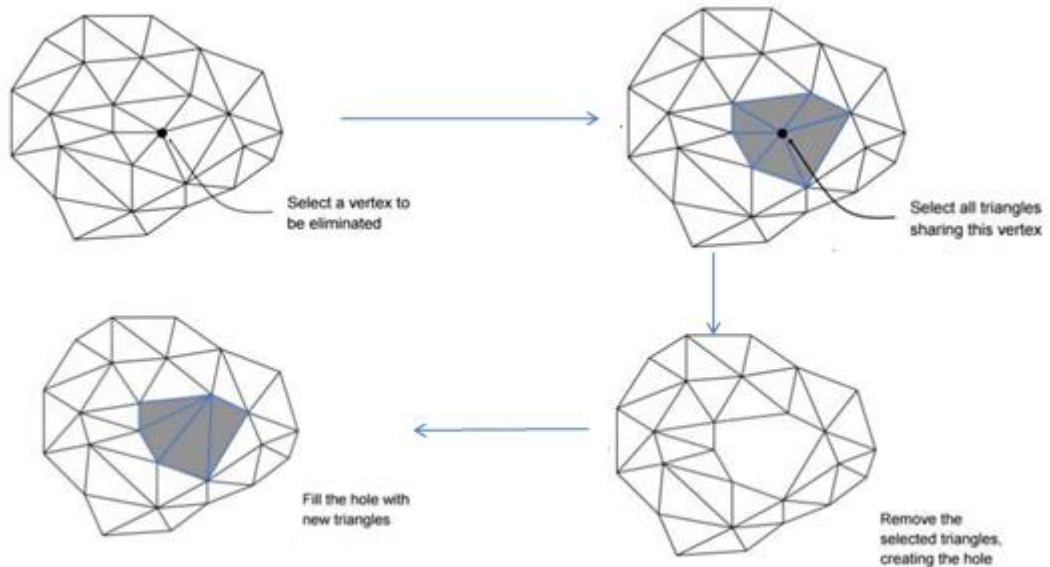
Update queue;

}

Until Mục tiêu tối giản được đáp ứng.

❖ Quy trình loại bỏ đỉnh và phục hồi bề mặt:

- Áp dụng thứ tự ưu tiên để loại bỏ điểm ra khỏi mô hình đối tượng, sau khi đã xóa điểm xong thì cần thêm bước nữa để phục hồi bề mặt của đối tượng.



Hình 2.2 Loại bỏ và phục hồi bề mặt

❖ **Phác thảo thuật toán loại bỏ:**

Phân hạng các đỉnh như loại bỏ (hay không) theo thứ tự ưu tiên

Repeat:

{

- Áp dụng hoạt động tối ưu hóa (loại bỏ đỉnh);

- Kết quả sinh lỗ thủng lưới tam giác (thuật toán đệ quy chia tách);

- Cập nhật lỗi của đỉnh bị ảnh hưởng;

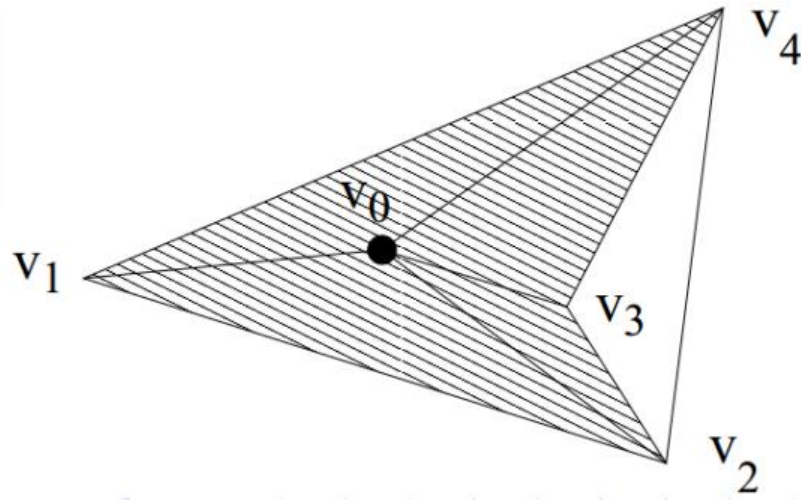
}

Until Mục tiêu giảm lưới là đạt yêu cầu.

❖ **Khi nào việc xử lý loại bỏ đỉnh dừng?**

Khi tối ưu hóa lưới mô hình chúng ta luôn luôn muốn số lượng lưới thấp nhất có thể mà bề mặt mô hình vẫn đảm bảo về hình dáng. Chúng ta thử nghiệm qua ví dụ sau:





Hình 2.3 Ví dụ về xóa điểm

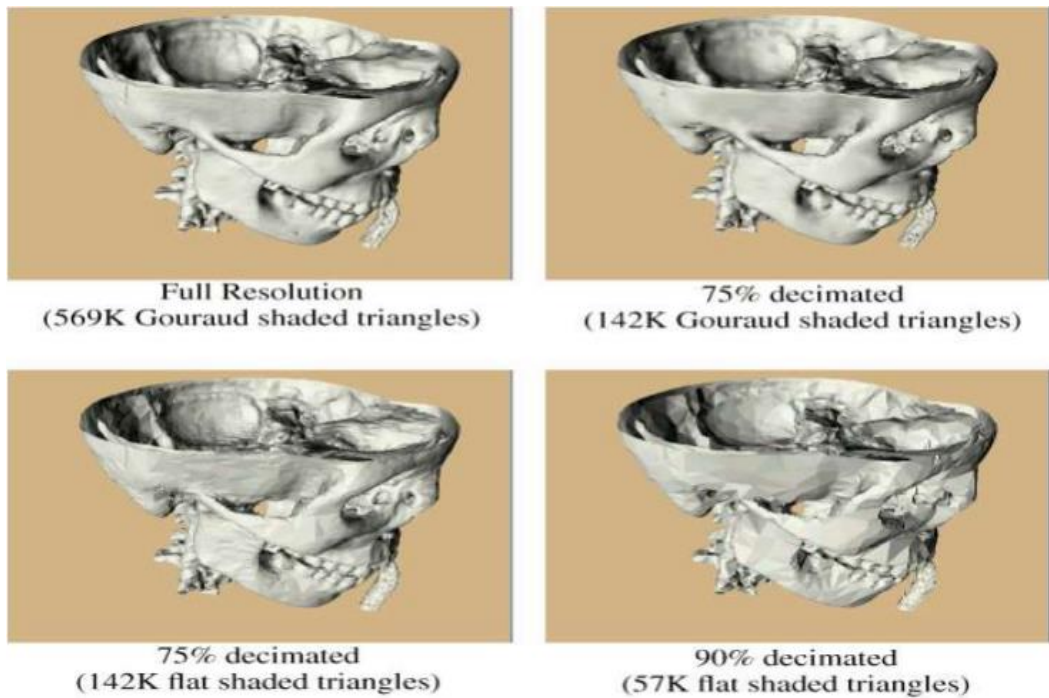
B1: Xóa bỏ  $v_0$

B2: Tái tạo lại tam giác bằng cách kết nối  $v_1$  với  $v_3$ ,  $v_2$  và  $v_4$  đã có đường kết nối trong lưới. Không thay thế các kết nối  $v_2$  và  $v_4$ .

Quy tắc: Không bao giờ kết nối hai đỉnh đã được kết nối, khả năng này có xu hướng xảy ra nhiều hơn và thường xuyên hơn là tối ưu hóa lưới.

B3: Tính toán lại độ ưu tiên cho các đỉnh còn lại  $v_1, v_2, v_3, v_4$ .

B4: Thiết lập số lượng lần lặp cho đến khi số lượng lưới là đạt yêu cầu.



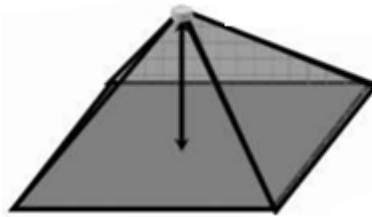
Hình 2.4 Tối ưu lưới theo William J. Schroeder bằng việc xóa đỉnh

❖ **Thứ tự mà các đỉnh được loại bỏ như thế nào?**

Một phép đo phải được thực hiện trong những Error dự kiến gây ra bởi áp dụng các toán tử tính toán. Phép đo Error gần đúng này được sử dụng để ưu tiên loại bỏ đỉnh, đánh giá chất lượng của các kết quả.

Error Metrics là sự khác nhau giữa hai mô hình lưới đa giác. Error Metrics giữa hai mô hình là nhỏ có nghĩa là hai mô hình gần giống nhau. Các độ đo:

- Edge length //Độ dài các cạnh kết nối giữa điểm đang xét đến các điểm láng giềng.
- Distance to plane [Schroeder et al.,1992], [Renze and Oliver,96]//Khoảng cách từ điểm tới mặt phẳng chứa láng giềng của nó.



Hình 2.5. Khoảng cách từ điểm tới mặt phẳng

- **Curvature**[Hamann,94]//Bề mặt cong của các mặt phẳng chứa điểm.



Hình 2.6. Bề mặt cong

Phép đo độ cong bề mặt tại một điểm Discrete Curvature Error Metric của SunJeong Kim theo công thức (1) sau:

$$\varepsilon = \max \{ |\mathbf{W} v_1 - \mathbf{W}' v_1|, |\mathbf{W} v_2 - \mathbf{W}' v_2|, \dots, |\mathbf{W} v_n - \mathbf{W}' v_n| \} \quad (1)$$

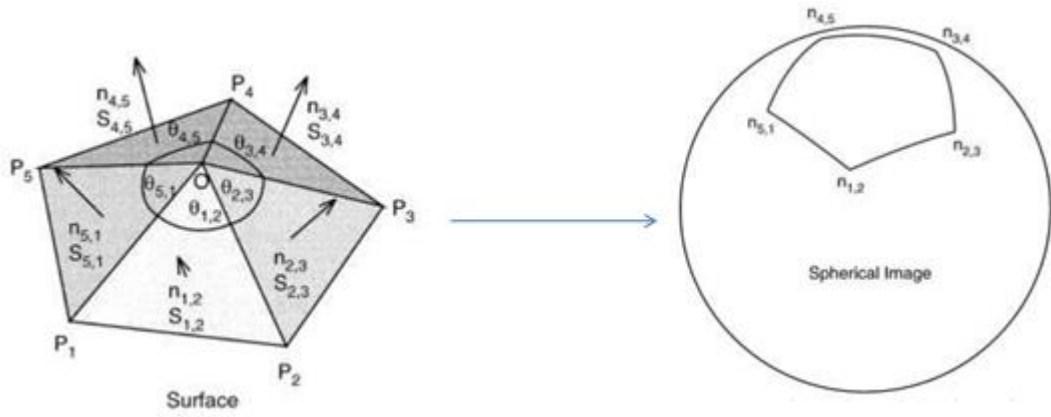
Với :  $\mathbf{W} v_i$  là độ cong rời rạc của một đỉnh  $v_i$  trên một bề mặt ban đầu.

$\mathbf{W}' v_i$  là độ cong rời rạc sau khi loại bỏ một đỉnh.

$v_i$  là một đỉnh hàng xóm của một  $v$  đỉnh loại bỏ.

$i = 1, 2, \dots, n.$

Phương pháp áp dụng công thức “The Gaussian Curvature” là phương pháp xấp xỉ bằng việc thiếu hụt góc xung quanh một đỉnh, phát triển từ định lý của Rodrigues (Kreyszig, 1991, tr 187; Hilbert, 1952).



Hình 2.7. Tối ưu lưới theo The Gaussian Curvature

Khi đó xấp xỉ độ cong bề mặt ( $k$ ) tại đỉnh  $O$  được tính theo công thức (2):

$$k = \frac{2*\pi - \sum \theta_{i,i+1}}{\frac{1}{3} * \sum S_{i,i+1}} \quad (2)$$

Với  $\theta_i$  : là các góc tại đỉnh  $O$  sinh ra từ các cạnh kề nhau cùng kết nối đến đỉnh  $O$ .

$S_i$ : là diện tích của các tam giác chứa đỉnh  $O$ .

Và theo công thức (2) thì, với  $k$  càng nhỏ thì đỉnh  $O$  càng dễ xóa bỏ. Do với 1 điểm bất kỳ, nằm trên 1 mặt phẳng thì tổng các góc xung quanh nó là  $2*\pi = 360^0$ .

Công thức này cũng được sử dụng rộng rãi như Azariadis, 1997; Calladine, 1983, p. 144; Daniel, 1997; Hinds, 1991; Lin, 1982; Regge, 1961; Sorkin, 1975; Stokey, 1992.

#### ❖ Thuật toán chi tiết:

Khởi tạo:

$\forall v \in V: v.errormetric := CalcErrorMetric(v);$

Sort\_Vertexlist (); //Thứ tự tăng dần

Repeat:

{

Lấy  $v_0$  (một đỉnh trong lưới tam giác); // Có độ đo( $k$ ) nhỏ nhất

% If ( $v_0.errormetric < given\_errormetric$ )

Loại bỏ ( $v_0$ );

Lưới tam giác (Nghb ( $v_0$ )); // Tính toán lại lưới tam giác khi đã xóa đỉnh và phục hồi bề mặt.

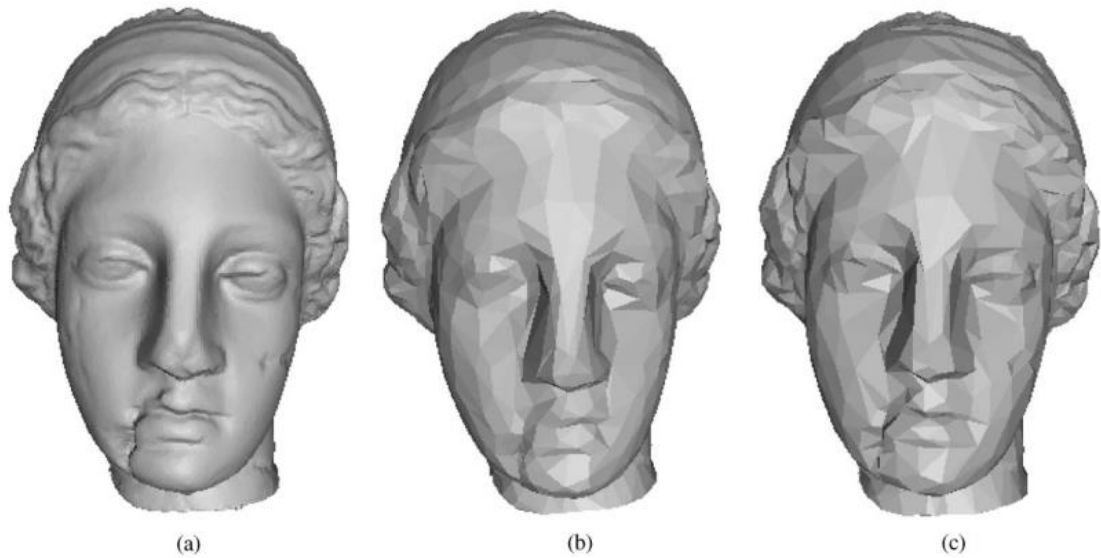
Cập nhật ( $v_0$ ):

% Else break;

}

Until Mục tiêu cần giảm là đạt.

## ❖ Kết Quả:

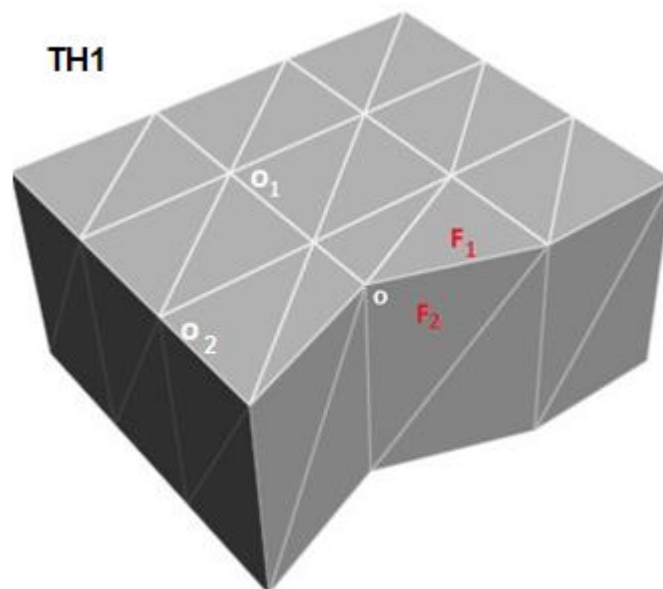


Hình 2.8. Các kết quả tối ưu hóa của một mô hình Igea: (a) Bản gốc (số mặt tam giác: 268.686), (b) Sử dụng phép đo khoảng cách từ điểm đến mặt phẳng, và (c) Sử dụng độ cong gauss (số mặt tam giác: 3.000).

### 2.1.3. Đề xuất thuật toán cải tiến

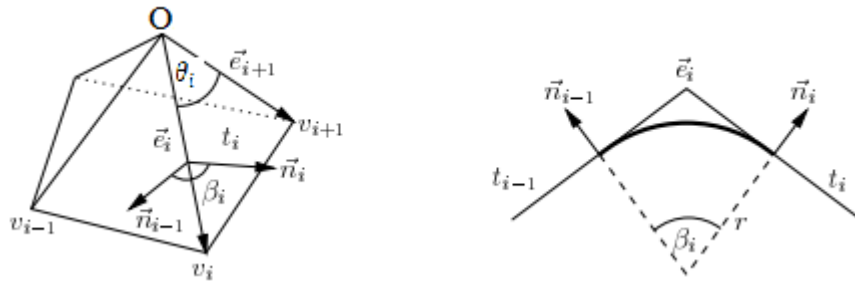
Chúng ta vẫn sử dụng phương pháp Curvature cho từng đỉnh của mô hình, nhưng chúng ta nhận thấy một số trường hợp ngoại lệ của các điểm trên mô hình khi áp dụng phương pháp này.

Trường hợp 1: Những điểm O nằm trên cạnh bên có tổng các góc bằng hoặc lớn hơn  $360^\circ$  nên theo công thức:  $\bar{K} = \int_S K = 2*\pi - \sum_{i=1}^n \theta_i < 0$  tại điểm O.



Hình 2.9. Góc tại đỉnh O

Do đó, đỉnh  $O$  tại vị trí này sẽ được phân biệt với đỉnh  $O$  khác tại vị trí bên trong mặt phẳng bằng cách so sánh góc giữa hai véc tơ pháp tuyến của hai mặt phẳng chứa hai tam giác kề nhau (có chung đỉnh  $O$ ) không được lớn hơn một giá trị cho phép.

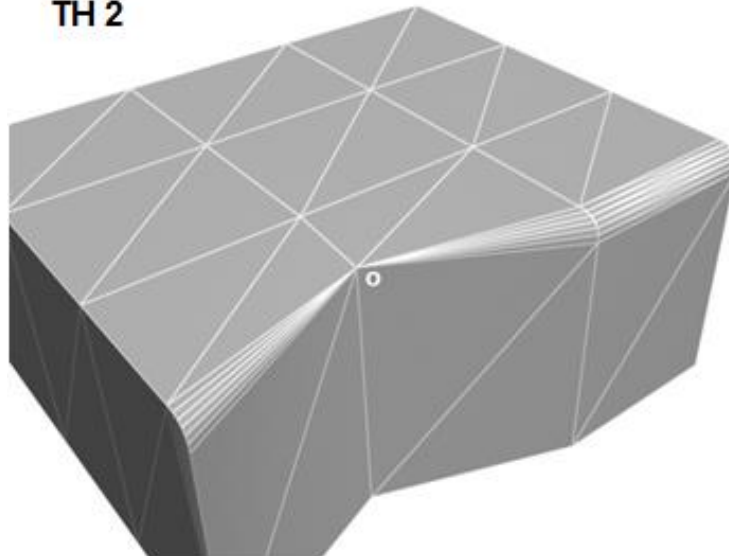


Hình 2.10. Góc tại đỉnh  $O$  và góc giữa 2 mặt phẳng kề nhau

Chúng xác định các cạnh  $\vec{e}_i = v_i - O$  và góc giữa hai cạnh liên tiếp  $\theta_i = \angle(\vec{e}_i, \vec{e}_{i+1})$ . Các tam giác chứa  $\vec{e}_i, \vec{e}_{i+1}$  được đặt tên  $t_i = \Delta(v, v_i, v_{i+1})$ , véc tơ pháp tuyến tương ứng là  $\vec{n}_i = \frac{\vec{e}_i * \vec{e}_{i+1}}{\|\vec{e}_i * \vec{e}_{i+1}\|}$ . Trên cạnh  $\vec{e}_i$  góc giữa các véc tơ pháp tuyến của các tam giác  $t_i$  và tam giác liền kề  $t_{i-1}$  là  $\beta_i = \angle(\vec{n}_{i-1}, \vec{n}_i)$ .

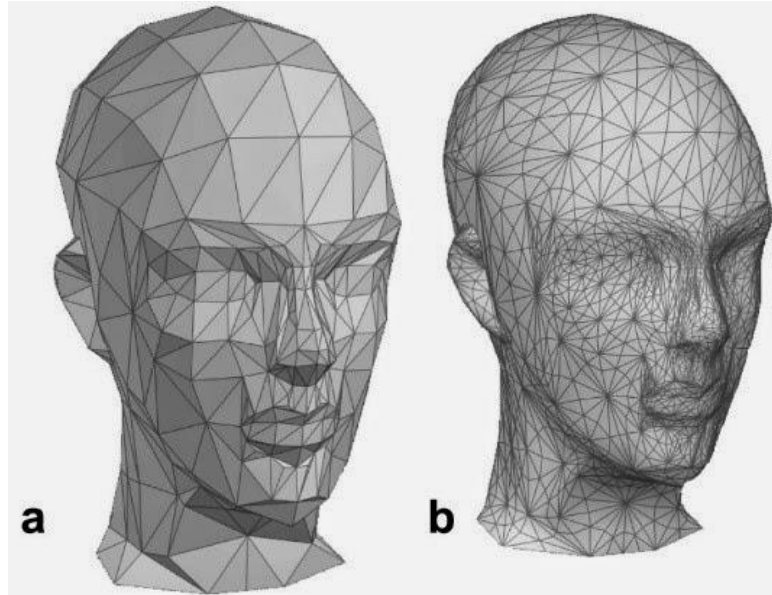
Trường hợp 2: Khi các góc của công thức:  $\bar{K} = \int_S K = 2 * \pi - \sum_{i=1}^n \theta_i$ , thỏa mãn điều kiện cho phép và góc giữa 2 véc tơ pháp tuyến của 2 tam giác kề nhau nhỏ hơn một giá trị cho trước, ta lại xét đến việc số đường kết nối đến điểm  $O$  là bao nhiêu (hay bậc đỉnh  $O$  là cao). Chúng ta sẽ giữ lại đỉnh  $O$  nếu số lượng cạnh kết nối đến nó là nhiều hơn 1 ngưỡng cho phép.

TH 2



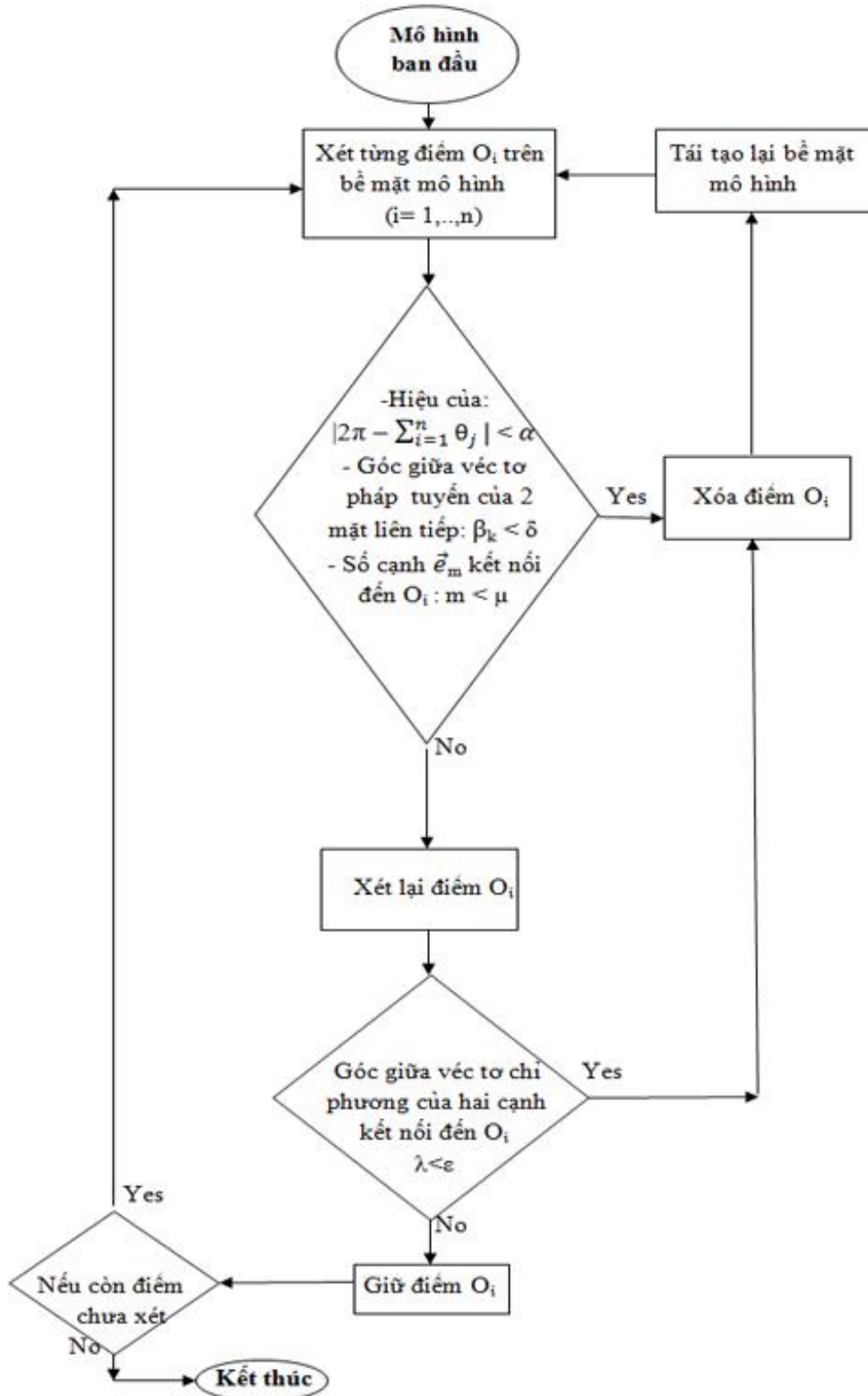
Hình 2.11. Đỉnh  $O$  với nhiều cạnh kết nối

Kết quả khi thử nghiệm trường hợp 2:



*Hình 2.12. Mô hình sau và trước tối ưu*

Để giải quyết 2 trường hợp tôi xin đề xuất sơ đồ khối xóa điểm như sau:



Hình 2.13. Sơ đồ khối việc xóa điểm

Một tập tam giác lưới  $M$  bao gồm một tập các đỉnh  $V = \{v_i\}_{i \in \mathbb{R}^3}$ , được kết nối bởi một tập các cạnh  $E = \{e_j = (v_{j1}, v_{j2})\}_j$  và một bộ tam giác  $T = \{t_k = \Delta(v_{k1}, v_{k2}, v_{k3})\}_k$ .

$v_{k3})\}_k$ . Với  $v \in V$  là một đỉnh của một tam giác lưới  $M$  và để  $v_1, \dots, v_n$  là các đỉnh lân cận của  $O$ .

### Thuật toán đề xuất chi tiết:

*Khởi tạo:*

$\forall v \in V: v.errormetric := CalcErrorMetric(v);$

$Sort\_Vertexlist();$  //Thứ tự tăng dần

*Repeat:*

{

Lấy  $v0$  (một đỉnh trong lưới tam giác); // Có độ đo ( $k$ ) nhỏ nhất

% If  $((v0.errormetric < given\_errormetric)$  and  $(\downarrow (\vec{n}_{i-1}, \vec{n}_i) < \delta)$  and  $(\vec{e}_i ; i < \mu)$ )

Loại bỏ ( $v0$ );

Lưới tam giác ( $Nghb(v0)$ ); // Tính toán lại lưới tam giác khi đã xóa điểm và phục hồi bề mặt với các đỉnh hàng xóm.

Cập nhật ( $v0$ );

% Else break;

}

Until Mục tiêu cần giảm là đạt .

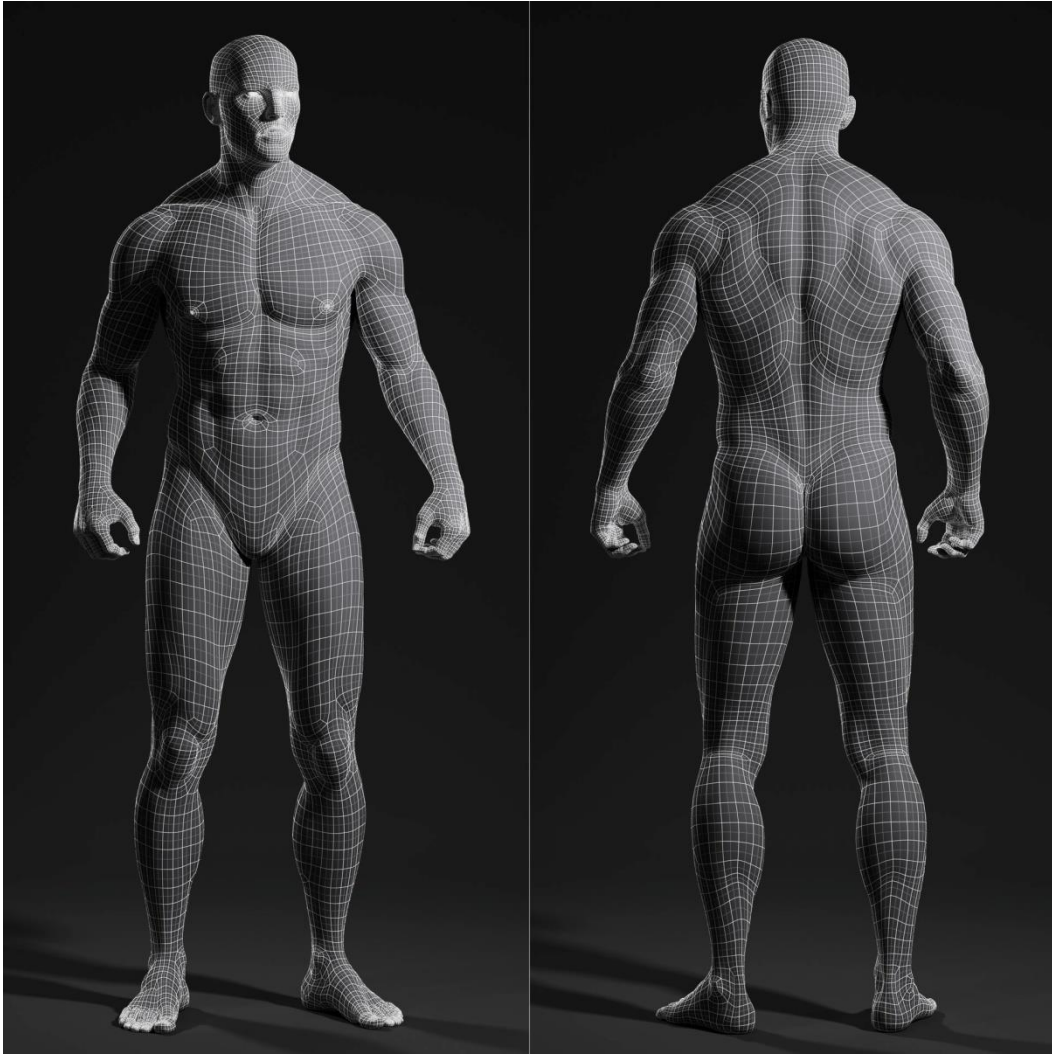
## 2.2. Kỹ thuật tối ưu mô hình dựa trên lưới tứ giác

Trong phương pháp tối ưu hóa mô hình lưới tam giác. Chúng ta thấy bề mặt lưới thu được của mô hình là có thể đáp ứng được so với yêu cầu đặt ra về phân kích thước dữ liệu và mô hình chủ yếu là những mô hình tĩnh ví dụ như tượng, đồ vật, ... Tuy nhiên một vấn đề lớn ở đây là chúng ta muốn các mô hình sau khi tối ưu phải đạt được cả nhu cầu ứng dụng về chuyển động như hoạt hình và biểu cảm nhân vật. Những mô hình này có không gian lưới phải đáp ứng được cho các chuyển động mà mô hình không bị biến dạng. Đây là điều mà việc tối ưu hóa bề mặt lưới tam giác đang còn hạn chế.

- Đầu vào : là mô hình với bề mặt lưới tam giác.

- Đầu ra: là mô hình với bề mặt lưới tứ giác (có thể có một vài lưới tam giác), và đã tối ưu lưới tứ giác.





*Hình 2.14. Mô hình lưới cho animation*

Để giải quyết điều này, dựa vào kiến thức của mình, tôi xin đưa ra các bước tiến hành như sau:

*Bước 1:* Chuyển mô hình với bề mặt lưới tam giác của về mô hình bề mặt lưới tứ giác.

*Bước 2:* Làm mềm bề mặt lưới tứ giác sau khi đã chuyển đổi ở bước 1.

*Bước 3:* Tối ưu hóa bề mặt lưới tứ giác sau khi đã làm mềm ở bước 2.

Chúng ta sẽ đi sâu để giải quyết từng bước sau:

### **2.2.1. Chuyển mô hình bề mặt lưới tam giác của về mô hình bề mặt lưới tứ giác**

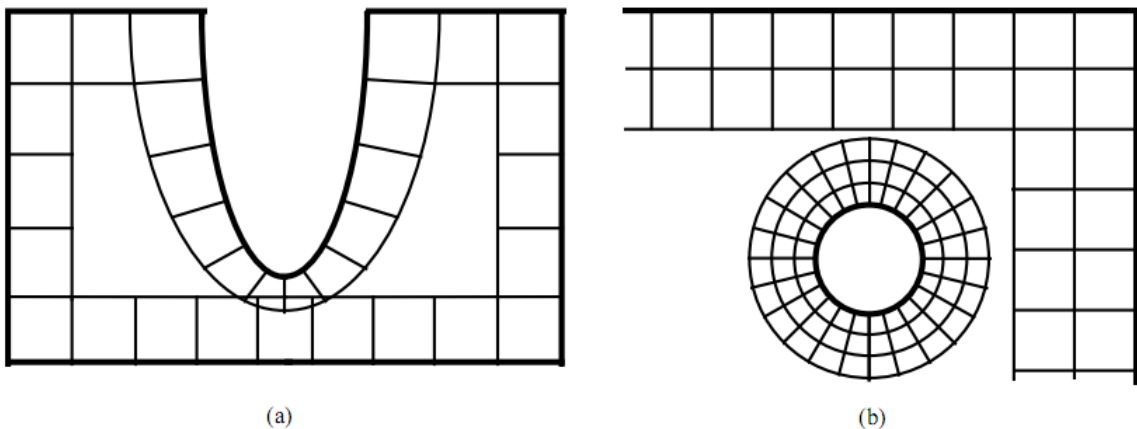
Công việc chuyển mô hình có bề mặt lưới tam giác về mô hình có bề mặt lưới tứ giác [9] gồm cả hai phương pháp trực tiếp và gián tiếp:

Phương pháp gián tiếp bao gồm các thủ tục đòi hỏi bề mặt lưới mô hình có lưới tam giác ban đầu được kết hợp với tam giác liền kề một cách hệ thống, trong hầu hết các trường hợp, kết quả là một bề mặt lưới gồm tất cả các tứ giác. Trong khi phương pháp này có thể được thực hiện nhanh chóng, đôi khi phương pháp này có thể để lại

một số lượng lớn các đỉnh bất quy tắc. Một đỉnh của một lưới tứ giác là “bất thường” nếu nút có nhiều hơn hoặc ít hơn bốn đỉnh lân cận (hay số bậc của đỉnh khác 4).

Phương pháp trực tiếp lại không quá quan tâm đến bề mặt lưới tam giác ban đầu. Tứ giác thay thế đặt trực tiếp lên bề mặt lưới mô hình hoặc tứ giác có thể được đặt sau khi phân hủy bề mặt mô hình thành các vùng đơn giản. Trong số các phương pháp trực tiếp, thuật toán “lát nền” cung cấp một số đặc tính mong muốn. Blacker mô tả điều này như: “Ranh giới nhạy cảm”, “Định hướng Insensitive”. Xoay hoặc dịch lưới mô hình đưa ra các kết quả nên hay không nên thay đổi lưới topo. Một lưới được tạo ra trong một mô hình chuyển đổi được tương đương với lưới ban đầu, và ít điểm “bất thường”. Vì vậy, một lưới với vài nút bất thường, đặc biệt là ranh giới nơi hình dạng của đối tượng biến đổi.

Mặc dù trong thuật toán lát nền có những đặc điểm có lợi, tuy nhiên các vấn đề về chất lượng và hiệu quả chưa được giải quyết. Phương pháp được sử dụng bởi kỹ thuật lát nền đòi hỏi nhiều tính toán giao điểm tại nơi có sự biến đổi để tránh yếu tố chùng chéo. Hình 2.15(a) cho thấy một trường hợp đơn giản, nơi kiểm tra giao nhau phải được thực hiện. Hình 2.15(b) cho thấy một trường hợp khác thường gặp phải trong quá trình lát nền nơi mặt tiếp giáp phải hợp nhất. Nếu kích thước nguyên tố khác nhau rất nhiều, yếu tố chất lượng kém là có thể xảy ra.



Hình 2.15. (a) Hàng đầu tiên của các yếu tố được đặt sử dụng thuật toán mở minh họa sự can thiệp của các yếu tố đối lập.

(b) Sự khác biệt yếu tố kích thước lớn giữa mặt thường gặp.

Quad-morphing là một kỹ thuật mới được sử dụng để tạo ra các tứ giác từ một lưới tam giác hiện tại. Bắt đầu với một tam giác ban đầu, hình tam giác được biến đổi một cách hệ thống và kết hợp. Thuật toán Quad-Morph là một phương pháp gián tiếp để tạo bề mặt lưới tứ giác có thể tận dụng lợi thế của thông tin topo địa phương từ các tam giác ban đầu.

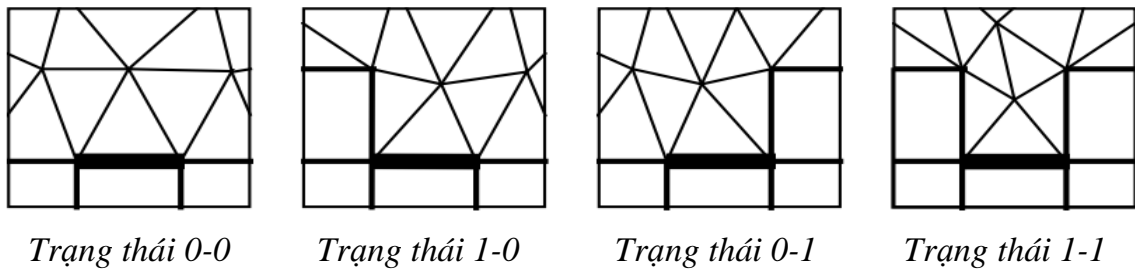
#### ❖ Phác thảo của thuật toán Quad-Morphing [9]

Quad-morphing được trình bày ngắn gọn trong các bước sau đây:

<1>. Xác định tam giác ban đầu của bề mặt lưới: Xác định thông tin được xây dựng thành các tam giác ban đầu.

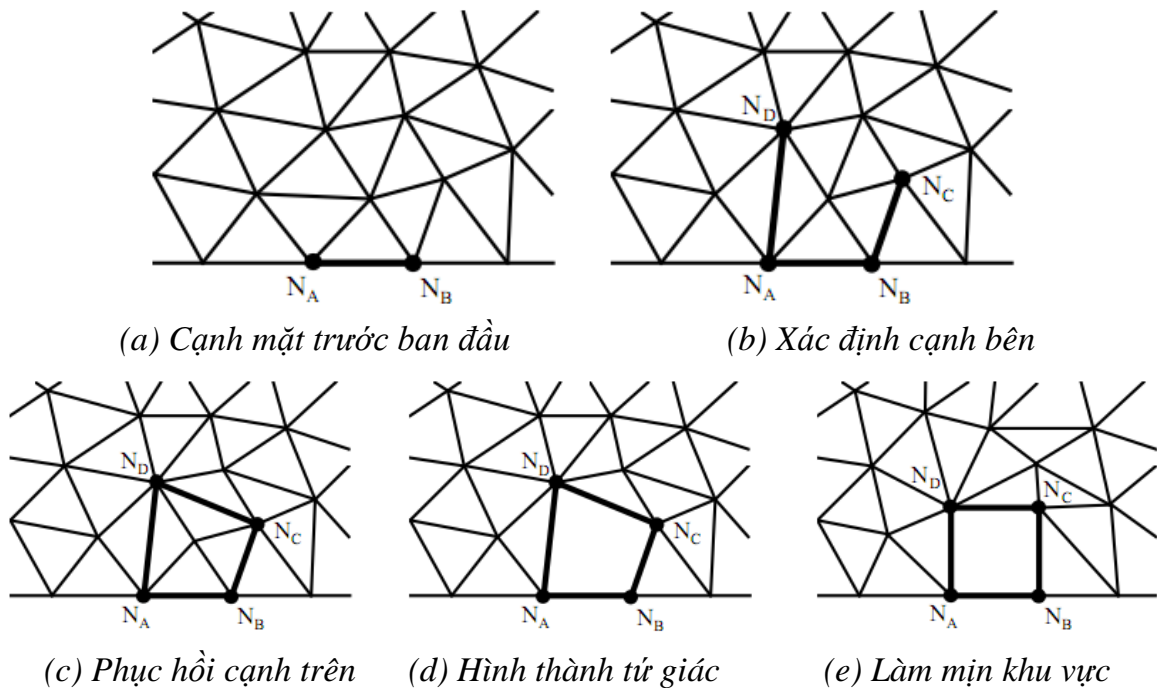
<2>. Định nghĩa cạnh mặt trước: cạnh mặt trước ban đầu được xác định từ các lưới tam giác ban đầu. Các cạnh khác trong tam giác đó là tiếp giáp với cạnh mặt trước ban đầu.

<3>. Phân loại cạnh mặt trước: Mỗi cạnh ở mặt trước ban đầu được sắp xếp theo trạng thái của nó. Trạng thái của một cạnh trước định nghĩa, cạnh cuối cùng sẽ được sử dụng trong việc hình thành một tứ giác. Hình 2.16 cho thấy bốn trạng thái có thể có của một cạnh mặt trước, được chỉ định bởi các dòng chữ đậm.



Hình 2.16: Các trạng thái của cạnh mặt trước

<4>. Xử lý cạnh mặt trước: Mỗi cạnh mặt trước được xử lý riêng để tạo ra một tứ giác mới từ lưới tam giác ban đầu. Hình 2.17(a) cho thấy cạnh mặt trước  $N_A - N_B$  trong tam giác sẵn sàng để được xử lý. Tứ giác được hình thành, cạnh mặt trước được xác định lại và các trạng thái cạnh mặt trước liền kề được cập nhật.



Hình 2.17. Các bước của quá trình xử lý tạo ra một tứ giác từ mặt trước  $N_A - N_B$

<5>.Xác định cạnh bên: Sử dụng các cạnh mặt trước như là cạnh cơ sở ban đầu của tứ giác, cạnh bên có thể được xác định bằng cách sử dụng một cạnh trong lưới tam giác kề của tam giác ban đầu, hình 2.17(b) cạnh  $N_B - N_C$ , hoặc bằng cách hoán đổi đường chéo của hình tam giác liền kề, hình 2.17(c) cạnh  $N_A - N_D$ , hoặc bằng cách tách hình tam giác để tạo ra một cạnh mới.

<6>. Phục hồi cạnh trên: Cạnh kết thúc trong tứ giác được tạo ra bởi một quá trình phục hồi cạnh. Trong quá trình này, là việc kết nối giữa hai đỉnh ở hai đầu của hai cạnh bên. Cạnh  $N_C - N_D$  trong hình 2.17(c) được hình thành từ một hoạt động trao đổi duy nhất.

<7>. Tứ giác hình thành: Việc sáp nhập bất kỳ hình tam giác bao quanh bởi các cạnh trước và các cạnh bên mới được tạo ra và cạnh trên như thể hiện trong hình 2.17(d) tạo thành tứ giác. Đồng thời xóa các đỉnh và cạnh bên trong tứ giác.

<8>. Khu vực cạnh mặt trước phân loại lại: cạnh trước là tịnh tiến bằng cách loại bỏ các cạnh mặt trước của tứ giác và thêm các cạnh mặt trước cho một tứ giác mới là cạnh giao của tam giác kề với tứ giác đó. Cạnh mặt trước mới được phân loại theo từng vùng. Xử lý cạnh mặt trước vẫn tiếp tục cho đến khi tất cả các cạnh trên mặt trước đã bị cạn kiệt.

#### ❖ Thực hiện:

<1> Định nghĩa và phân loại cạnh trước:

Các thiết lập ban đầu của các cạnh mặt trước được xác định từ các tam giác ban đầu. Tất cả các cạnh trong tam giác liền kề với một hình tam giác đơn được sử dụng như mặt trước. Trạng thái của một cạnh phía trước được xác định bằng cách tính góc tại các đỉnh trên hai đầu của cạnh với mỗi cạnh phía trước liền kề của nó. Thực tế, trạng thái của một cạnh phía trước được xác định bằng 2 bit, bit đầu tiên đại diện cho trạng thái tại đỉnh trái và bit thứ hai đại diện cho trạng thái tại đỉnh bên phải. Nếu góc ở hai đỉnh nhỏ hơn ngưỡng quy định ( $3\pi / 4$ ) thì bit của đỉnh được thiết lập (1); nếu không bit của đỉnh là unset (0).

Các góc tại các đỉnh ở cạnh mặt trước có thể được ước tính bằng cách tổng hợp các góc ở hình tam giác liền kề. Bằng cách xấp xỉ góc ở cạnh mặt trước trước từ các tam giác liền kề, đánh giá hình học có thể được loại bỏ. Các cạnh được đặt trên một trong bốn danh sách trạng thái như thể hiện trong hình 2.17. Phân loại các cạnh mặt trước theo trạng thái phục vụ hai mục đích.

Đầu tiên, nó định nghĩa đó cạnh bên phải được định nghĩa trước khi một tứ giác có thể được hình thành.

Thứ hai, nó ưu tiên mà cạnh mặt trước sẽ được xử lý đầu tiên. Cạnh mặt trước trong trạng thái 1-1 được ưu tiên đầu tiên tiếp theo cạnh trong trạng thái 0-1 và 1-0, tiếp theo là các cạnh trong trạng thái 0-0.

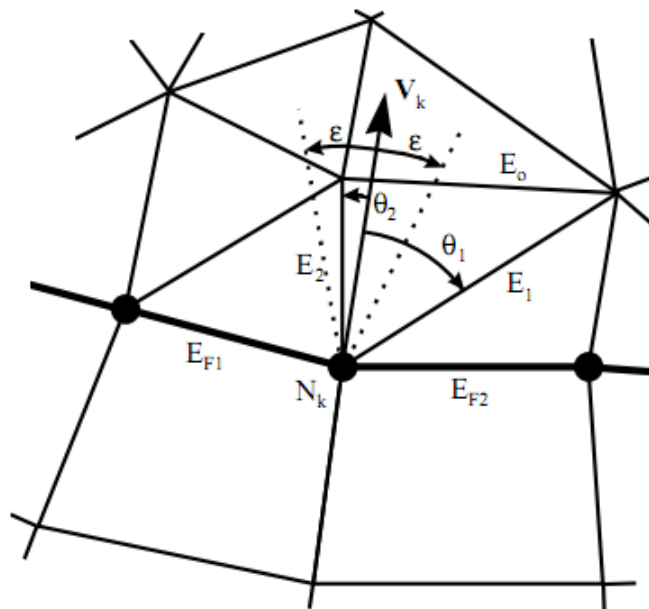
<2> Xử lý cạnh mặt trước:

Cạnh mặt trước được xử lý cùng một lúc để tạo thành tứ giác từ tam giác ban đầu. Một cạnh phía trước được lấy từ một trong bốn danh sách trạng thái, vẽ từ trạng thái cao hơn trước. Ưu tiên cũng được trao cho các cạnh mức thấp nhất trong danh sách. Các cạnh ở mức 0 là những cạnh mặt trước ban đầu; mức 1 là những cạnh mặt trước sau khi hàng đầu tiên của tứ giác đã được đặt; mức 2 sau hàng thứ hai; và tiến hành như vậy cho các trường hợp tiếp. Điều này đảm bảo rằng toàn bộ một hàng của tứ giác sẽ được đặt trước khi bắt đầu một hàng mới.

<3> Xác định cạnh bên:

Các trạng thái hiện tại của một cạnh mặt trước xác định cạnh được xử lý. Các cạnh mặt trước ở các trạng thái 0-0, 1-0 và 0-1 trước tiên phải xác định một hoặc hai cạnh bên. Một cạnh bên có thể được hình thành theo một trong ba cách:

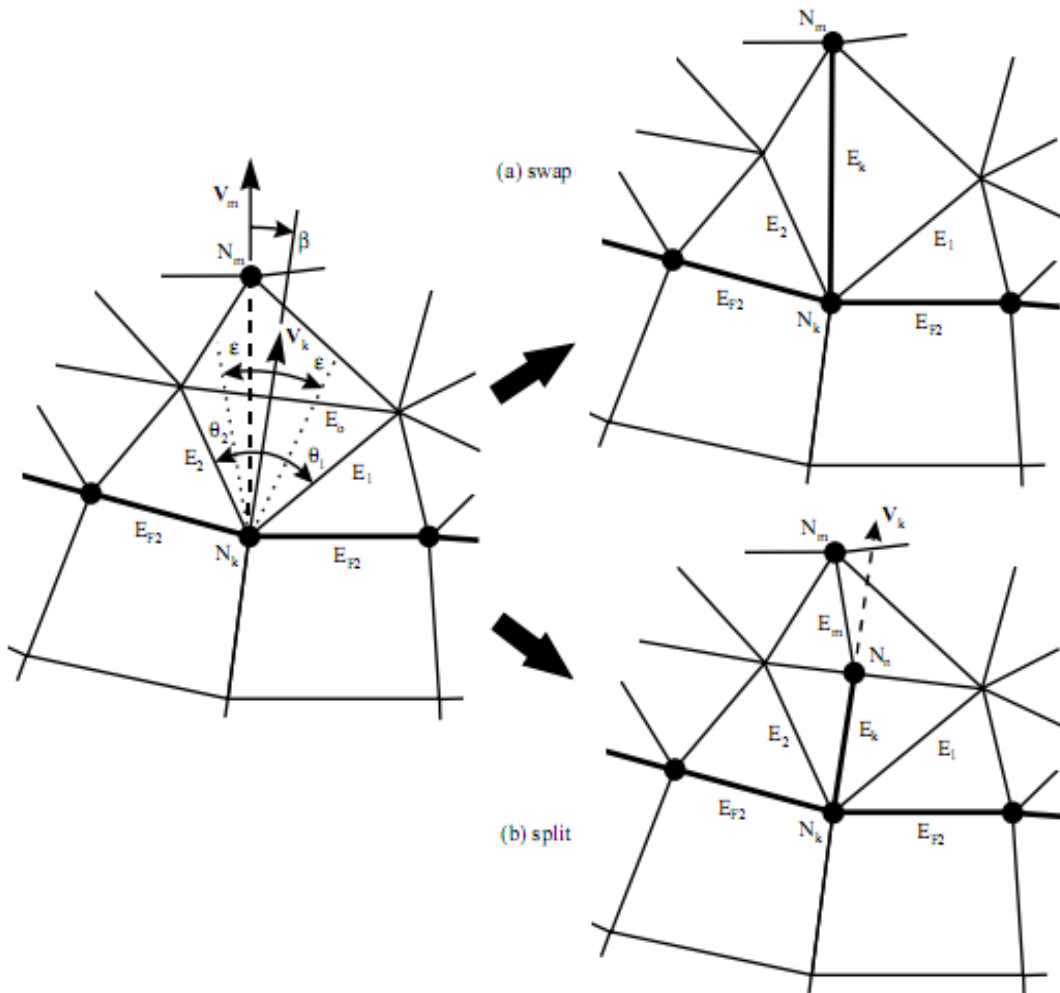
- (1) Một cạnh hiện trong lưới tam giác ban đầu có thể được sử dụng.
- (2) Các đường chéo giữa hai hình tam giác liền kề có thể được hoán đổi.
- (3) Một cạnh có thể được tạo ra bởi tách một cặp tam giác.



Hình 2.18. Lựa chọn cạnh bên

Hình 2.18 cho thấy một cạnh bên mới được quy định tại các đỉnh  $N_k$ , đó là một đỉnh trên mặt trước giữa các cạnh  $E_{F1}$  và  $E_{F2}$ . Các vectơ  $V_k$  lý tưởng cho các cạnh bên mới được định nghĩa bằng cách cắt đôi các vectơ hình thành bởi  $E_{F1}$  và  $E_{F2}$  (hoặc gọi là phân giác của 2 cạnh tại đỉnh  $N_k$ ). Các góc  $\theta_i$  được tính toán giữa  $V_k$  và các cạnh  $E_i$ , là cạnh của các hình tam giác chứa đỉnh  $N_k$ . Các cạnh với góc  $\theta$  nhỏ nhất sẽ được chọn làm ứng cử viên bên cạnh. Các cạnh được chọn, góc  $\theta$  cung cấp nhỏ hơn một góc không đổi  $\varepsilon$  ( $\pi / 6$ ). Cạnh  $E_2$  trong hình 2.18 được chọn là cạnh bên trong trường hợp này.

Khi không có góc  $\theta_i$  nhỏ hơn  $\varepsilon$ , một trong hai tùy chọn có thể được sử dụng. Cạnh đối diện ( $E_o$  trong hình 2.18) có thể được hoán đổi hoặc chia tách. Việc hoán đổi được sử dụng nếu góc  $\beta$  giữa  $V_k$  và  $V_m$  là nhỏ hơn  $\varepsilon$ . Các tùy chọn chia được thực hiện nếu  $\beta > \varepsilon$  hoặc chiều dài kết quả của  $E_k$  từ một trao đổi được quá dài so với  $E_{F1}$  và  $E_{F2}$ . Trong trường hợp sau này, một nút mới  $N_n$  được xác định, tách cạnh  $E_o$  tại giao điểm của vector  $V_k$  và cạnh  $E_o$ . Các cạnh  $E_k$ ,  $E_m$  cũng được thêm vào lưới tam giác khi tách hai tam giác liền kề để cạnh  $E_o$ . Sau đó cạnh  $E_k$  được sử dụng như cạnh bên phải của tứ giác.



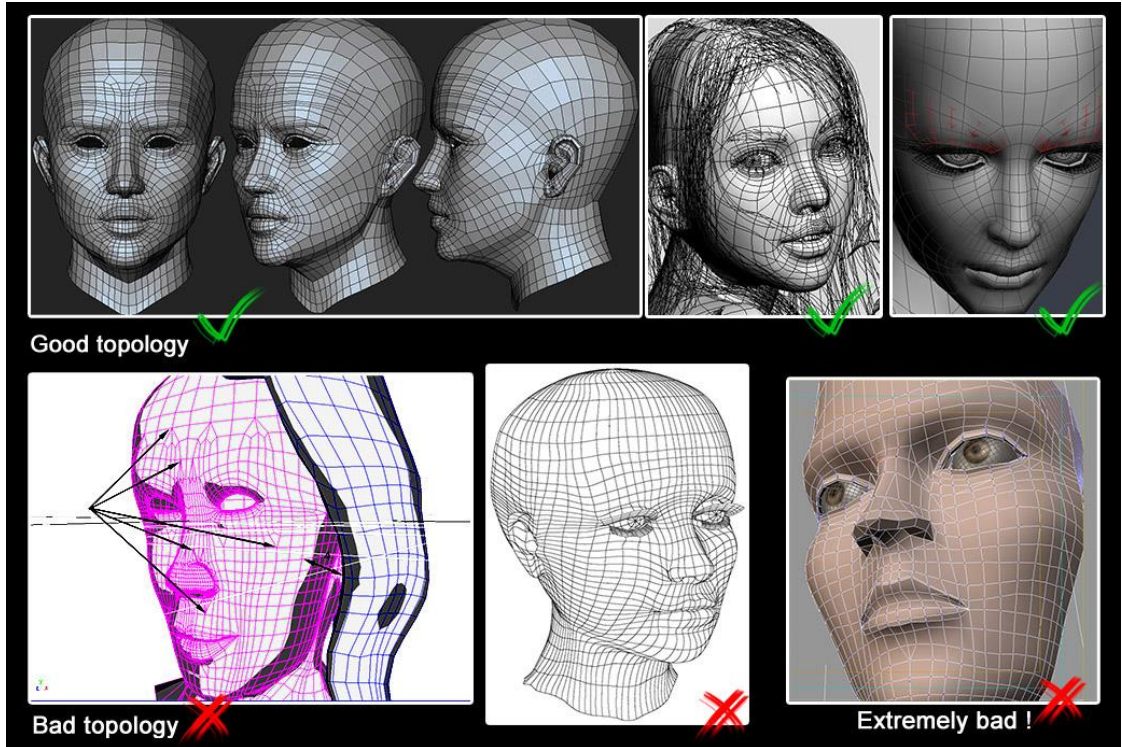
Hình 2.19. Tạo ra cạnh bên

Giải quyết cạnh bên trái của tứ giác tương tự như cạnh bên phải. Cạnh trên sinh ra khi kết nối 2 đỉnh cuối của 2 cạnh bên.

### 2.2.2. Làm mềm lưới tứ giác [10]

Các vị trí điểm lưới tứ giác sau khi đã được chuyển đổi đang ở những vị trí chưa hợp lý trên bề mặt lưới cho mô hình 3D vì vậy công việc tiếp theo của chúng ta là làm mềm bề mặt lưới này, phương pháp ở đây là ta có thể thêm điểm hoặc thay đổi vị trí một số điểm trên bề mặt lưới làm cho bề mặt lưới trở nên đẹp hơn về mặt thẩm

mỹ, thuận tiện cho việc chuyển động mô hình và tốt cho quá trình tối ưu hóa lưới ở bước tiếp theo.



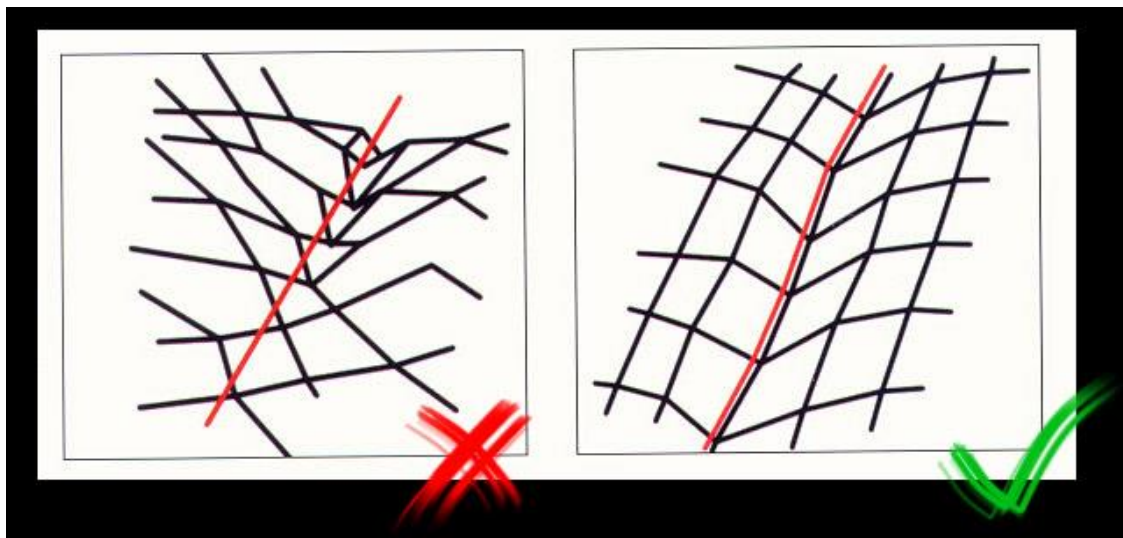
Hình 2.20. Bề mặt lưới 3D của mô hình

#### ❖ Các yêu cầu khi làm mịn

Một số yêu cầu khi làm mịn bề mặt đối tượng:

- Xử lý các tứ giác bị bóp méo nghiêm trọng.
- Được xử lý trong một phạm vi rộng các bề mặt của mô hình dựa vào tính vật

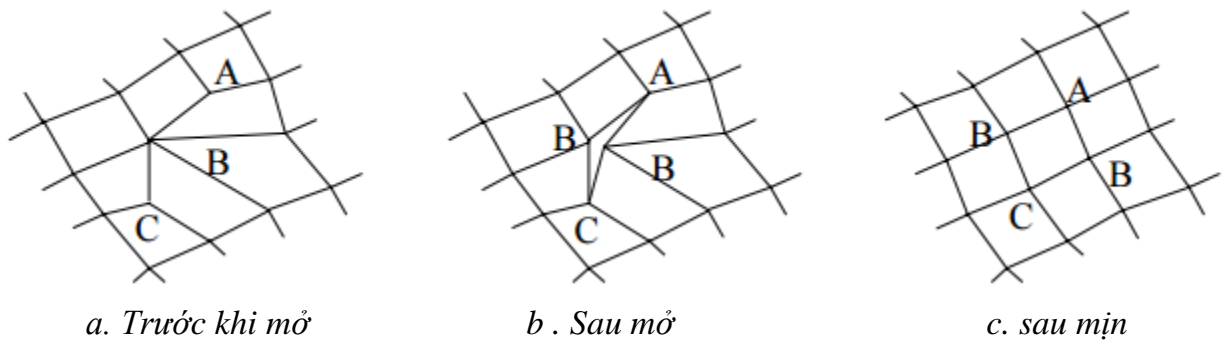
lý và sinh học của mô hình đó.



Hình 2.21. Bề mặt lưới 3D của mô hình

Trường hợp phát sinh nơi nó là cần thiết để cải thiện các yếu tố bóp méo nghiêm trọng. Sự biến dạng này có thể được gây ra bởi các thuật toán chia lưới. Một ví

dụ về một trường hợp kiểu này là biến dạng có thể được sửa chữa bằng việc thêm một nút trong lưới tứ giác, sau đó mở rộng tứ giác vừa sinh ra để tạo ra một lưới tứ giác hoàn hảo hơn, như thể hiện trong hình 2.22.

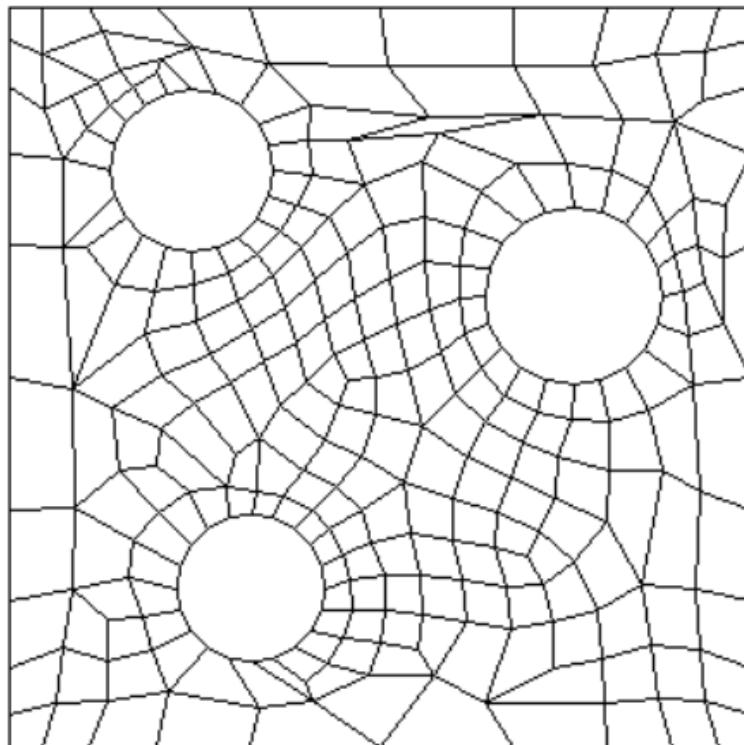


Hình 2.22. Xử lý một trường hợp làm mịn

### ❖ Thuật toán làm mịn tổng thể

Trong phần này, chúng ta sẽ trình bày chi tiết cho từng bước chính của việc làm mịn:

- Tính toán số liệu biến dạng ban đầu cho tất cả các tứ giác.
- Nếu lưới là một lớp lưới ta đánh dấu các nút kết hợp với các lớp biên.



Hình 2.23. Lưới nhiều lớp

- Dựa trên kích thước mô hình, khoảng cách tính toán  $\delta$ .
- Thiết lập số lần lặp lại  $niter = 1$

REPEAT (MAIN Smoothing LOOP):

FOR mỗi nút v DO:



1. If đỉnh  $v$  là không thể di chuyển hoặc đã được đã được làm mềm từ bước trước, then:

+ Break .

+ Vòng lặp đi đến nút tiếp theo trong danh sách.

2. If đỉnh  $v$  chưa được di chuyển bằng cách làm mềm trong một lần lặp lại trước đó, then

a) If nút  $v$  là trên lớp biên:

+ Di chuyển nút  $v$  sử dụng làm mềm.

+ Tách ra lớp ranh giới.

Else:

+ Thực hiện một làm mềm ràng buộc .

b) If khoảng cách di chuyển là nhỏ hơn so với khả năng độ sai lệch di chuyển:

+ Giữ vững vị trí cũ cho đỉnh

+ Di chuyển đỉnh  $v$  từ danh sách các nút làm mềm.

Else cho phép di chuyển:

+ Đặt hàng xóm của đỉnh vào trong danh sách các đỉnh đã làm mềm.

If chúng chưa phải là đã có.

+ So sánh với khoảng cách lớn nhất di chuyển bằng một đỉnh và theo dõi trong những khoảng cách lớn nhất.

+ Cập nhật biến dạng các giá trị số liệu các yếu tố tiếp giáp

3. Nếu  $niter \geq 2$ , sau đó gọi mềm dựa trên Tối ưu hóa. Đó là, để cho làm mềm có 2 lần lặp lại để cải thiện lưới, trước khi gọi cao hơn tối ưu hóa dựa trên làm mềm:

a) Tìm các số liệu biến dạng tối thiểu từ tất cả các khía cạnh kết nối đến đỉnh.

b) Nếu số liệu tối thiểu  $\leq a$  độ sai lệch nhất định

+ Thử tối ưu hóa dựa trên làm mềm

c) Nếu tối ưu hóa dựa trên làm mềm chuyển các nút, sau đó:

+ Đặt hàng xóm của đỉnh vào danh sách đã làm mềm, nếu chưa có

+ So sánh với khoảng cách lớn nhất di chuyển bằng một đỉnh và theo dõi trong những khoảng cách lớn nhất

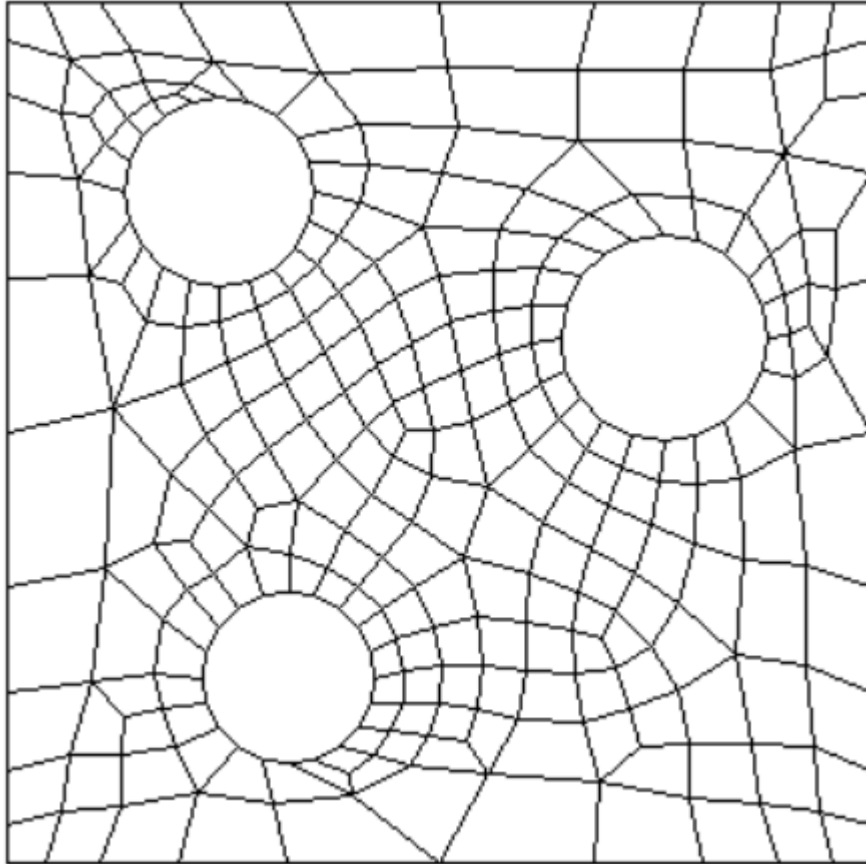
+ Cập nhật biến dạng các giá trị số liệu các thành phần tiếp giáp

END DO (vòng lặp cuối qua các đỉnh)

$niter = niter + 1;$

*UNTIL không có nhiều đỉnh đã di chuyển đủ để đảm bảo sự lặp lại khác. Đó là, nếu không đỉnh đã di chuyển hoặc nếu khoảng cách tối đa di chuyển ít hơn ( $1,75 * \text{độ sai lệch di chuyển}$ ).*

*(END MAINSMOOTHING LOOP)// kết thúc vòng lặp làm mềm*



Hình 2.24. Lưới nhiều lớp đã được làm mịn

### 2.2.3. Tối ưu hóa lưới tứ giác [11]

Sau khi lưới được làm mềm thì bề mặt lưới của mô hình vẫn còn khá lớn, mà việc tối ưu hóa mô hình là làm cho mô hình trở nên ít lưới hơn nữa. Vì thế công việc tiếp theo của chúng ta là sẽ làm tinh giảm tối đa số lưới trên mà vẫn đảm bảo bề mặt của mô hình đồng thời vẫn giữ được cấu tạo số lưới tối thiểu để hỗ trợ cho các chuyển động của mô hình hay liên kết giữa các thành phần của mô hình nhất là ở các vị trí khớp. Công việc này thường thì chúng ta sẽ làm bằng tay trên các phần mềm 3D, việc này tiêu tốn khá nhiều thời gian. Để khắc phục vấn đề này chúng ta sẽ tạo một thuật toán tối ưu hóa bề mặt lưới tứ giác 3D một cách tự động mà vẫn có thể đáp ứng được yêu cầu đề ra ở trên.

Hiện nay, các công trình nghiên cứu về tối ưu hóa trên bề mặt lưới tứ giác vẫn đang còn hạn chế, do độ phức tạp trong tính toán bề mặt mô hình. Công việc tối ưu hóa vẫn thường được các nhà thiết kế làm việc trực tiếp bằng tay trên các mô hình dựa

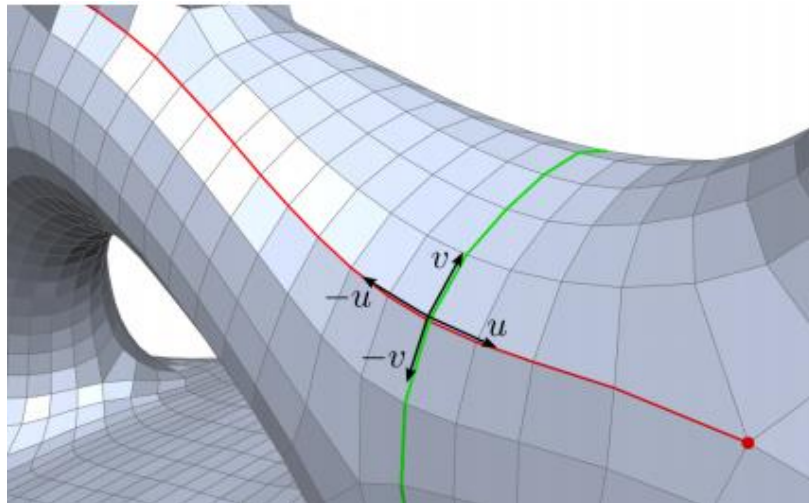
vào các phần mềm 3D. Sau đây chúng ta sẽ trình bày một phương pháp tối ưu hóa phổ biến hiện nay.



Hình 2.25. Mô hình lưới chuẩn để tạo chuyển động cho cánh tay

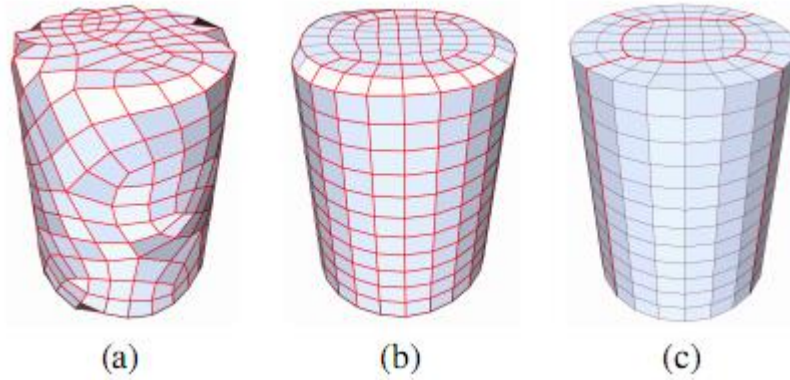
❖ **Phương pháp tối ưu hóa lưới tứ giác Quadrilateral Meshes[11]:**

Phương pháp tối ưu hóa lưới tứ giác dựa trên phương pháp “Global Structure Optimization of Quadrilateral Meshes” của David Bommes, Timm Lempfer Leif Kobbelt; Volume 30 (2011), Number 2; RWTH Aachen University.



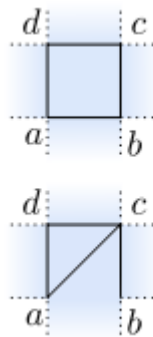
Hình 2.26. Mỗi đỉnh thường xuyên gây ra một hệ trục tọa độ tự nhiên bằng cách truy cập chiều kim đồng hồ ghi nhãn các cạnh đi ra là  $u$ ,  $v$ ,  $-v$ ,  $-u$ . Dòng tham số, như thể hiện trong màu đỏ và màu xanh lá cây có thể được mở rộng cho đến khi họ kết thúc trong một miền cá biệt (điểm màu đỏ).

Để thuật toán được chính xác thì khi làm mềm bề mặt lưới chúng ta nên đưa các lưới về lưới cơ sở của khối, hình (c), như sau:



Hình 2.27. Sự khác nhau của bề mặt đối tượng với số lưới bằng nhau  
 Quad loop: là việc chọn các cạnh liên tiếp nhau theo dạng đường kéo dài, chứa các cạnh mà với 2 cạnh liên tiếp sẽ cùng cắt 2 cạnh chứa điểm chung với nó.

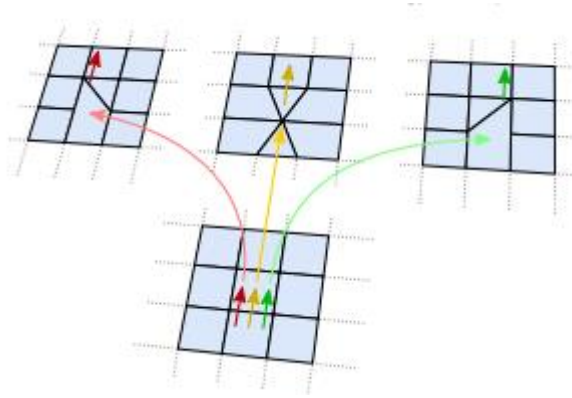
Giả sử rằng chúng ta có một lưới tứ giác khép kín không biên giới và chúng ta muốn thay đổi các kết nối trong một tứ giác duy nhất với các điểm a,b,c,d sao cho a được kết nối với c thay b, như mô tả trong hình sau.



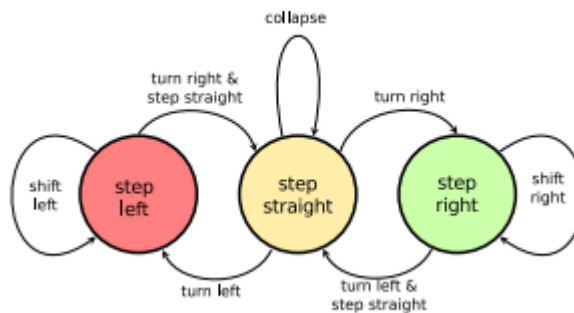
Hình 2.28 Thể hiện sự chuyển cạnh

Vấn đề là sau khi thực hiện lật(thay) cạnh này, kết thúc chúng ta có một tam giác và một hình ngũ giác mới. Nếu quad-loop tương ứng là lựa chọn tự do, một giải pháp sẽ được đề đạt lật cạnh dọc theo toàn bộ quad-loop như vậy mà cuối cùng các tam giác và ngũ giác sẽ bị hủy bỏ. Nhưng không phải tất cả quad-loops là giao điểm tự do và thậm chí nếu chúng được tự do, hoạt động kết hợp này là hoàn toàn xác định bởi cấu trúc quad-loop và không tự do để kiểm soát các khu vực của lưới tốt hơn cần được sửa đổi. Cách này là mâu thuẫn với yêu cầu để bảo vệ các bộ phận của lưới mà có các tính năng quan trọng hoặc khu vực có chất lượng tốt.

Để có được nhiều bậc tự do, chúng ta kết hợp các hoạt động lật cạnh trên với một hoạt động gộp theo cách như vậy, chúng ta có thể tạo ra một loạt lớn hơn nhiều của các toán tử, nhưng vẫn có thể đảm bảo để bảo vệ cấu trúc tứ giác của lưới đầu vào. Hình 2.29 cho thấy ba hoạt động nguyên tử cần thiết, cụ thể là lật chuyển trái, gộp, lật phải



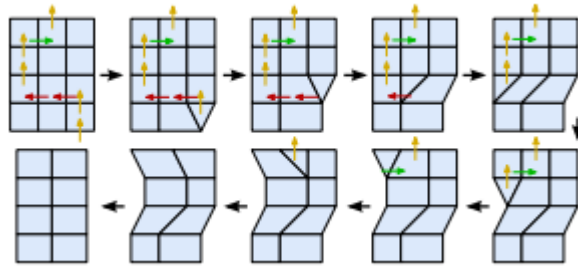
Hình 2.29. Ba hoạt động của việc gộp và tách kết nối của các điểm và tương ứng là hình tượng với một mũi tên màu đỏ, màu vàng và màu xanh lá cây tương ứng. Cả ba hoạt động có thể được liên kết với một half-edge kép và kết hợp cùng một đường kép tự động hữu hạn trong hình 2.30 để tạo thành một toán tử lưới bảo quản hợp lệ (grid-preserving operator :GP-toán tử). Các đặc tính quan trọng nhất của một GP - toán tử như vậy là nó không sinh thêm đỉnh cá biệt mới hoặc các yếu tố không phải tứ giác.



Hình 2.30. Tập mô tả hữu hạn hợp lệ các khả năng để kết hợp ba hoạt động việc gộp và tách kết nối của các điểm

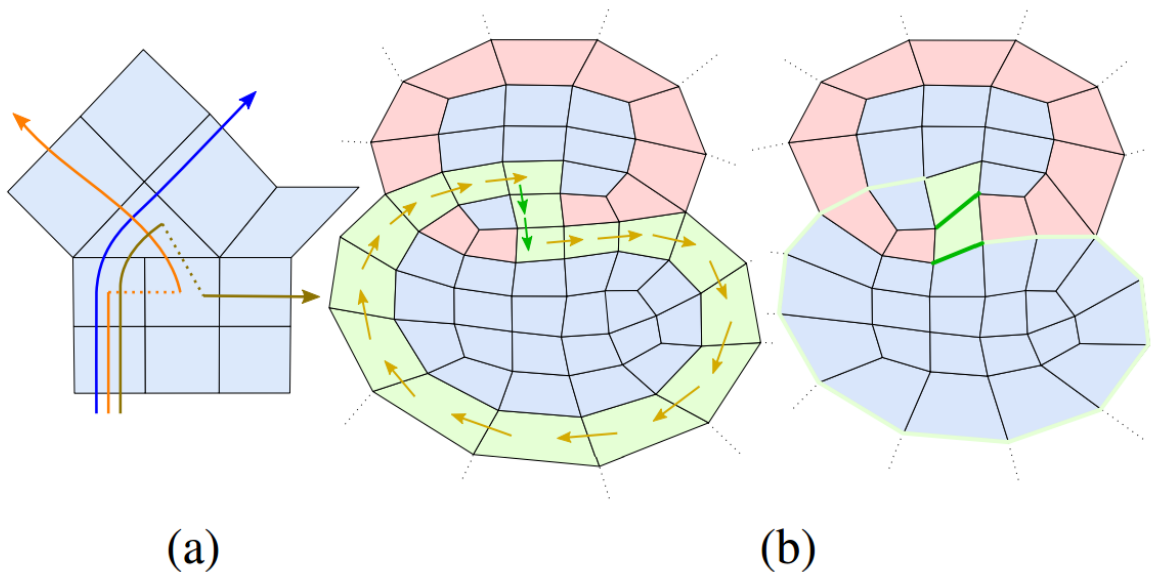
Điều này có nghĩa, nếu chúng ta bắt đầu từ một cạnh của lưới trong trạng thái bước trái,

chúng ta có thể làm tương tự đối bước rời trái như chúng ta muốn bằng cách đi theo con đường kép trong cùng một hướng, nơi tất cả các cạnh chéo được chuyển. Để thoát khỏi trạng thái bước trái, trong vòng một mặt chúng ta có thể rẽ phải và thay đổi trạng thái để dịch chuyển thẳng. Từ đây chúng ta có thể di chuyển thẳng để gộp nhiều cạnh như mong muốn, hoặc rẽ phải để áp dụng các toán tử phải, hoặc một lần nữa rẽ trái và áp dụng các thay đổi trạng thái trái. Nhìn chung, sử dụng máy trạng thái này, chúng ta có thể đi qua một con đường kép được lắp ráp các bước thẳng, các bước bên trái và các bước bên phải, nhưng chúng ta không bao giờ có thể bước trở lại, tức là bước hai lần vào cùng một vị trí (Hình 2.31).



Hình 2.31. Ví dụ về việc kết hợp 3 hoạt động trên và làm mềm kết quả

Lý do là chúng ta có thể khai thác các trường hợp ngoại lệ trong lưới để thay đổi hướng đi, ví dụ đi xung quanh một đỉnh có 2 cạnh kề nhau, như quay một góc bằng  $\pi$  ( $180^\circ$ ) trong một lưới liên tục. Do đó, điều hướng giữa và xung quanh đỉnh cá biệt cung cấp một lượng lớn các con đường có thể. Hình 2.32(a) đưa ra một ví dụ về hoạt động này.



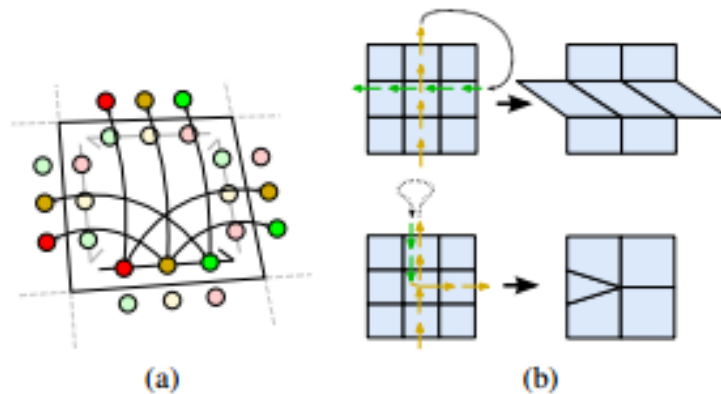
Hình 2.32. Đường nét đứt kiểm soát hướng đi bằng cách điều hướng giữa các lưới cá biệt

Để đảm bảo rằng trong cuối cùng tất cả các tam giác và ngũ giác được hủy bỏ, cần thiết phải có đường dẫn kép được đóng trong bộ trạng thái, có nghĩa là có một quá trình chuyển đổi từ trạng thái tại cuối cùng hai nửa cạnh để trạng thái kép tại nửa cạnh đầu tiên. Chú ý rằng điều này là hoàn toàn đúng khi đóng kép vòng của đường. Để minh họa, hình 2.32(b) cho thấy đó là một đường và tứ giác hình thành sau khi áp dụng các hành động tương ứng.

Nếu chúng ta muốn thay đổi các cạnh giữa đỉnh a và đỉnh b (hình 2.24) thành cạnh giữa đỉnh a và đỉnh c, trong khi duy trì không thêm một tứ giác cá biệt, chúng ta có thể bắt đầu tại cạnh giữa đỉnh a và đỉnh b với sự chuyển đổi trạng thái sang phải và đi dọc theo đường nào kép kín tương thích với máy trạng thái và thực hiện các hành động. Những hành động tốt nhất phụ thuộc nhiều vào các bước đã nhớ.

Một sự lựa chọn ngẫu nhiên là để giảm thiểu chi phí, nghĩa là số lượng các hoạt động toán tử thêm đó là cần thiết để đóng các đường đi. Điều này có thể làm được bằng cách liệt kê tất cả các đường tạo ra bởi các trạng thái, với việc tăng thời gian cho đến khi chu trình ngắn nhất được tìm thấy. Rõ ràng phương pháp này dẫn đến việc phức tạp hàm mũ, khó khăn cho các ứng dụng thực tế.

Đồ thị trạng thái: Để tìm thấy hiệu quả một chu trình là hợp lý với máy trạng thái đầu tiên chúng ta thử với một đồ thị có hướng, như mô tả trong hình 2.33(a). Trong biểu đồ này tất cả các chu trình là tương thích với các trạng thái. Ý tưởng là đồ thị sở hữu ba đỉnh khác nhau cho mỗi half-edge kép của lưới tứ giác đó để mã hóa ba trạng thái khác nhau. Thêm cạnh chỉ đường mà tái tạo các hiệu ứng chuyển tiếp của bộ trạng thái như minh họa trong hình 2.33(a), chúng ta đạt được một đồ thị có hướng với các kết quả mong muốn. Tất cả các chu trình trong đồ thị này tương ứng với đường dẫn kép trên lưới tứ giác được đóng trong lưới cũng như trong bộ trạng thái.



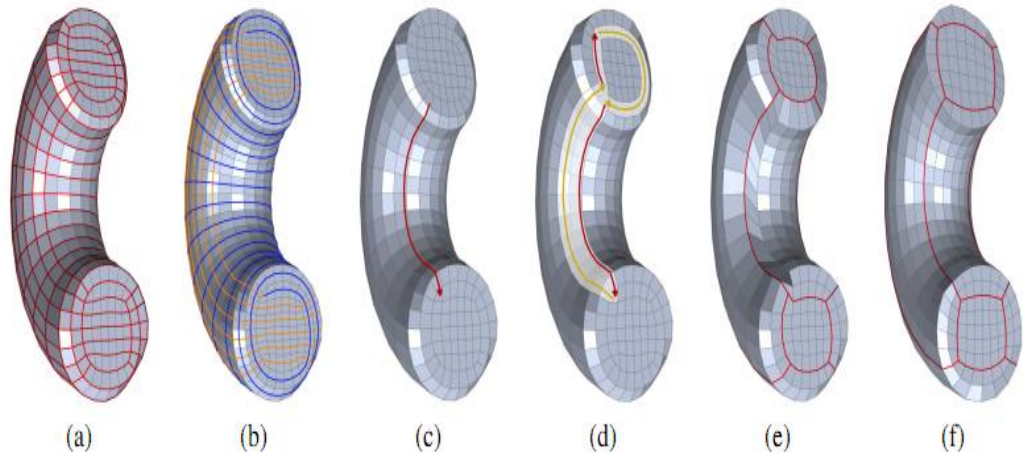
Hình 2.33 Đường đi một chu trình

Trong hình vẽ này một chu trình ngắn nhất thông qua một đỉnh đầu có thể được tìm thấy bởi việc tìm kiếm theo chiều rộng đơn giản và hiệu quả. Từ hình vẽ là tĩnh, và ta thấy được những thay đổi được thực hiện bởi các hành động trước đó của cùng một đường đi. Rõ ràng chúng ta không thể thay đổi một cạnh được gộp hoàn toàn, mặc dù một đường đi như vậy tồn tại trong đồ thị. Vì vậy chúng ta phải làm một bài đánh giá của chu kỳ để kiểm tra xem nó thuộc về một tập thực hiện các hành động hay không. Nếu nó không có khả năng để thực hiện, chúng ta lặp đi lặp lại để sửa đổi các đồ thị và thực hiện tìm kiếm mới, cho đến khi đã tìm thấy một chu kỳ hợp lệ hoặc thuật toán kết thúc mà không tìm thấy một trường hợp nào.

Trường hợp không hợp lý trong một chu kỳ thường được gây ra bởi một đường đi kép tương ứng trên lưới tứ giác mà việc đi thăm bề mặt mô hình nhiều hơn một lần, ví dụ bằng cách thực hiện nhiều hơn một hoạt động trên một cạnh duy nhất hoặc quay đầu tại một cạnh và sau đó thực hiện bất kỳ hành động khác khi đi qua bề mặt. Chỉ có hai trường hợp ngoại lệ, mà nó được phép vào thăm một bề mặt hai lần được gộp đầu tiên qua một bề mặt trong hai trục giao hướng và gộp lần thứ hai thông qua một mặt

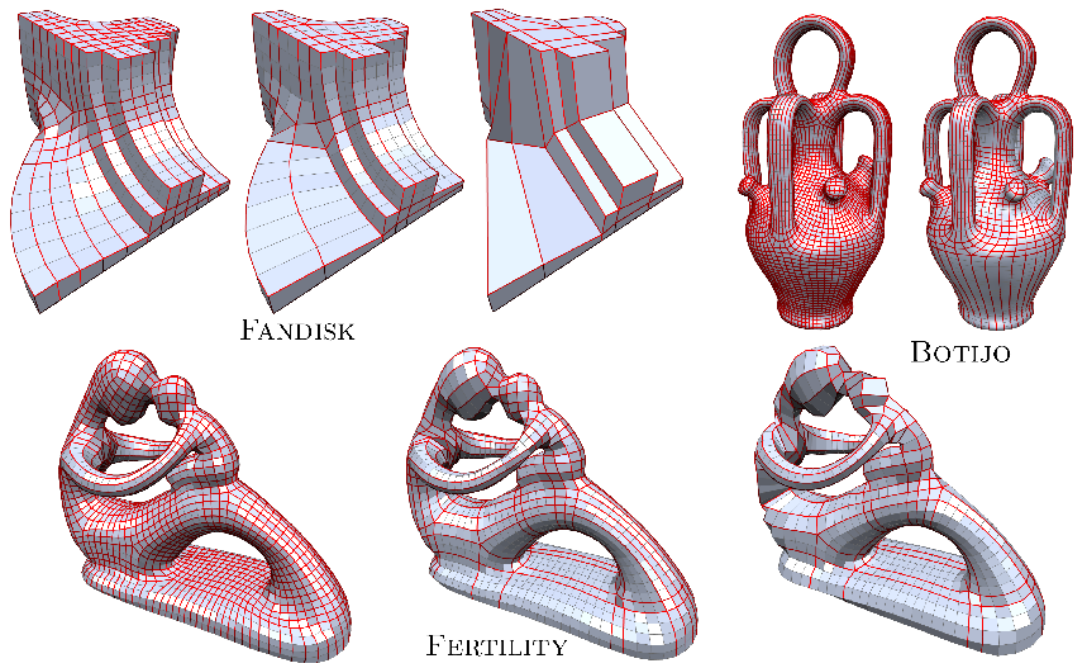
trong một hướng, sau đó chuyển qua mặt theo hướng trục giao. Trong cả hai trường hợp, các cấu trúc đồ thị tĩnh vẫn dẫn đến đường đi hợp lệ.

Nếu một chu trình không hợp lý được tìm thấy, đầu tiên chúng ta xác định các trường hợp không lý nơi một bề mặt được đến thăm hai lần, dẫn đến một cặp đồ thị đỉnh  $v_i$  và  $v_j$ , trong việc truy cập  $v_i$  đầu tiên. Để sửa đổi các đồ thị, chúng ta loại bỏ tất cả các đỉnh đồ thị và các cạnh liền kề đó là không tương thích cho các đường đi lên đến đỉnh  $v_i$  và sau đó khởi động lại tìm kiếm từ  $v_i$



Hình 2.34. Sử dụng Quad-loop toàn cục

Một số kết quả đạt được khi áp dụng phương pháp Quadrilateral Meshes:



Hình 2.35. Các mô hình trước và sau khi được tối ưu



## CHƯƠNG 3. THỰC NGHIỆM VÀ ỨNG DỤNG TỐI ƯU MÔ HÌNH 3D

### 3.1. Yêu cầu thực nghiệm, ứng dụng

Các ứng dụng của thực tại ảo phục vụ cho nhiều lĩnh vực khác nhau của đời sống, xã hội v.v.. Và khi yêu cầu đó tăng thì số lượng mô hình hóa cho việc các dữ liệu trong thực tại ảo tăng lên đáng kể, số lượng mô hình tăng lên sẽ ảnh hưởng vô cùng lớn đến tốc độ xử lý của máy tính, điện thoại v.v. hay tốc độ truyền của mạng. Tóm lại, có thể chia mô hình 3D thành 2 loại đó là các mô hình tĩnh và các mô hình động. Mô hình tĩnh gồm nhà, đồi, đá, tượng v.v. các mô hình này sẽ được tối ưu hóa theo lưới tam giác. Còn với các mô hình động gồm con người, con vật v.v. là các thực thể cần được tối ưu hóa theo lưới tứ giác, vì các chuyển động ảnh hưởng hiệu đến bề mặt lưới, đặc biệt là tại những vùng khớp và cơ. Tùy vào ứng dụng khác nhau, chúng ta sẽ sử dụng mô hình nào là hợp lý, hoặc chúng ta có thể sử dụng cả 2 loại mô hình.

Chương 3 của luận văn ứng dụng kết quả nghiên cứu trong trong chương 2 từ đó xây dựng chương trình cài đặt thử nghiệm thuật toán. Qua đó, đánh giá sự chính xác của các phần nội dung lý thuyết được trình bày tại chương 2 và minh chứng khả năng ứng dụng của những tìm hiểu nghiên cứu được báo cáo trong luận văn.

#### 3.1.1. Yêu cầu với thực nghiệm

Với mục tiêu là tối ưu hóa lưới của mô hình 3D, chương trình thực nghiệm phải đảm bảo tính năng chính là tối ưu được mô hình về số lượng lưới nhưng vẫn đảm bảo hình dáng và chất lượng hình ảnh khi render. Để tối ưu được mô hình, công việc đầu tiên là phải đọc và hiển thị được mô hình. Quá trình đọc và hiển thị giúp người vận hành phần mềm có thể thao tác và nhìn thấy kết quả trên phân mềm ứng dụng.

Tiếp đó, chương trình thực nghiệm cần xây dựng các thanh phần giúp hiển thị và tương tác 3D với mô hình đang được chọn (các tính năng xoay, phóng to, thu nhỏ v.v. mô hình). Để đảm bảo tính khách quan chương trình thực nghiệm cần cho phép người vận hành chọn lựa nhiều mô hình khác nhau trong khi thực nghiệm.

Đặc biệt, phần quan trọng không thể thiếu khi xây dựng chương trình là modun tối ưu mô hình, ở đó với một mô hình được chọn làm đầu vào, người vận hành có thể lựa chọn các tham số từ đó sinh ra một mô hình mới được tối ưu hơn so với mô hình được chọn ban đầu.

Các thành phần không thể thiếu đối với phần mềm là hệ thống giao diện và các tính năng khác kèm theo như chương trình sau khi biên dịch, các lựa chọn chạy trên hệ điều hành v.v..

### 3.1.2. Kiểm tra các mô hình đầu vào

Các mô hình để test là các mô hình dưới dạng 3D, có số lượng lưới bề mặt lớn. Một số mô hình đã được thử nghiệm cho các bài thử nghiệm trong các công trình tối ưu hóa trước đó, ví dụ như mô hình: Con thỏ (như hình 3.1)



*Hình 3.1. Mô hình con thỏ trong các bài thử nghiệm tối ưu lưới*

Một số mô hình còn lại tôi tự tạo ra bằng các phần mềm 3D hoặc sưu tập trên internet. ...



*Bình gốm*



*Lọ cây nhỏ*



*Ghê tựa*

*Hình 3.2: Các mô hình khác sử dụng trong chương trình*

### 3.2. Phân tích, lựa chọn công cụ

Dựa trên những nghiên cứu về các kỹ thuật tối ưu hóa mô hình đã có, trong luận văn sử dụng phương pháp Curvature, kết hợp với các ràng buộc khi phát hiện một số trường hợp ngoại lệ của điểm trên bề mặt lưới khi áp dụng công thức để xóa điểm trong mô hình lưới tam giác. Mô hình mới sinh ra có số lượng lưới bề mặt giảm đáng kể mà vẫn giữ được nguyên mẫu về hình dáng.

Đầu vào của chương trình mô phỏng là các mô hình 3D có kích thước lưới lớn cần được giảm thiểu. Khi một mô hình được đưa vào thì chúng ta sẽ tính được số lượng lưới tam giác trên bề mặt chúng. Trên thực tế các mô hình có thể từ máy quét 3D, hay được mô hình hóa bởi các phần mềm nhưng có số lượng lưới lớn. Các tham số của từng mô hình là số lượng điểm, cạnh, ta suy ra được số lượng mặt. Tùy vào mô

hình khác nhau mà việc tính toán tối giản bề mặt lưới có thể thay đổi trong chương trình.

Một vấn đề quan trọng khi xây dựng một chương trình tối ưu hóa bề mặt lưới chính là nền tảng đồ họa và ngôn ngữ lập trình đồ họa. Trong nội dung luận văn sử dụng ngôn ngữ lập trình Visual C# và nền tảng lập trình đồ họa thuộc bộ thư viện của Unity. Đây là một sản phẩm của Engine Game cung cấp cho người dùng một số phương thức đồ họa cơ bản: có thể tải một bản thiết kế 2D, 3D vào hệ thống, vẽ các ảnh 2D và 3D, phương thức lập trình, tương tác với card đồ họa v.v.

Để xây dựng chương trình tối ưu hóa với công cụ được mô tả người lập trình phải thực hiện các công việc: xây dựng lớp quản lý các đối tượng 3D trong chương trình, xây dựng các hàm tính số lượng lưới tam giác của các đối tượng, lập trình tính toán tối ưu hóa bề mặt lưới theo việc xóa điểm và phục hồi bề mặt theo phần trăm của tổng số lưới đã có. Với lựa chọn môi trường chuyên dụng như Unity tạo điều kiện để người lập trình hiểu sâu hơn về hệ thống và tương tác với card đồ họa.



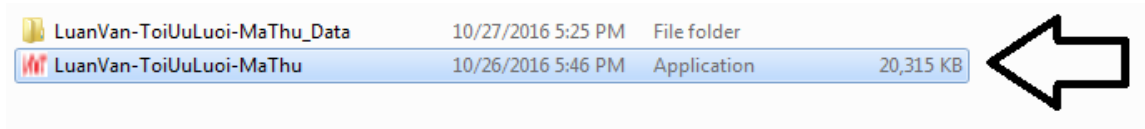
Hình 3.3. Ảnh chụp chương trình thực nghiệm

### 3.3. Một số kết quả thực nghiệm tối ưu mô hình

Như đã trình bày, chương trình này được cài đặt với ngôn ngữ lập trình Visual C# và công cụ hỗ trợ là Engine Unity 3D, các mô hình đầu vào được định dạng với phần mở rộng là \*.fbx.

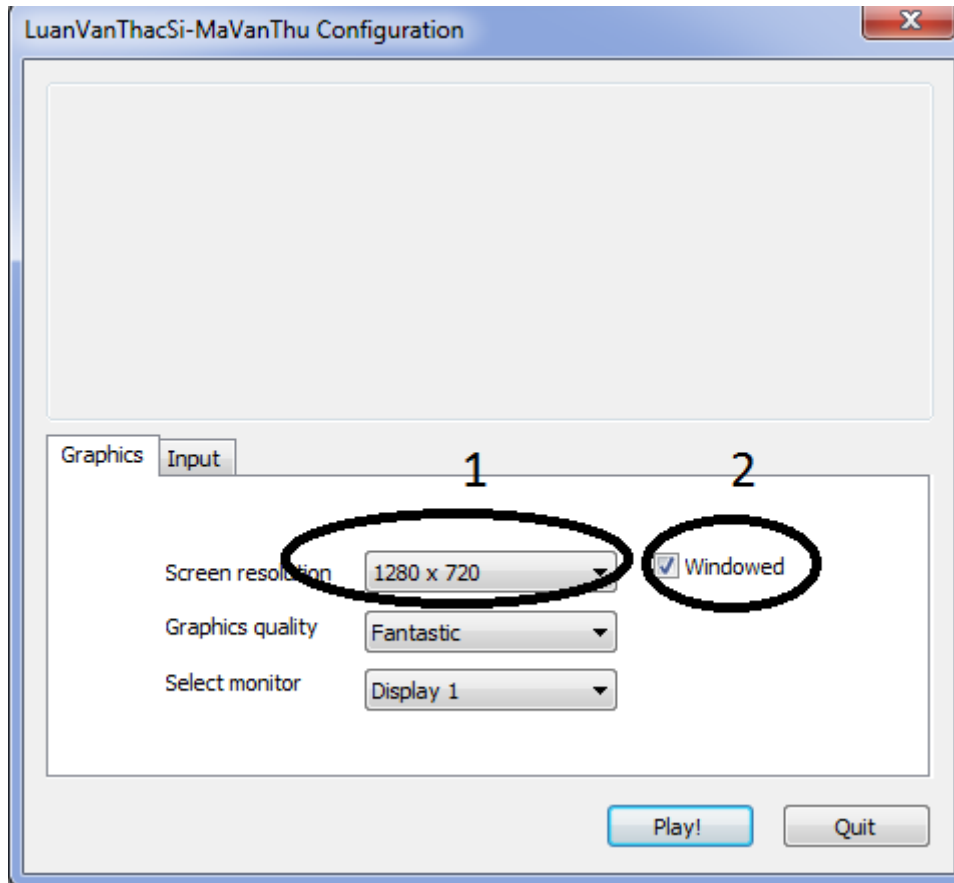
#### 3.3.1. Hướng dẫn sử dụng chương trình thực nghiệm

Chạy file Luanvan-tôiuuluioi .exe của để mở chương trình hình 3.4



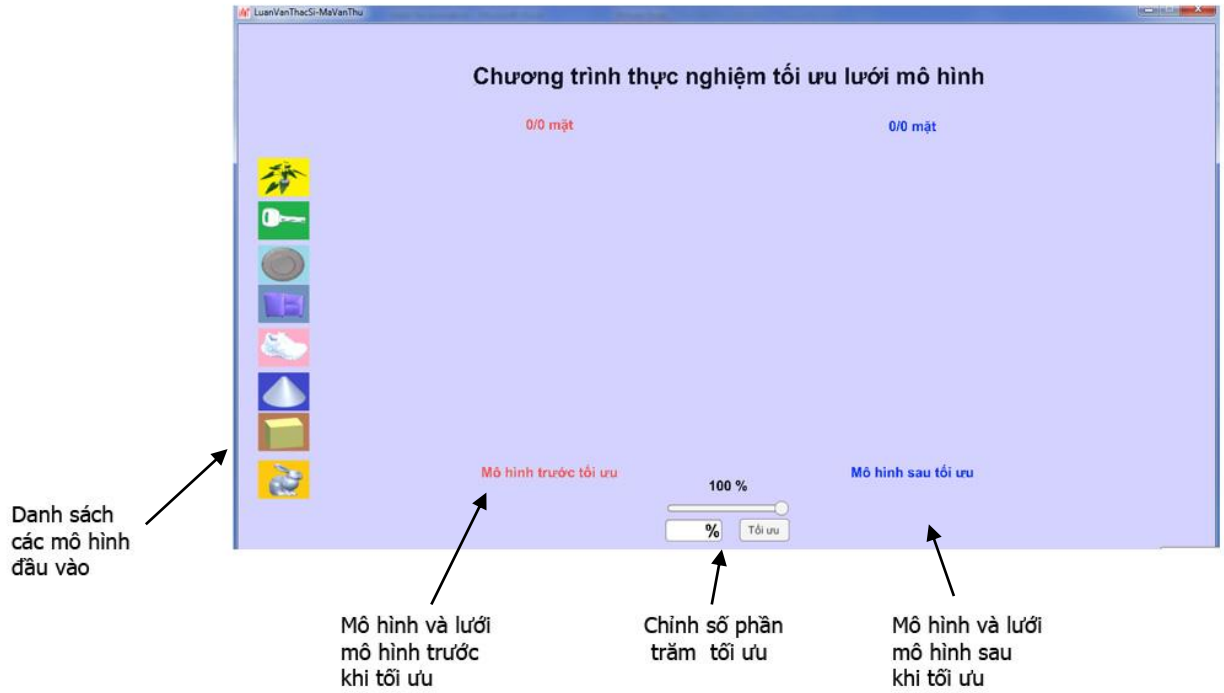
*Hình 3.4. Chạy file chương trình*

Sau đó, một bảng lựa chọn cài đặt màn hình chương trình hiện ra. Trong hình 3.5 chúng ta chọn chế độ phân giải của màn hình chương trình và chọn chế độ cửa sổ bằng cách tích vào ô “windowed”.



*Hình 3.5. Cửa sổ lựa chọn màn hình chạy chương trình*

Trong chương trình, chúng ta sẽ lựa chọn mô hình bằng cách kích vào mô hình đó. Sau đó chúng ta sẽ lựa chọn phần trăm tối ưu bằng cách gõ vào ô bên dưới, hoặc chúng ta có thể kéo trên thanh trượt để lựa chọn % tối ưu (Hình 3.6)



Hình 3.6. Tổng quan về chương trình

### 3.3.2. Một số kết quả tối ưu mô hình trên chương trình thực nghiệm



7496 Mặt



5526 Mặt



94160 Mặt



69839 Mặt



69630 Mặt



51644 Mặt



49314 Mặt



36984 Mặt

*Hình 3.7 Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 75%*  
 Nhận xét: với tham số tối ưu là 75% thì mô hình sau tối ưu vẫn giữ được hình dạng tương tự mô hình ban đầu



7496 Mặt



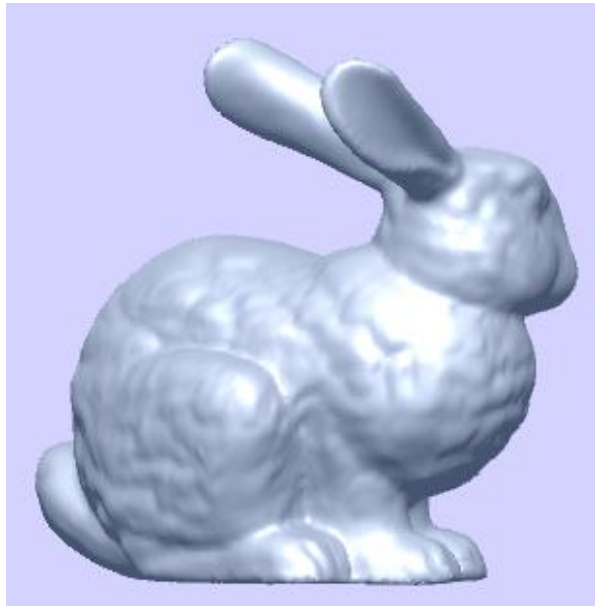
3556 Mặt



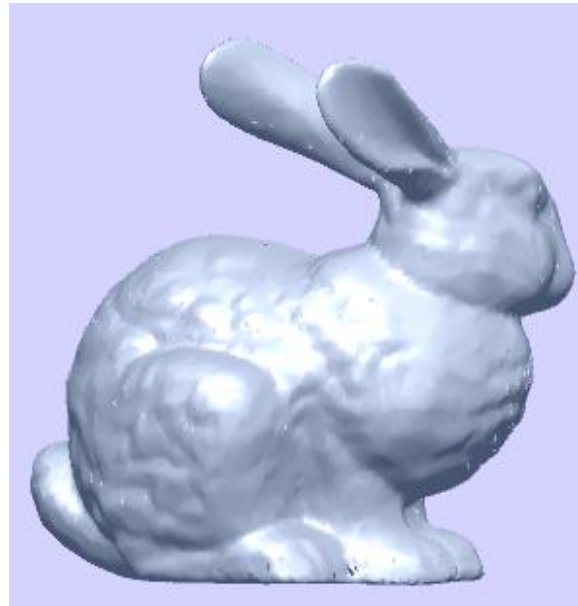
94160 Mặt



45625 Mặt



69630 Mặt



34632 Mặt



49314 Mặt

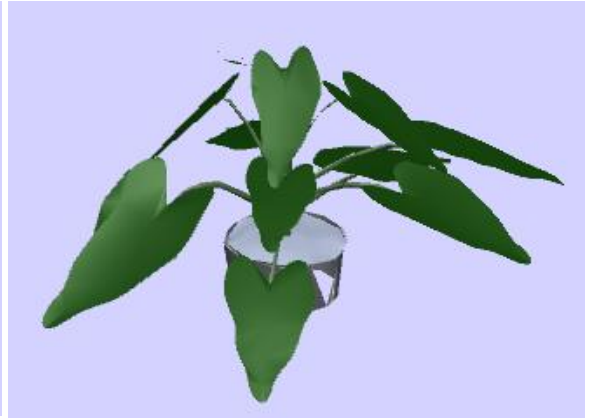


24656 Mặt

*Hình 3.8. Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 50%*  
 Nhận xét: với tham số tối ưu là 50% thì mô hình đã có chút khác biệt về lưới của mô hình sau tối ưu



7496 Mặt



1576 Mặt



94160 Mặt



21407 Mặt



69630 Mặt



17624 Mặt





49314 Mặt

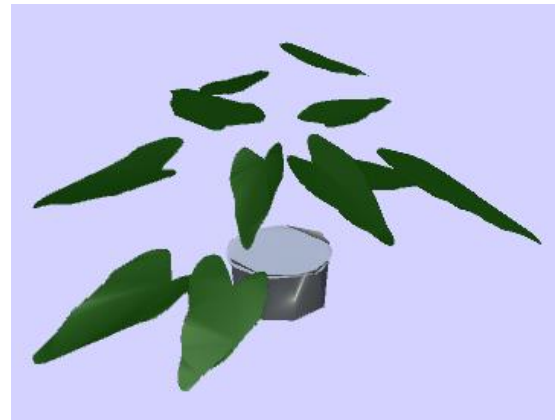


12326 Mặt

*Hình 3.9. Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 25%*  
 Nhận xét: với tham số tối ưu là 25% thì bề mặt đã bị biến đổi, tuy nhiên vẫn chấp nhận được so với mô hình gốc.



7496 Mặt



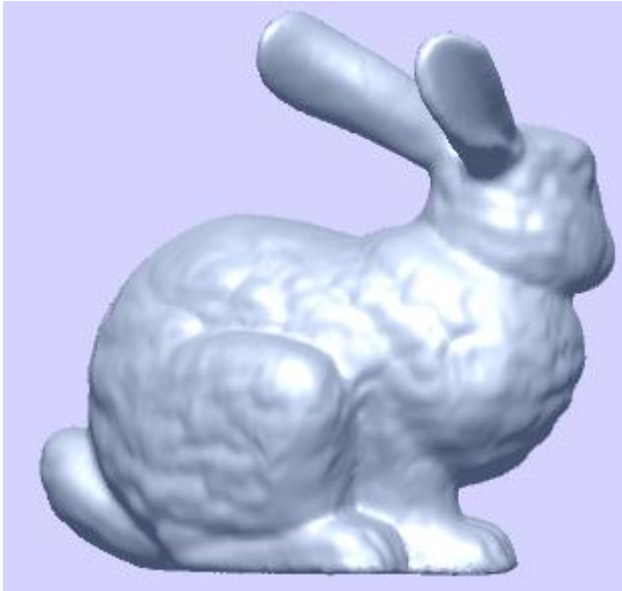
918 Mặt



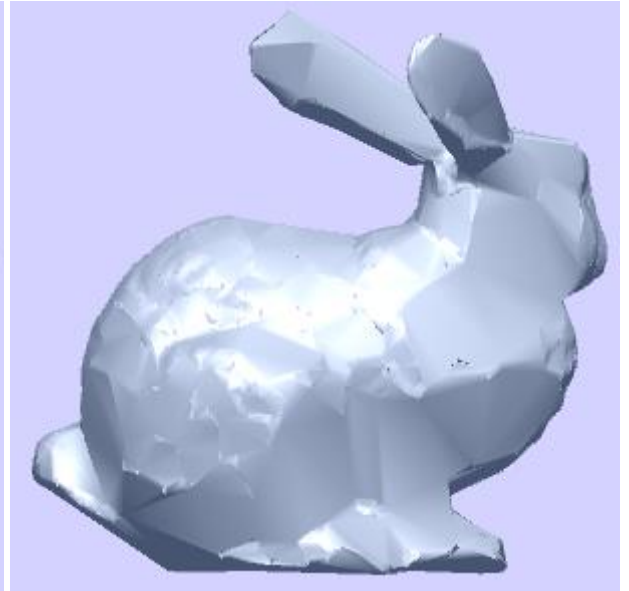
94160 Mặt



8989 Mặt



69630 Mặt



11612 Mặt



49314 Mặt

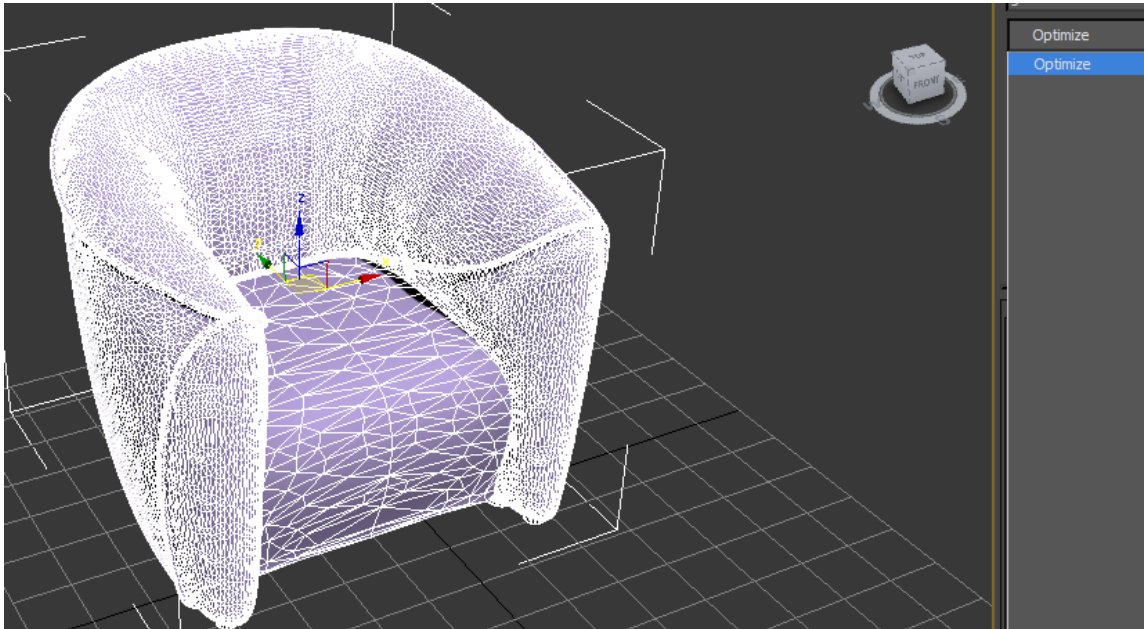


4922 Mặt

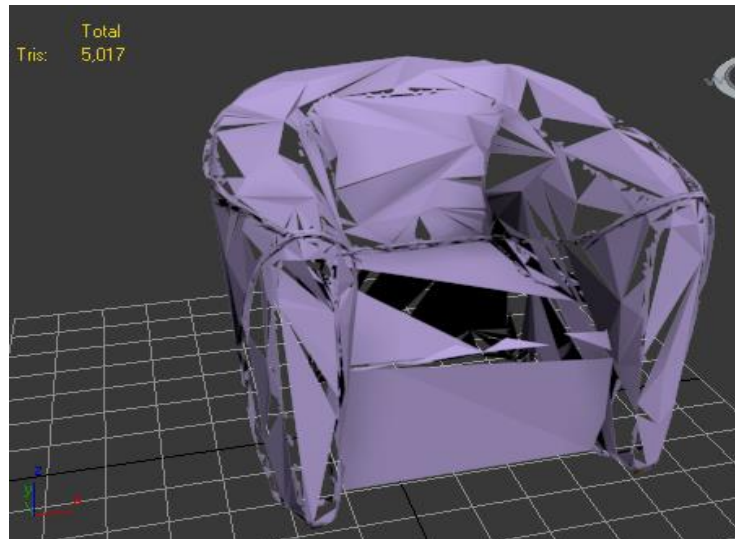
*Hình 3.10. Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là 10%*  
 Nhận xét: với tham số tối ưu là 10% thì sự khác biệt rõ giữa mô hình trước và sau khi thực hiện tối ưu. Tuy nhiên nhiều chỗ của mô hình bị méo mó nặng.

Chốt lại với tham số là 25% thì mô hình vẫn giữ được hình ảnh tốt của mô hình sau khi tối ưu.

So sánh kết quả tối ưu với 3Ds max, khi lựa chọn plugin Optimize



*Hình 3.11. Lưới ghế tựa trước khi tối ưu với số lưới là 49314 mặt*



*Hình 3.12. Lưới ghế tựa sau khi tối ưu 10% với số lưới là 5.017 mặt tam giác*

Trong 3Ds max thì việc sử dụng lệnh tối ưu là khá khó khăn, bên cạnh đó chất lượng lưới khi giảm làm cho bề mặt mô hình trở nên méo mó, rất khó có thể sử dụng được.

## KẾT LUẬN

Luận văn “ Mô hình 3D và tối ưu hóa mô hình trong thực tại ảo” với nội dung nghiên cứu chính là các kỹ thuật tối ưu hóa mô hình 3D lưới tam giác, tứ giác và các ứng dụng tương ứng của nó trong thực tại ảo. Qua đó người đọc có được các tri thức tổng quan về cấu tạo mô hình 3D trong thực tại ảo, các phương pháp để tạo ra các mô hình phổ biến hiện nay. Đồng thời có cái nhìn sâu sắc hơn về mô hình 3D trong thực tại ảo, tầm quan trọng của nó trong khoa học, kỹ thuật và đời sống, cũng như các ưu, nhược điểm các mô hình 3D.

Tiếp đó luận văn đã trình bày bài toán tối ưu hóa mô hình 3D, các đặc điểm mô hình cũng như đầu vào và đầu ra cả bài toán này. Sau đó là các kỹ thuật có thể áp dụng trong bài toán tối ưu hóa lưới tam giác và lưới tứ giác. Với phương pháp tối ưu hóa mô hình lưới tam giác, luận văn trình bày và phân tích rõ cách thức thực hiện của hệ thống cũng như chi tiết hóa các tham số khi tối ưu hóa lưới. Trong đó đã tổng hợp với 8 mẫu mô hình được sử dụng cho việc thử nghiệm tối ưu mô hình. Trong đó là một số mô hình được sử dụng lại trong các nghiên cứu trước, còn lại là các mô hình mới do tôi tạo ra.

Luận văn chỉ ra những hạn chế khi tối ưu hóa bằng phương pháp thông thường thông qua một số trường hợp ngoại lệ để chúng ta có thể loại bỏ hoặc giữ lại các điểm là cần thiết cho mô hình, tuy nhiên khi chúng ta thêm các ràng buộc thì độ phức tạp của thuật toán được tăng lên. Kết quả thực nghiệm cho thấy hình ảnh của đối tượng chấp nhận được khi chúng ta giảm tương đối số lưới trên bề mặt.

Cuối cùng, luận văn trình bày so sánh những mô hình kết quả nghiên cứu với plugin tối ưu hóa đã được tích hợp sẵn trong 3Ds Max, ... Chương trình mô phỏng việc tối ưu này giúp người dùng mô hình 3D nhẹ nhàng hơn nhờ việc tối ưu, cũng như giúp các chương trình ứng dụng thực tại ảo chạy với chế độ thời gian thực hơn. Tuy nhiên vẫn cần những nghiên cứu để tiếp tục cải tiến nâng cao hình ảnh của mô hình mà số lượng lưới bề mặt là thấp nhất có thể. Nhất là việc tối ưu hóa cho mô hình lưới tứ giác, hỗ trợ cho việc tạo chuyển động mà các hình không bị méo mó, một công việc làm khiến các công ty tốn rất nhiều thời gian và tiền bạc để các nhà thiết kế làm việc bằng tay.

Bài toán tối ưu hóa bề mặt lưới mô hình là bài toán có nhiều ý nghĩa trong khoa học, công nghệ và đời sống, điều đó thúc đẩy các nghiên cứu về mô phỏng và thực tại ảo tiếp tục phát triển. Tác giả hi vọng luận văn này sẽ đóng góp một phần cho những phát triển của ngành công nghệ thực tại ảo trên máy tính, điện thoại thông minh,.. nói riêng và công nghệ thông tin nói chung. Rất cảm ơn những nhà nghiên cứu, quý thầy cô và quý vị đã quan tâm và bỏ thời gian đọc luận văn này.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

- [1]. Nguyễn Văn Huân, Vũ Đức Thái (2006), *Kỹ thuật lập trình mô phỏng thế giới thực dựa trên Morfit*, Nhà xuất bản Khoa học & Kỹ thuật.
- [2]. Lê Sơn Thái (2014), *Nghiên cứu một số kỹ thuật tạo hiệu ứng khói trong thực tại ảo*, Luận văn thạc sỹ, Trường ĐH Công nghệ, ĐH QG Hà nội.
- [3]. Đỗ Phú Duy(2012), *Xây dựng bề mặt lưới từ tập điểm 3D và phương pháp chia nhỏ bề mặt lưới*, Luận văn thạc sỹ, Trường Đại học Đà Nẵng.

### Tiếng Anh

- [4]. J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans (2001), *Reconstruction and Representation of 3D Objects with Radial Basis Functions*.
- [5]. Enrique Valero, Antonio Adan (2012), *Automatic Construction of 3D Basic-Semantic Models of Inhabited Interiors Using Laser Scanners and RFID Sensors*.
- [6]. David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, Robert Huebner (2002), *Level of detail for 3D graphics*, Morgan Kaufmann.
- [7]. Nira Dyn, Kai Hormann, Sun-Jeong Kim, and David Levin (2000), *Optimizing 3D Triangulations Using Discrete Curvature Analysis*.
- [8]. L. Kobbelt (1997), *Discrete fairing*, in *The Mathematics of Surfaces VII*, T. Goodman and R. Martin (eds.), Clarendon Press, Oxford, P101–131.
- [9]. Steven J. Owen, Matthew L. Staten, Scott A. Canann and Sunil Saigal (1998), *Advancing Front Quadrilateral Meshing Using Triangle Transformations*,
- [10]. Scott A. Canann, Joseph R. Tristano, and Matthew L. Staten (1998), *An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes*.
- [11]. David Bommes, Timm Lempfer, Leif Kobbelt (2011), *Global Structure Optimization of Quadrilateral Meshes*.
- [12]. Sajid Hussain, Hakan Grahm and Jan Persson, *Feature preserving mesh simplification: A vertex cover approach*.
- [13]. H. Hoppe (1996), *Progressive Meshes*. *Computer Graphics*, (SIGGRAPH'96 Proceedings), P 99–108.
- [14]. H. Hoppe, T. Duchamp (1993), *Mesh Optimization*. *Computer Graphics*, (SIGGRAPH'93), P 19–26.
- [15]. P. Lindstrom., G. Turk (1998), *Fast and Memory Efficient Polygonal Simplification*, Proceedings of IEEE Visualization'98, P 279–286.
- [16]. W. J. Schroeder, J. A. Zarge, W. E. Lorensen (1992), *Decimation of Triangle Meshes*, *Computer Graphics* (SIGGRAPH'92 Proceedings), P 65–70.

- [17]. J. Cohen, A. Varshney, D. Manocha (1996), *Simplification Envelopes*. *Computer Graphics*, (SIGGRAPH'96 Proceedings), P 119–128.
- [18]. J. Rossignac, P. Borrel (1993), *Multi-resolution 3D Approximation for Rendering Complex Scenes*. *Modeling in Computer Graphics: Methods and Applications*, P 279–286.
- [19]. T. S. Gieng, B. Hamann, K. I. Joy (1997), *Smooth Hierarchical Surface Triangulation*, Proceedings of IEEE Visualization'97, P 379-386.
- [20]. B. Hamann (1994), *A Data Reduction Scheme for Triangulated Surfaces*, *Computer Aided Geometric Design*, P 197-214.
- [21]. D. P. Luebke (2001), *A Developer's Survey of Polygonal Simplification Algorithms*, IEEE Computer Graphics and Applications'01, P 24-35.
- [22]. C. Chang, S. K. Yang, D. Z. Duan, M. F. Lin (2002), *A Fuzzy Based Approach to Mesh Simplification*, *Journal of Information Science and Engineering* 18, P 459-466.
- [23]. Y. Wu, Y. He, H. Cai (2004), *QEM-based Mesh Simplification with Global Geometry Features Preserved*, *Computer Graphics (SIGGRAPH'04 Proceedings)*, P 50 –57.
- [24]. S. Mata, L. Pastor, A. Rodriguez (2006), *Attention Based Mesh Simplification using Distance Transforms*, *Lecture Notes in Computer Science (LNCS'06)*, Vol (4245), P 83-294.
- [25]. C. DeCoro, N. Tatarchuk (2007), *Real-time Mesh Simplification using GPU*. *Computer Graphics*, (SIGGRAPH'07 Proceedings), P 161–166.
- [26]. S. Hussain and H. Grahm (2007), *Fast kd-tree construction for 3d-rendering algorithms like ray tracing*, In Proc. Third International Symposium on Advances in Visual Computing, *Lecture Notes in Computer Science (LNCS'07)* , vol. 4842, P 681–690.
- [27]. M Roy, S. Foufou, F. Truchetet (2004), *Mesh comparison using attribute deviation metric*, *International Journal of Image and Graphics* , P 1–14.