

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

ĐỖ PHÚ DUY

**XÂY DỰNG BỀ MẶT LƯỚI TỪ TẬP HỢP ĐIỂM 3D VÀ
PHƯƠNG PHÁP CHIA NHỎ BỀ MẶT LƯỚI**

Chuyên ngành: Khoa học máy tính
Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng – Năm 2012

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: TS. NGUYỄN TẤN KHÔI

Phản biện 1:

Phản biện 2:

Luận văn được bảo vệ trước Hội đồng chấm Luận
văn tốt nghiệp thạc sĩ kỹ thuật họp tại Đại học Đà
Nẵng vào ngày tháng năm 2012

Có thể tìm hiểu luận văn tại:

- Trung tâm thông tin – Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. Lý do chọn đề tài

Trong những năm gần đây sự phát triển của đồ họa máy tính làm thay đổi hoàn toàn tương tác giữa người và máy, khi các kỹ thuật ứng dụng đồ họa ngày càng cao hơn nên có nhiều người quan tâm nghiên cứu đến lĩnh vực này. Nhờ đó mà các ứng dụng đồ họa trên máy tính ra đời như: phim hoạt hình, game và định hướng tương lai với các hệ thống thực tại ảo... đã đóng góp cho sự phát triển chung của ngành Công nghệ Thông tin. Do vậy, đồ họa máy tính trở thành một lĩnh vực hấp dẫn và có nhiều ứng dụng trong thực tế.

Biểu diễn bề mặt các đối tượng 3 chiều (3D) được coi là các bước khởi đầu trong một hệ thống mô phỏng thực tại ảo, góp phần tạo nên một hệ thống mô phỏng hoàn chỉnh. Một trong những cách tiếp cận biểu diễn bề mặt các đối tượng 3 chiều phổ biến hiện nay là dựa trên kỹ thuật biểu diễn bề mặt lưới.

Trong biểu diễn bề mặt ảo 3 chiều ngoài các vấn đề biểu diễn bề mặt đảm bảo chất lượng còn phải đáp ứng yêu cầu về tính đơn giản nhằm giảm thiểu không gian lưu trữ, rút ngắn thời gian biểu diễn bề mặt phục vụ cho các bước mô phỏng sau này. Việc xây dựng một hệ thống mô hình hóa là bao gồm xây dựng và biểu diễn bề mặt lưới các đối tượng 3 chiều nhằm tái tạo lại các mô hình thực tế trong không gian từ tập điểm đo được của đối tượng thực.

Với nhu cầu mô hình hóa bề mặt lưới tam giác từ tập hợp điểm đo được trong không gian 3 chiều, tối ưu bề mặt lưới để lưu trữ, tinh chỉnh một lưới rời rạc hội tụ về một bề mặt nhẵn.

Xuất phát từ nhu cầu thực tiễn như trên, tôi đề xuất đề tài luận văn cao học như sau:

“XÂY DỰNG BỀ MẶT LƯỚI TỪ TẬP HỢP ĐIỂM 3D VÀ PHƯƠNG PHÁP CHIA NHỎ BỀ MẶT LƯỚI”

2. Đối tượng, phạm vi, và phương pháp nghiên cứu

Đề tài tập trung vào nghiên cứu xây dựng bề mặt lưới 3D từ tập hợp điểm rời rạc và kỹ thuật chia nhỏ bề mặt nhằm cải thiện chất lượng bề mặt lưới được xây dựng và làm mịn mặt lưới. Để đạt được mục tiêu nghiên cứu đã đề ra, trong đề tài đã thực hiện cách tiếp cận nghiên cứu sau:

Trên cơ sở lý thuyết về đồ họa, tìm hiểu các vấn đề về mô hình hóa các đối tượng 3D, về các phương pháp biểu diễn bề mặt lưới 3D trên máy tính. Nghiên cứu các thuật toán xây dựng bề mặt lưới từ tập hợp điểm rời rạc trong không gian. Tìm hiểu các phương pháp chia nhỏ bề mặt lưới tam giác.

Từ việc nghiên cứu và tiếp thu các phương pháp, các thuật toán và các phần mềm chuyên dụng tiên tiến của thế giới trong lĩnh vực đồ họa trên máy tính. Tìm hiểu rõ các ưu, nhược điểm của từng phương pháp để đề xuất cách tiếp cận phù hợp cho đề tài. Sử dụng các công nghệ, các công cụ lập trình như: Ngôn ngữ lập trình Visual C++ 6.0, thư viện hỗ trợ xử lý đồ họa OpenGL.

3. Mục đích của đề tài

Tìm hiểu cách biểu diễn mô hình 3D.

Xây dựng cấu trúc dữ liệu để lưu trữ các phần tử và biểu diễn các bề mặt lưới 3D.

Ứng dụng giải thuật Delaunay xây dựng lưới tam giác từ tập hợp điểm 3D đầu vào để tạo bề mặt lưới tam giác nhằm tái tạo mô hình.

Ứng dụng phương pháp Loop chia nhỏ bề mặt lưới nhằm cải tiến, xây dựng bề mặt mịn, tron.

4. Ý nghĩa khoa học thực tiễn

Mô hình hóa đối tượng 3D, xử lý trên đối tượng 3D và hiển thị các thông số hình học thuộc tính của đối tượng.

Khả năng tái tạo vật thể từ tập hợp điểm rời rạc 3D thành mô hình đối tượng lưới 3D.

Thiết kế và hiệu chỉnh mô hình lưới 3D, kết xuất dữ liệu sang định dạng của phần mềm CAD/CAM chuyên dụng.

Phục vụ cho công tác nghiên cứu thiết kế mô hình đối tượng tham số 3D trong các phòng thí nghiệm, công ty.

5. Bố cục luận văn

Nội dung luận văn bao gồm các chương sau:

Chương 1: Tổng quan về cơ sở đồ họa 3D

Chương 2: Xây dựng lưới từ tập hợp điểm 3D và phương pháp chia nhỏ bề mặt lưới Loop

Chương 3: Phân tích xây dựng chương trình xây dựng bề mặt lưới 3D

Chương 1 TỔNG QUAN VỀ CƠ SỞ ĐỒ HỌA

Trong kỹ thuật biểu diễn mô hình, ta phân thành hai nhóm biểu diễn mô hình sau: mô hình hóa hình học và mô hình hóa vật thể.

Kỹ thuật mô hình hóa hình học được phát triển trong các ngành công nghiệp tự động hóa và chủ yếu được sử dụng để thiết kế các hình dạng xe hơi. Hiện nay, mô hình này còn được ứng dụng trong các ngành như công nghiệp hàng không, hải quân... và một số lĩnh vực khác. Mô hình này cũng hỗ trợ chính cho việc điều khiển về mặt hình dạng.

Kỹ thuật mô hình hóa vật thể được xây dựng dựa trên các thông tin biểu diễn đầy đủ, chính xác, rõ ràng một đối tượng trong không gian chúng có thể tạo ra các mô hình trên máy tính với khả

năng phân loại bất kỳ điểm nào trong không gian ba chiều: phía trong, phía ngoài, hoặc là trên bề mặt của đối tượng.

1.1 Hệ tọa độ trong không gian

Hệ tọa độ trong không gian là dựa vào 3 trục vuông góc nhau từng đôi một $x'Ox$, $y'Oy$, $z'Oz$ mà trên đó đã chọn 3 véc-tơ đơn vị i, j, k sao cho độ dài của 3 véc-tơ này bằng nhau [1], [2].

1.2 Phép biến đổi trong mô hình 3D

Các điểm trong không gian ba chiều được biểu diễn bằng hệ trục tọa độ ba chiều, có thể xem là mở rộng của hệ trục tọa độ hai chiều. Trong thế giới hai chiều, mặt phẳng XY chứa toàn bộ đối tượng. Trong thế giới ba chiều, một trục Z vuông góc được đưa ra để tạo thêm hai mặt phẳng chính khác là YZ và ZX [1], [2].

1.3 Biểu diễn đối tượng 3D

Các đối tượng trong thế giới thực phần lớn là các đối tượng ba chiều còn các thiết bị hiển thị chỉ biểu diễn hai chiều. Do vậy muốn có hình ảnh ba chiều ta cần phải tiến hành giả lập. Ngoài ra khi chúng ta mô hình hoá và hiển thị một đối tượng ba chiều, ta cần phải xem xét rất nhiều khía cạnh khác nhau như mặt phẳng, mặt cong và một số thông tin về bên trong các đối tượng.

1.3.1 Biểu diễn mô hình khung dây

Biểu diễn mô hình ta có thể biểu diễn dưới dạng mô hình khung dây, mô hình lưới đa giác, ...

Biểu diễn mô hình dạng khung dây cho ta thấy được các chi tiết bên trong của mô hình, sử dụng các phương pháp di chuyển, xoay, xoá đi các đường khuất (để thể hiện mô hình dạng mặt).

1.3.2 Biểu diễn đường, mặt cong

Các đường cong và mặt cong có thể được tạo ra từ một tập các hàm toán học định nghĩa các đối tượng hoặc tập các điểm trên đối

tượng. Đối với các đối tượng hình học như hình tròn hay elip thì thư viện đồ họa đã cung cấp sẵn hàm vẽ đối tượng lên mặt phẳng hiển thị. Hình biểu diễn đường cong là tập các điểm dọc theo hình chiếu của đường mô tả bởi hàm số. Nhưng với các đường cong hay mặt cong không có quy luật, thì tập điểm hay lưới đa giác xấp xỉ với đường mặt cong sẽ tạo ra. Hệ đồ họa hay tạo các lưới tam giác để đảm bảo tính đồng phẳng của các cạnh [2].

1.4 Thư viện xử lý đồ họa OpenGL

OpenGL là một tiêu chuẩn kỹ thuật đồ họa nhằm mục đích tạo ra một giao diện lập trình ứng dụng đồ họa 3D được phát triển đầu tiên bởi Silicon Graphic, Inc. OpenGL đã trở thành một chuẩn công nghiệp và các đặc tính kỹ thuật của OpenGL do Ủy ban kỹ thuật ARB. OpenGL cho phép phát triển các ứng dụng đồ họa sử dụng nhiều ngôn ngữ lập trình khác nhau như C/C++, Java, Delphi, v.v..., tuy nhiên OpenGL cũng có thể được dùng trong ứng dụng đồ họa 2D. Giao diện lập trình này chứa khoảng 250 hàm để vẽ các cảnh phức tạp từ những hàm đơn giản và được ứng dụng rộng rãi trong các trò chơi điện tử. Ngoài ra còn được dùng trong các ứng dụng CAD, thực tại ảo, mô phỏng khoa học, mô phỏng thông tin, phát triển trò chơi. OpenGL sử dụng hệ tọa độ theo quy tắc bàn tay phải.

Chương 2 PHƯƠNG PHÁP TẠO LƯỚI 3D VÀ CHIA NHỎ BỀ MẶT LƯỚI LOOP

Mô hình hoá các đối tượng 3D trên máy tính bằng cách số hoá các đối tượng thực đã và đang được đầu tư rộng rãi trên thế giới. Tại Việt Nam, công nghệ CNC đã được biết đến và ứng dụng rộng rãi vào nhiều ngành công nghiệp khác nhau.

Trong công nghệ CNC các đối tượng thực được tạo ra từ các hình khối 3 chiều mẫu có trong máy tính. Để tạo ra các mô hình ba chiều mẫu trên máy tính, nhiều kỹ thuật khác nhau được áp dụng và được gọi là công nghệ đảo ngược. Trên thế giới công nghệ đảo ngược được ứng dụng rộng rãi vào các lĩnh vực mới như CAD/CAM, thực tại ảo, kiến trúc, bảo tồn các di sản văn hoá, v.v... Công nghệ đảo ngược thu thập dữ liệu của đối tượng thực dưới dạng các điểm, từ các dữ liệu điểm thu được một lưới đa giác được xây dựng để liên kết các điểm ba chiều trong không gian. Mô hình đối tượng thực được số hoá hoàn toàn sau khi thực hiện một số bước tinh chỉnh như làm mịn (bóng), vá lỗ thủng, hoặc chia nhỏ lưới đa giác để tăng cường chi tiết hoá mô hình số. Xuất phát từ các vấn đề trên, trong luận văn này tôi triển khai nghiên cứu các phương pháp biểu diễn bề mặt lưới từ các tập hợp điểm đo được nhằm tìm kiếm một giải pháp tốt nhất để triển khai mô hình hoá các đối tượng 3D từ tập hợp điểm đã cho.

2.1 Biểu diễn bề mặt lưới 3D

Để lưu trữ cơ sở dữ liệu số trong không gian nhằm tham chiếu chính để tạo mô hình. Hệ thống thông tin địa lý GIS là một cơ sở dữ liệu số chuyên dụng trong hệ trục tọa độ không gian. Như vậy GIS có quan hệ với các ứng dụng cơ sở dữ liệu. Toàn bộ thông tin trong GIS đều liên kết với tham chiếu không gian. Cách thức nhập, lưu trữ, phân tích dữ liệu trong GIS phải phản ánh đúng cách thức thông tin sẽ được sử dụng trong việc lập quyết định hay nghiên cứu cụ thể.

2.2 Xây dựng bề mặt lưới

Phương pháp xây dựng lưới tam giác từ một tập hợp điểm là phương pháp dựng hình trong không gian 2-chiều và 3-chiều. Các quá trình tam giác hoá sẽ được thực hiện với dữ liệu đầu vào là tập

hợp hỗn độn các điểm đã thu thập được. Thuật toán tam giác hoá có thể trình bày trong [4], [10], [11]:

2.3 Các lược đồ xây dựng bề mặt lưới 3D

Cho V là tập hữu hạn các đỉnh trên mặt phẳng R^2 [3]. Cho E là tập các cạnh mà các điểm đầu cuối là các đỉnh thuộc tập V . Ta có các định nghĩa sau:

Định nghĩa 1. Lưới tam giác $T = (V, E)$ là một đồ thị phẳng mà mỗi cạnh không chứa đỉnh nào khác ngoài hai đỉnh đầu cuối của nó, không có hai cạnh nào cắt nhau và tất cả các mặt là những tam giác với hội của chúng là bao lồi của tập đỉnh V .

Định nghĩa 2. Bài toán nối các điểm cho trước trên mặt phẳng bằng các đoạn thẳng không cắt nhau để tạo thành lưới tam giác gọi là bài toán xây dựng lưới tam giác. Về mặt bản chất, bài toán xây dựng lưới tam giác từ tập điểm cho trước là không duy nhất. Một trong các điều kiện xây dựng lưới tam giác hay được sử dụng trong các ứng dụng là điều kiện Delaunay.

Định nghĩa 3. Ta nói rằng lưới tam giác thỏa điều kiện Delaunay nếu bên trong đường tròn ngoại tiếp của mỗi tam giác không chứa bất kỳ điểm nào thuộc lưới tam giác đó.

2.3.1 Sơ đồ Voronoi

Sơ đồ Voronoi biểu diễn như sau: mỗi vùng Voronoi $V(p_i)$ là đa giác lồi, với $V(p_i)$ là không đóng kín nếu p_i thuộc bao lồi của tập điểm. Nếu v là đỉnh của Voronoi ở điểm giao nhau của $V(p_1)$, $V(p_2)$ và $V(p_3)$ thì v là tâm của đường tròn $C(v)$ xác định bởi p_1 , p_2 , p_3 . Trong đó $C(v)$ là đường tròn ngoại tiếp tam giác Delaunay tương ứng với v . Bên trong $C(v)$ không chứa điểm p_j . Nếu p_j là điểm gần nhất đến p_j thì (p_j, p_j) là cạnh tam giác Delaunay. Bất kỳ một đường

tròn nào đi qua hai điểm p_j, p_j mà không chứa bất kỳ điểm nào thì (p_j, p_j) là cạnh tam giác Delaunay [7], [8], [9].

2.3.2 Sơ đồ Delaunay

Với $D(P)$ là các đường thẳng đối ngẫu của $V(P)$. Mỗi tam giác của $D(P)$ tương ứng đến đỉnh của $V(P)$. Mỗi cạnh của $D(P)$ tương ứng đến cạnh của $V(P)$. Mỗi nút của $D(P)$ tương ứng đến vùng của $V(P)$. Đường bao của $D(P)$ là bao lồi của P . Bên trong tam giác của $D(P)$ là bao lồi của P .

Phép chia nhỏ tạo ra số tam giác lớn nhất không tồn tại một cạnh nào nối hai điểm có thể thêm vào mà không phá vỡ phép chia nhỏ S . Tam giác của tập P ($n \geq 2$ điểm) là Delaunay nếu và chỉ nếu đường tròn qua nó không chứa điểm thứ tư.

2.4 Thuật toán Delaunay

Thuật toán tạo lưới tam giác Delaunay là một bài toán cơ bản trong hình học tính toán và nó được sử dụng trong rất nhiều lĩnh vực như hệ thống thông tin địa GIS, phân tử hữu hạn, đồ họa máy tính và đa phương tiện...

Để xây dựng lưới tam giác Delaunay từ tập hợp điểm đầu vào ta có thể chia thành các hướng tiếp cận sau:

a) Hướng tiếp cận chia để trị

Đầu tiên, tập điểm đầu vào được chia thành các tập con, thực hiện xây dựng lưới tam giác cho mỗi tập con rồi hợp nhất lại để tạo ra lưới tam giác kết quả cuối cùng. Tuy nhiên, phân hợp nhất các kết quả con thường cài đặt khá phức tạp. Giải pháp này được đề cập bởi Dwyer trong công trình [6].

b) Hướng tiếp cận sử dụng dòng quét

Giải thuật theo hướng tiếp cận này đã được đưa ra bởi Fortune [7], sau đó được tổng hợp lại bởi de Berg trong công trình [5]. Fortune đã phát triển giải thuật xây dựng đồ thị đối ngẫu giữa lưới tam giác Delaunay và sơ đồ Voronoi. Việc cài đặt giải thuật là khá phức tạp.

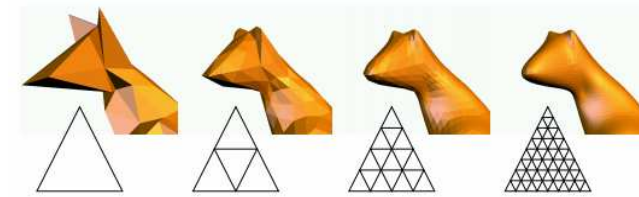
c) Hướng tiếp cận thêm từng điểm tuần tự

Các giải thuật thuộc hướng tiếp cận này khá đơn giản để cài đặt. Giả sử chúng ta có lưới tam giác Delaunay được xây dựng từ $i-1$ điểm. Điểm thứ i sẽ được thêm vào lưới tam giác theo cách sau: Xác định tam giác chứa điểm i mới thêm vào lưới. Phân rã tam giác đó thành các tam giác con thuộc lưới và hiệu chỉnh thoả điều kiện Delaunay [4].

2.5 Phương pháp chia nhỏ bề mặt lưới tam giác Loop

Các mô hình 3D được tạo ra từ các tập hợp điểm bằng giải thuật Delaunay sẽ tạo ra các bề mặt lưới tam giác thô. Để tạo ra được các bề mặt mịn hơn ta có thể áp dụng các thuật toán chia nhỏ bề mặt lưới.

Phương pháp Loop chia nhỏ bề mặt lưới là dựa vào các lưới tam giác đầu vào. Với bề mặt lưới đầu vào là các bề mặt tam giác được tạo ra từ phương pháp tạo lưới thông qua giải thuật Delaunay. Trong phạm vi của luận văn này, tôi đã tập trung vào phương pháp Loop chia nhỏ bề mặt lưới, bằng các phép tính xấp xỉ, nhằm tạo ra các bề mặt có tính liên tục. Với thuật toán Loop, chia nhỏ mỗi mặt trong lưới đầu vào được chia thành bốn mặt nhỏ hơn.



Hình 2.16 Biểu diễn giải thuật chia nhỏ Loop

Mô hình chia nhỏ theo phương pháp Loop tiến hành chia nhỏ mặt lưới tam giác bằng cách thực hiện một số bước lặp. Chú ý rằng các đa giác tạo nên các bề mặt phải là các tam giác, các bề mặt không thể là những đa giác phẳng bất kỳ.

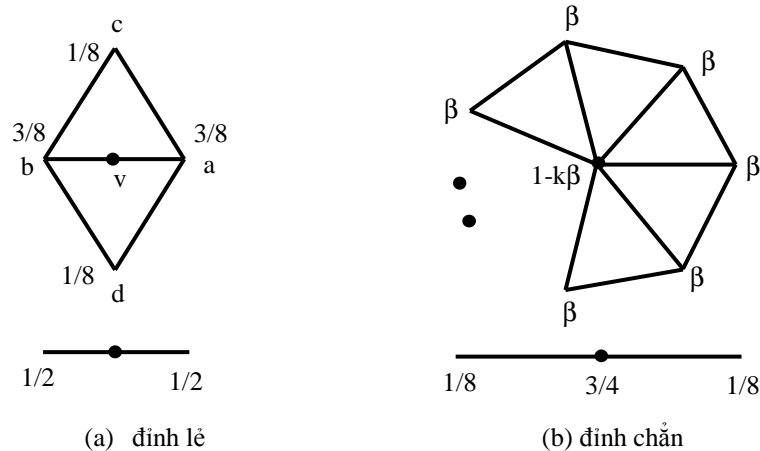
Giải thuật này do Charles Teorell Loop đề cập trong [12], ông đã cải tiến giải thuật chia nhỏ mỗi tam giác thành 4 tam giác con. Giải thuật này cũng được biết đến với tên là giải thuật chia nhỏ Loop nhị phân.

Giải thuật Loop nhị phân bắt đầu bằng tập điểm là các đỉnh của tam giác. Mỗi phép lặp tính một tập các cạnh và các điểm đỉnh mới tạo nên các đỉnh mới của các tam giác nhỏ hơn. Đặc biệt, đối với mỗi cạnh và một điểm đỉnh mới thì một điểm cạnh mới được tính cho mỗi đỉnh của lưới tam giác.

Giải thuật chia nhỏ Loop được chia làm hai bước. Bước đầu tiên tạo ra tất cả các điểm và tam giác mới, và bước thứ hai sẽ định vị lại tất cả các điểm cũ. Giải thuật Loop là giải thuật tính xấp xỉ nghĩa là các đỉnh sẽ được hiệu chỉnh lại trong suốt quá trình phân chia.

Các đỉnh lẻ là các đỉnh được thêm vào trong suốt quá trình phân chia trong khi đó các đỉnh chẵn là các đỉnh sẵn có trước lúc chia. Các cạnh biên là cạnh nằm trên đường biên của lưới. Nghĩa là, nếu tồn tại một cạnh ab của tam giác mà không có chung với cạnh tam giác nào khác thì nó chính là cạnh biên. Đỉnh a, b có thể mô tả

như là các đỉnh biên. Các cạnh bên trong (không phải cạnh biên) là tập bổ sung vào bất cứ cạnh nào có chung 2 tam giác. Một đỉnh bên trong chỉ nằm trên các cạnh bên trong.

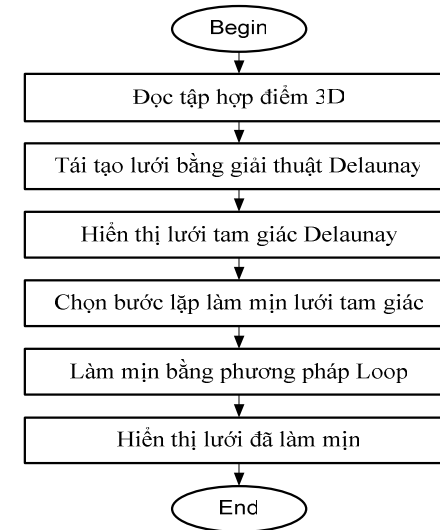


Hình 2.18 Hình minh họa đỉnh lẻ và đỉnh chẵn

Chương 3 PHÂN TÍCH XÂY DỰNG CHƯƠNG TRÌNH XÂY DỰNG BỀ MẶT LƯỚI 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 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 3 chiều trong không gian sử dụng thuật toán Delaunay. Mô hình đối tượng được số hoá, sau khi thực hiện một số bước tinh chỉnh, làm mịn, vá lỗ thủng hoặc chia nhỏ lưới tam giác để tăng cường chi tiết hoá mô hình số bằng giải thuật chia nhỏ Loop.

Các bước xây dựng chương trình tái tạo bề mặt lưới từ tập hợp điểm 3D và phương pháp chia nhỏ bề mặt lưới được minh họa như sau:



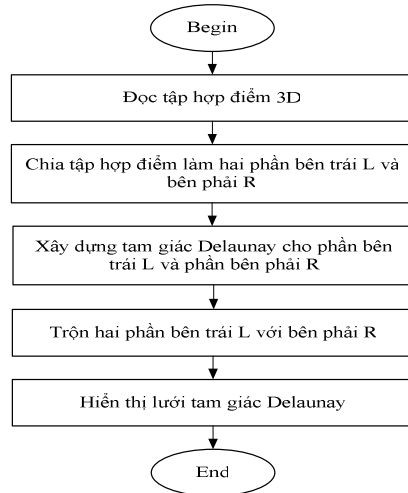
Hình 3.1 Sơ đồ tạo lưới tam giác Delaunay và làm mịn

3.1 Tổ chức dữ liệu bề mặt lưới

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 các 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. Để lưu trữ tập hợp điểm 3D thu được và mô hình hóa hình học ta sử dụng ngôn ngữ VRML (*Virtual Reality Modeling Language*).

3.2 Xây dựng lưới tam giác Delaunay từ tập hợp điểm rời rạc 3D

3.2.1 Sơ đồ khối của thuật toán được mô tả như sau:



Hình 3.2 Sơ đồ xử lý tập hợp điểm tạo lưới tam giác Delaunay

3.2.2 Cấu trúc dữ liệu

3.2.2.1 Phân tích cấu trúc dữ liệu

Phép chia nhỏ S cắt miền M chứa các điểm $P := \{p_1, p_2, \dots, p_3\}$ thành ba tập hợp đỉnh, cạnh, mặt phẳng liên kết với nhau có các thuộc tính sau: tập đỉnh là các điểm p_i thuộc M, tập cạnh là các đoạn thẳng nối các đỉnh thuộc M, tập mặt phẳng chứa các tam giác đều thuộc M, đường biên của mặt phẳng qua các cạnh và đỉnh

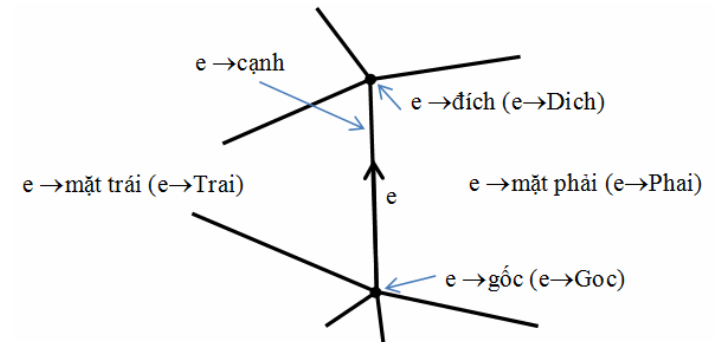
3.2.2.2 Cạnh góc (QuadEdge)

Chúng ta biểu diễn phép chia nhỏ S bởi cấu trúc dữ liệu cạnh góc (quad-edge). Mỗi mẫu tin về cạnh (edge record) chứa bốn cạnh $e[0], e[1], e[2], e[3]$ mỗi cạnh $e[r]$ với $r=0,1,2,3$ chứa hai trường Data và Next. Trường Data lưu trữ thông tin hình học, trường Next chứa tham chiếu cạnh tiếp theo.

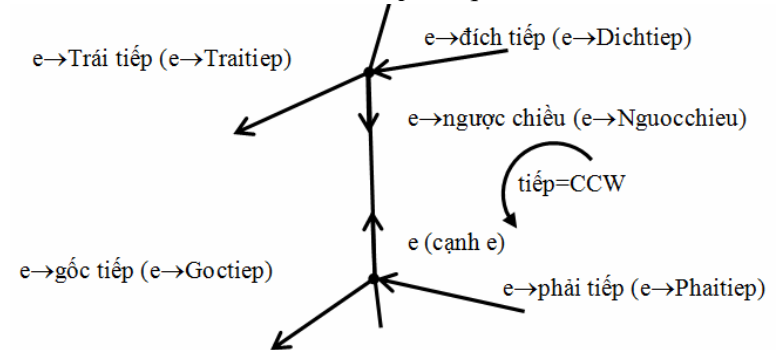
Với mỗi cạnh của phép chia nhỏ luôn có hướng ứng với chiều các cạnh. Mỗi cạnh ký hiệu là e chúng ta có thể định nghĩa như

sau: với đỉnh gốc được ký hiệu là $e \rightarrow \text{Goc}$, đỉnh đích ký hiệu là $e \rightarrow \text{Dich}$, mặt phẳng bên trái ký hiệu $e \rightarrow \text{Trai}$, mặt phẳng bên phải ký hiệu là $e \rightarrow \text{Phai}$, cạnh ngược chiều ký hiệu là $e \rightarrow \text{NguocChieu}$, cạnh tiếp theo có cùng gốc ký hiệu là $e \rightarrow \text{Goctiep}$, cạnh tiếp theo ngược chiều kim đồng hồ thuộc mặt phẳng bên trái là ký hiệu là $e \rightarrow \text{Traitiep}$, cùng chiều kim đồng hồ ký hiệu là CCW và ngược chiều kim đồng hồ ký hiệu là CW.

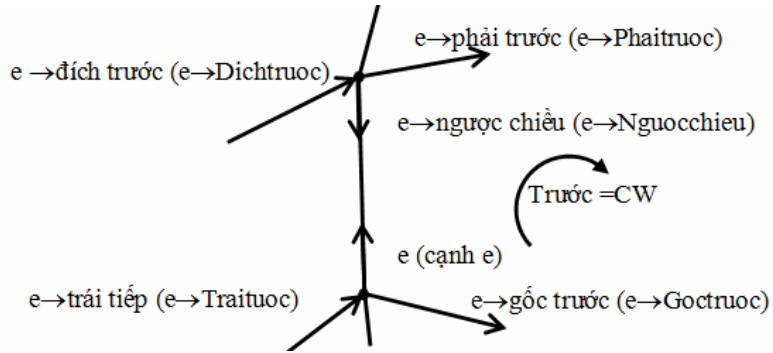
Với một cạnh bất kỳ e (ký hiệu cạnh là e) thì ta dễ tìm thấy các đỉnh lân cận, các cạnh, các mặt và cạnh đối xứng theo hướng ngược lại.



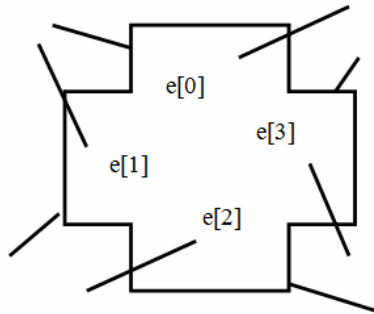
Hình 3.3 Các thành phần quanh cạnh e



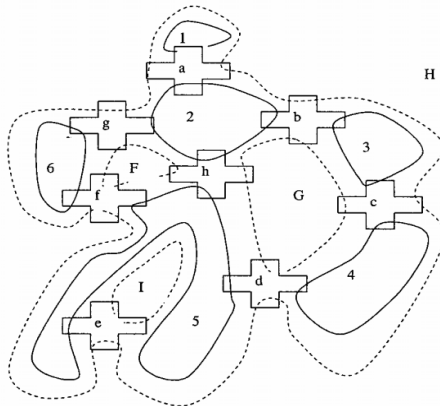
Hình 3.4 Biểu diễn hướng theo chiều ngược chiều kim đồng hồ



Hình 3.5 Biểu diễn theo hướng cùng chiều kim đồng hồ



Hình 3.6 Cấu trúc cạnh góc (quad-edge)



Hình 3.7 Cấu trúc quad-edge của đồ thị (vòng đậm là đỉnh, vòng nét đứt là mặt phẳng)

3.2.3 Các toán tử thao tác trên cạnh góc (Quad-edge)

3.2.3.1 Toán tử tạo cạnh $e \rightarrow MakEdge()$

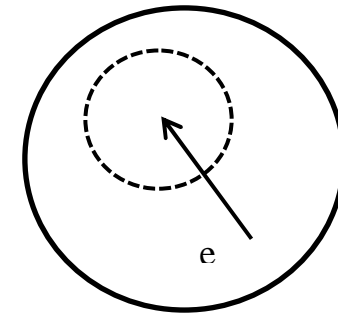
Trả về cạnh e khi khởi tạo cấu trúc dữ liệu. Do đó, có một cạnh duy nhất của phép chia nhỏ nên không có vòng lặp:

$eGoc \# eDich$ //góc của cạnh e khác với đích của cạnh e

$eTrai = ePhai$ //cạnh bên trái của e bằng với cạnh bên phải của e

$eTraitiep = ePhaitiep = eNguocchieu$
// cạnh bên trái tiếp của e bằng với cạnh bên phải tiếp của e bằng cạnh ngược chiều của e

$eGoctiep = eGoctruoc = eNguocchieu$
// gốc bên trái tiếp của e bằng với gốc bên phải trước của e bằng với cạnh e ngược chiều



Hình 3.8 Khởi tạo cạnh e

3.2.3.2 Toán tử nối cạnh $Splice(a,b)$

Toán tử này nối vòng $aGoc$ và $bGoc$ không phụ thuộc hai cạnh vòng $aTrai$ và $bTrai$. Trong trường hợp

Nếu hai vòng là riêng biệt $Splice$ kết hợp chúng thành một.

Nếu cả hai cùng một vòng, Splice bẻ nó thành hai phần.

Nếu cả hai cùng một vòng nhưng có chiều đối ngược, Splice đảo ngược thứ tự của đoạn trong vòng.

3.2.3.3 *Toán tử ghép cạnh Connect(a,b)*

Thực hiện tạo cạnh e mới đi từ đỉnh của a đến góc của b do đó aTrai=eTrai=bTrai. Được định nghĩa qua toán tử Splice như sau:

```
connect(a,b)
{
    e=MakEdge(); e.Goc=a.Dich;
    e.Dich=b.Goc;
    splice(e,a.Traitiep);
    splice(e.Nguocchieu,b);
    return e; }
```

3.2.3.4 *Toán tử loại bỏ cạnh Delete(e)*

Thực hiện loại bỏ cạnh e ra khỏi cấu trúc của quad-edge.

Ngược với toán tử ghép cạnh Connect.

```
delete(e)
{
    splice(e,e.Goctruoc);
    splice(e.Nguocchieu,e.Nguocchieu.Go
ctruoc); }
```

3.2.3.5 *Toán tử hoán vị cạnh Swap(e)*

```
swap(e)
{
    a=e.Goctruoc;
    b=e.Nguocchieu.Goctruoc;
    splice(e,a);
    splice(e.Nguocchieu,b);
    splice(e,a.Traitiep);
    splice(e.Nguocchieu,b.Traitiep);
    e.Goc=a.Dich;
```

```
e.Dich=b.Dich; }
```

3.2.4 **Xây dựng lưới tam giác Delaunay**

Xây dựng sơ đồ Voronoi và các tam giác Delaunay từ cấu trúc dữ liệu quad-edge theo phương pháp chia để trị “divide and conquer” của Guibas và Stofi với bậc $n \log n$ có cải tiến tối ưu hoá cấp phát bộ nhớ, thủ tục trộn và kiểm tra bốn điểm trong đường tròn [6].

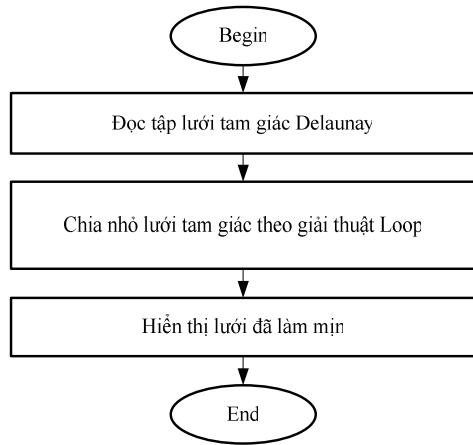
Nội dung chính là phương pháp chia để trị ta chia nhỏ tập điểm làm hai phần bên trái L và bên phải R, xây dựng tam giác Delaunay theo điều kiện kiểm tra Incircle(A,B,C,D) tuần tự cho các điểm ở phần bên trái L (ký hiệu L) và bên phải R (ký hiệu là R) theo thứ tự tăng của y. Sau đó trộn hai phần có kết quả.

3.3 **Chia nhỏ bề mặt lưới Loop**

Chia nhỏ là một kỹ thuật mà nhằm tăng chất lượng của mặt lưới bằng cách tạo ra thêm nhiều đỉnh. Sự chia bề mặt là một quá trình lặp đi lặp lại. Quá trình này bắt đầu với một mặt lưới đa giác cho trước và sau mỗi bước lặp một mô hình cụ thể được áp dụng vào mặt lưới. Nhiều đỉnh mới và các mặt cơ sở được tạo dựa trên các cạnh xung quanh nó.

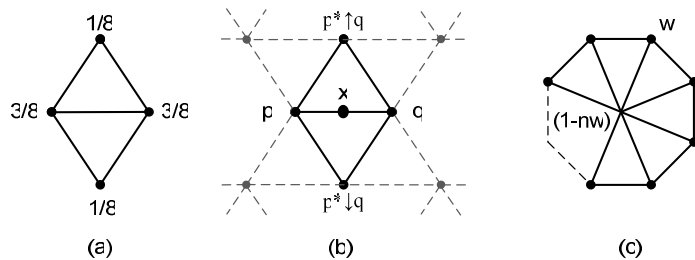
Trong luận văn này, đơn giản tôi thực hiện thuật toán chia nhỏ Loop. Vẫn còn nhiều thuật toán chia nhỏ tốt hơn nhưng đối với những người mới bắt đầu trong việc phân chia mặt lưới tam giác với thuật toán Loop là sự khởi đầu phù hợp.

Khởi đầu cho thuật toán Loop là đọc lưới tam giác Delaunay, bước kế tiếp là áp dụng thuật toán chia nhỏ theo phương pháp Loop, quá trình chia nhỏ tạo ra một lưới tam giác mịn, tô bóng lưới tam giác và xuất ra màn hình theo sơ đồ khối của thuật toán Loop như sau:



Hình 3.26 Sơ đồ xử lý lưới tam giác tạo ra lưới tam giác mịn

Thuật toán chia nhỏ Loop cũng tương tự như thuật toán chia nhỏ đa giác, vì cả hai đều thực hiện trên các lưới tam giác và chia nhỏ các mặt tam giác thành 4 tam giác trong mỗi bước. Trong trường hợp của Loop thì mục tiêu là xây dựng một bề mặt mịn với một hình dạng tổng quát được điều khiển bằng đa giác gốc. Khi kết thúc, giải thuật loop đặt các đỉnh mới sử dụng mặt nạ gồm có 4 đỉnh cũ và đặt lại đỉnh cũ bằng cách sử dụng mặt nạ khác sáp nhập tất cả đỉnh trung gian



Hình 3.27 Hình minh họa cho cài đặt thuật toán Loop

Thuật toán chia nhỏ Loop chỉ có thể áp dụng cho mặt lưới tam giác nhưng đó không phải là vấn đề vì mặt lưới các đa giác liên kề có thể chuyển thành dạng mặt lưới tam giác bằng cách phân chia bề mặt gồm 2 bước:

Bước 1: tạo tất cả các điểm và tam giác mới

Bước 2: tinh chỉnh tất cả các điểm cũ không được tạo trong bước 1. Đây là thuật toán gần đúng. Có nghĩa là có sự điều chỉnh các đỉnh đang tồn tại trong suốt quá trình phân chia.

Trong chương trình đã xây dựng các lớp mới LoopSubdivisionEdge và LoopSubdivision. Lớp LoopSubdivisionEdge lưu trữ các thông tin về cạnh bao gồm các điểm kết thúc, các tam giác liên kề và các đỉnh mới trên cạnh này để giúp kiểm tra xem có phải là cạnh bao quanh hay không và để có được mối quan hệ liên kề một cách nhanh chóng trong thuật toán phân chia Loop. Lớp LoopSubdivision đảm trách toàn bộ công việc thực hiện của thuật toán Loop.

Quy trình tính toán :

Gọi hàm `build_edge_set(mesh, edge_set, adjacent_edge_set)` để tìm tất cả các cạnh và tạo ra danh sách các cạnh liên kề cho mỗi đỉnh.

Theo số lượng các cạnh và hình tam giác, chúng ta có thể biết số lượng đỉnh và hình tam giác trong lưới mới. Cấp phát bộ nhớ cho lưới mới.

Bước 1: tạo tất cả các điểm mới. LoopSubdivisionEdge có thể giúp kiểm tra có đúng là các cạnh bao quanh hay không một cách nhanh chóng.

Bước 2: tinh chỉnh tất cả các điểm cũ. Sử dụng các đỉnh của danh sách cạnh liền kề để kiểm tra đỉnh ranh giới và tìm các đỉnh ranh giới liền kề.

Kết nối mỗi đỉnh và tạo lưới mới.

Thay lưới cũ bằng các đỉnh mới và các tam giác mới;

Khi quá trình phân chia sẽ làm thay đổi kích cỡ mặt lưới, chương trình phải cấp phát bộ nhớ mới cho mỗi hoạt động phân chia. Và chương trình cũng nên xoá bộ nhớ cũ sau khi lưới mới được tạo ra.

3.4 Kết quả thực hiện chương trình

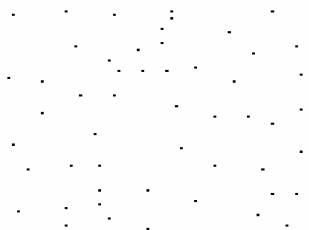
Chương trình được xây dựng trên môi trường Visual Studio C++ 6.0, kết hợp thư viện đồ họa OpenGL.

Với dữ liệu đầu vào khác nhau nên trong luận văn này tôi xây dựng hai chương trình để đọc dữ liệu khác nhau.

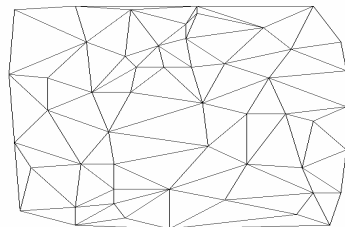
Sau khi chạy chương trình thì ta có các kết quả thực nghiệm như sau:

Với tập hợp điểm lưới đầu vào, được lưu trữ trong file có phần mở rộng là Vertex.nod. Chương trình đọc file và cho hiển thị dưới dạng điểm ta thu được như hình.

Thực hiện thuật toán Delaunay với tập hợp điểm đầu vào ta thu được kết quả hình sau:



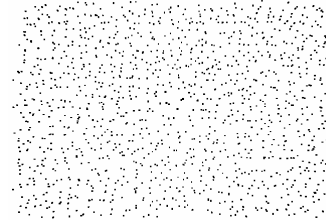
Hình 3.28 Tập hợp điểm 2D



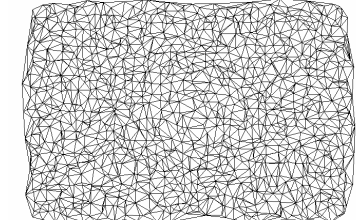
Hình 3.29 Sau khi thực hiện thuật

toán Delaunay với tập điểm 2D

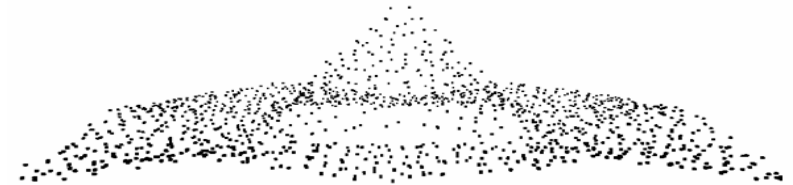
Với tập hợp điểm lưới đầu vào, được lưu trữ trong file có phần mở rộng là Vertex.tri. Chương trình đọc file và cho hiển thị dưới dạng điểm ta thu được như hình.



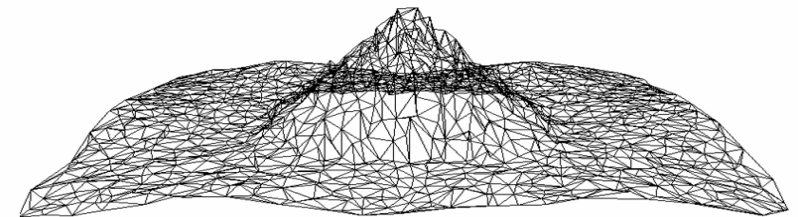
Hình 3.30 Tập điểm 3D đầu vào nhìn theo hướng hình chiếu bằng



Hình 3.31 Lưới Delaunay nhìn theo hướng hình chiếu bằng

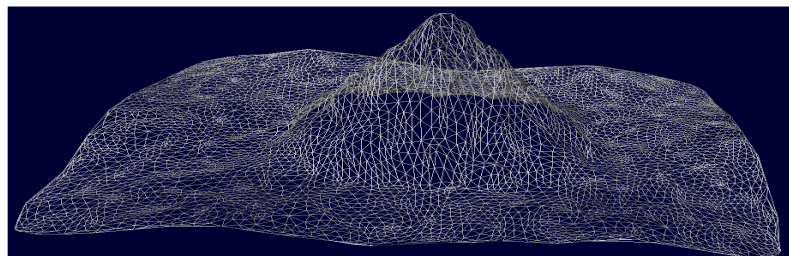


Hình 3.32 Tập điểm 3D đầu vào theo hướng hình chiếu đứng

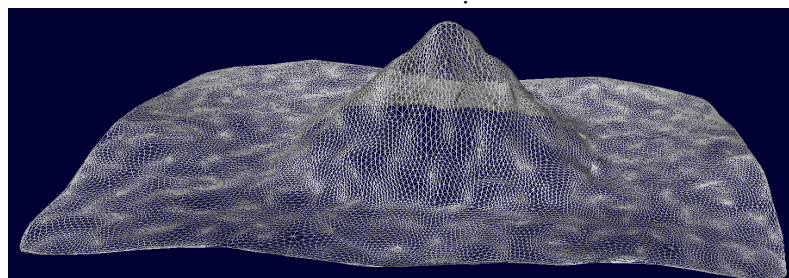


Hình 3.33 Lưới Delaunay nhìn theo hướng hình chiếu đứng

Thực hiện chạy chương trình chia nhỏ Loop với file đầu vào chứa lưới tam giác Delaunay ta có kết quả sau:



Hình 3.34 Sau khi chia nhỏ mặt lưới lần thứ nhất



Hình 3.35 Sau khi chia nhỏ mặt lưới lần thứ 2

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Ngày nay đồ họa máy tính ngày càng phát triển nhanh chóng thì việc nghiên cứu và ứng dụng vào từng lĩnh vực thực tế là một xu hướng tất yếu. Trong quá trình tìm hiểu, nghiên cứu tái tạo bề mặt lưới từ tập hợp điểm đo được trong không gian và chia nhỏ bề mặt lưới vừa tạo ra từ tập hợp điểm nhằm làm mặt lưới đạt được kết quả tốt hơn, luận văn đã đạt được kết quả khả quan. Kết quả đạt được của luận văn về mặt lý thuyết đã đi sâu vào tìm hiểu về cơ chế đồ họa trên máy tính, tìm hiểu các qui trình về xử lý, hiển thị đối tượng, mô hình hoá các vật thể trong môi trường đồ họa 3 chiều. Nghiên cứu cách xây dựng mặt lưới tam giác từ tập hợp điểm 3 chiều rời rạc trong không gian, trong quá trình nghiên cứu, tôi nhận thấy với thuật toán Delaunay ta có thể dễ dàng xây dựng mặt lưới tam giác từ một tập hợp điểm.

Kết quả thực nghiệm xây dựng được mặt lưới tam giác bằng phương pháp Delaunay.

Các tập hợp điểm thu thập được thông qua các hình mẫu (vật thể, địa hình,...) là tập hợp các điểm 2 chiều và 3 chiều, những tập hợp điểm này không có tổ chức. Từ tập hợp điểm, sử dụng phương pháp Delaunay tam giác hoá. Việc xây dựng được mặt lưới tam giác từ các tập điểm 2 chiều và 3 chiều chỉ là mặt lưới thô nên cần thực hiện một số thao tác tinh chế và làm mịn. Sau một quá trình xử lý thu được một lưới tam giác đặc tả về hình mẫu. Để thực hiện điều này luận văn đã đề xuất giải pháp chia nhỏ mặt lưới tam giác bằng cách sử dụng giải thuật Loop.

Để chia nhỏ mặt lưới, trong luận văn tiến hành tìm hiểu và nghiên cứu lý thuyết về lĩnh vực này. Có rất nhiều thuật toán áp dụng để chia nhỏ lưới để tối ưu bề mặt lưới. Nhưng do đề tài nghiên cứu chỉ sử dụng lưới tam giác Delaunay nên tôi tiến hành nghiên cứu về giải thuật Loop chia nhỏ bề mặt lưới. Kết quả đã xây dựng được một chương trình chia nhỏ mặt lưới tam giác, làm mịn bề mặt lưới.

Tuy nhiên trong luận văn này chỉ dừng lại ở mức xây dựng từ tập hợp điểm đã có, chưa có thể sử dụng các thiết bị đặc dụng để quét mô hình vật thể trong môi trường không gian thực. Nên luận văn chỉ dừng lại ở mức mô hình hoá vật thể.

Hiện nay công nghệ tái tạo ngược đã và đang phát triển rất mạnh trên thế giới và cả ở Việt Nam, công nghệ đã được áp dụng rộng rãi vào các lĩnh vực mới như CAD/CAM, thực tại ảo, kiến trúc, bảo tồn di sản văn hoá nhằm tái tạo lại các hình mẫu điều khác.

Hướng phát triển của đề tài là tiếp tục hoàn thiện chương trình, bổ sung thêm vào các chức năng của các phần mềm thiết kế CAD/CAM hiện có.