*Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công Nghệ thông tin (FAIR); Đà Nẵng, ngày 17-18/08/2017*
*DOI: 10.15625/vap.2017.00063*

# MỘT KỸ THUẬT RÚT GỌN BỀ MẶT LƯỚI MÔ HÌNH 3D TRONG CÁC ỨNG DỤNG CỦA THỰC TẠI ẢO

**Mã Văn Thu[1], Đỗ Năng Toàn[2], Lê Sơn Thái[1], Đỗ Hồng Quân[2]**

[1]Đại học Công nghệ thông tin và Truyền thông
[2]Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

*mvthu@ictu.edu.vn, dntoan@vnu.edu.vn, lsthai@ictu.edu.vn, quandh@vnu.edu.vn*

***TÓM TẮT**: Thực tại ảo là công nghệ sử dụng kỹ thuật mô hình hóa không gian ba chiều để mô phỏng lại thế giới thực hoặc thế giới theo tưởng tượng của con người. Các mô hình ba chiều trong thực tại ảo có thể là tĩnh hoặc động và có thể tương tác với nhau. Một vấn đề lớn ở đây là các mô hình khi tạo ra để thỏa mãn yêu cầu thẩm mỹ của con người thường có số lượng bề mặt lưới là lớn. Với các mô hình như vậy thì một số thiết bị chạy các ứng dụng thực tại ảo phải đáp ứng được về cấu hình và băng thông mạng internet. Bài báo trình bày một kỹ thuật về rút gọn bề mặt lưới mô hình ba chiều dựa trên công thức "Gaussian Curvature" kết hợp thêm một số ràng buộc để mô hình sau khi rút gọn lưới đảm bảo về hình dáng trong các ứng dụng.*

***Từ khóa:** optimize mesh, optimizing triangle, tối ưu bề mặt lưới mô hình, simplifying polygonal, decimation of triangle meshes, multiresolution modeling.*

## I. GIỚI THIỆU

Ngày nay, thực tại 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. và việc xây dựng các đối tượng 3D (ba chiều) là rất 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 thẩm mỹ của con người.

Bài toán rút gọn bề mặt lưới trên mô hình 3D có nhiều ý nghĩa khi mà số lượng mô hình cần đưa vào các ứng dụng thực tại ảo ngày một nhiều. Ví dụ chúng ta mô phỏng lại một thành phố, hay tái tạo lại một khu bảo tàng để người du lịch có thể thăm quan trong môi trường ảo. 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 yếu. Với khả năng ứng dụng cao thì cần phải có những nghiên cứu chuyên sâu về rút gọn lưới để các mô hình 3D có thể được ứng dụng rộng dãi hơn.

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.

Phương pháp sử dụng các thiết bị phần cứng là máy quét để tạo mô hình 3D mang nhiều ưu điểm như 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. 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.

Mô hình khoảng 7 triệu lưới

Mô hình khoảng 4 triệu lưới

**Hình 1.** Các mô hình được tạo ra có số lượng lưới cực lớn

Hai dạng bài toán rút gọn lưới 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 và sau rút gọn lưới.

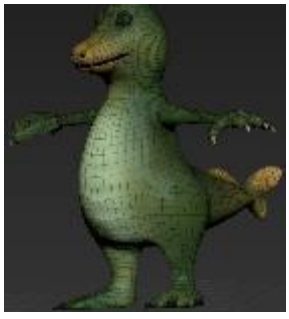
Thứ nhất, là rút gọn về mặt hình ảnh. Ở đó, với đầu vào là một mô hình 3D đã được thiết kế hoặc thu từ máy quét người xử lý cần nâng cao chất lượng hình ảnh của mô hình. Khi đó chúng ta cần chú ý tới việc tối ưu chất lượng hình ảnh hoặc lưới của mô hình, điều này dẫn tới các bài toán xử lý về ánh sáng, góc cạnh để 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 rút gọn số lượng lưới với bài toán này đầu vào là một mô hình 3D và đầu ra là mô hình đó với số lượng lưới giảm đi nhưng vẫn đảm bảo hình dạng của đối tượng không thay đổi nhiều giữa trước và sau rút gọn. Trong nội dung báo cáo tập trung vào giải quyết bài toán thứ hai.

Mắt lưới tam giác thường được sử dụng để đại diện cho các bề mặt 3D. Chúng ta quan niệm tập rời rạc các điểm, các cạnh nối đại diện cho các bề mặt là đại diện cho các dữ liệu của mô hình 3D.

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 bằng cách rút gọn lưới đa giác của mô hình.

Mô hình khoảng 8 triệu lưới



Mô hình khoảng 31 nghìn lưới

**Hình 2.** Rút gọn bề mặt lưới

## II. PHƯƠNG PHÁP RÚT GỌN LƯỚI MÔ HÌNH

### *A. Giới thiệu về các phương pháp tối ưu phổ biến*

Trong nhiều năm qua, vấn đề của việc rút gọn bề mặt lưới nhận được rất nhiều sự chú ý. Vấn đề này đã được đi sâu nghiên cứu trong suốt thế kỷ XIX 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. Các hướng tiếp cận việc rút gọn bề mặt lưới tam giác đã được nghiên cứu, có 3 hướng chính đó là:

Hướng thứ nhất, Multi resolution là hướng giải quyết việc rút gọn bề mặt lưới một cách toàn cục dựa trên các phương pháp sau: remeshing, parametric surfaces, subdivision surfaces.

Hướng thứ hai, Polygonal simplification là việc giảm số lượng đa giác của lưới trên mô hình dựa trên các giải pháp sau: theo toán tử cục bộ gồm các phương pháp phân cụm điểm, incremental decimation, tam giác rút gọn; theo toán tử toàn cục gồm các phương pháp bộ loc low-pass, alpha-hull.

Hướng thứ ba, Image imposters là dùng ảnh để làm tăng độ sần sùi hay độ phức tạp cho mô hình.

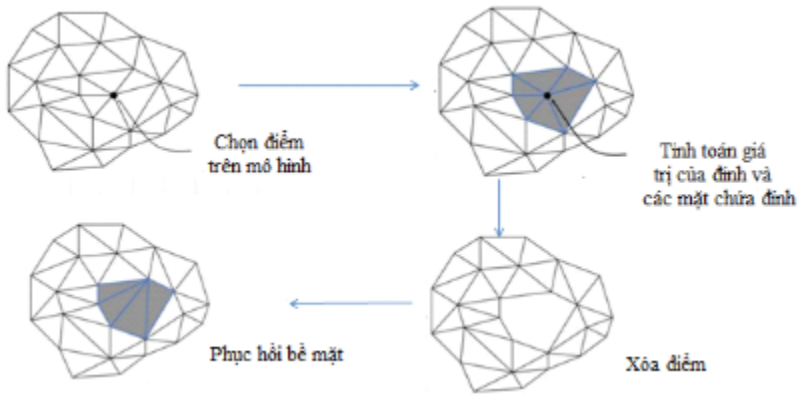
Tùy vào từng trường hợp mà người ta dựa vào các hướng giải quyết khác nhau để rút gọn bề mặt lưới được tốt nhất. Chúng tôi tiếp cận theo hướng thứ 2 là: Polygonal simplification và trong hướng này chúng ta sẽ quan tâm đặc biệt đến phương pháp Incremental decimation, vì phương pháp này được phát triển rộng rãi trong các gói chương trình đã nghiên cứu.

### *B. Phương pháp Incremental decimation*

Phương pháp incremental decimation là việc thực hiện xoá bỏ một đỉnh tại một thời điểm và phục hồi bề mặt mô hình. Ba toán tử chính để loại bỏ điểm là: vertex removal (xóa điểm), edge collapse (gộp hai điểm trên cạnh thành một điểm), half edge collapse (gộp hai điểm thành một điểm, trong đó một điểm sẽ được giữ nguyên vị trí như ban đầu). Thứ tự xóa điểm dựa trên một số hàm ưu tiên. Quy trình cho việc tối ưu hóa bề mặt lưới tam giác như sau:

```
Repeat
    {
        Chọn đối tượng 3D;
        Áp dụng toán tử decimation để giảm lưới đối tượng;
    }
Until  mục tiêu tối giản được đáp ứng.
```

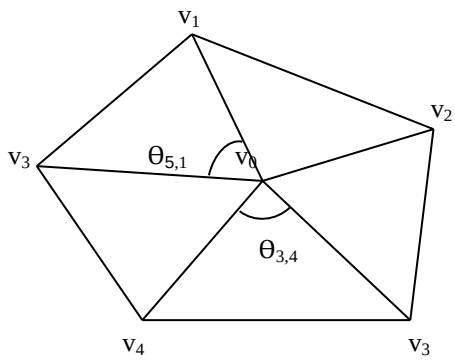
Quy trình loại bỏ đỉnh và phục hồi bề mặt áp dụng theo thứ tự ưu tiên để loại bỏ điểm ra khỏi mô hình đối tượng, sau khi đã xóa điểm xong thì cần thêm bước nữa để phục hồi bề mặt của đối tượng. Việc xử lý loại bỏ đỉnh dừng khi số lượng lưới thấp nhất có thể mà bề mặt mô hình vẫn đảm bảo về hình dáng.




**Hình 3.** Loại bỏ và phục hồi bề mặt

Một phép đo được thực hiện trong những lỗi (error) dự kiến xảy ra khi 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 trước và sau khi rút gọn lưới đa giác. Error metrics giữa hai mô hình là nhỏ, có nghĩa là hai mô hình gần giống nhau. Các độ đo khi tính toán việc giảm lưới gồm: edge length (độ dài các cạnh kết nối từ điểm đang xét đến các điểm láng giềng), distance to plane (khoảng cách từ điểm tới mặt phẳng chứa láng giềng của nó), curvature (bề mặt cong của các mặt phẳng chứa điểm), volume (thể tích), quadric error metrics.

Phương pháp áp dụng công thức "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 4.** Rút gọn lưới theo tổng góc Curvature

Khi đó xấp xỉ độ cong bề mặt K tại đỉnh $v_0$ được tính theo công thức (1):

$$K = 2.\pi - \sum \theta_{i,j} \quad (1)$$

Với $\theta_{i,j}$: là các góc tại đỉnh $v_0$ sinh ra từ các cạnh kề nhau cùng kết nối đến đỉnh $v_0$.

theo công thức trên thì, với K càng nhỏ thì đỉnh $v_0$ càng dễ xóa bỏ. Bởi vì với một điểm bất kỳ nằm trên một mặt phẳng thì tổng các góc xung quanh nó là $2\pi = 360^0$ , khi đó K = 0. Thì việc xóa điểm $v_0$ sẽ không hề ảnh hưởng đến hình dạng của mô hình. Một ngưỡng α được cho trước để thỏa mãn điều kiện loại bỏ điểm $v_0$, thuật toán chi tiết**:**

Khởi tạo:

∀v ∈ V: v.errormetric: = CalcErrorMetric (v);

Sort_Vertexlist ();

Repeat:

{

Lấy $v_0$ (một đỉnh trong lưới tam giác);

% If ($v_0$.errrormetric < α)

Loại bỏ ($v_0$);

Lưới tam giác (Nghb ($v_0$));

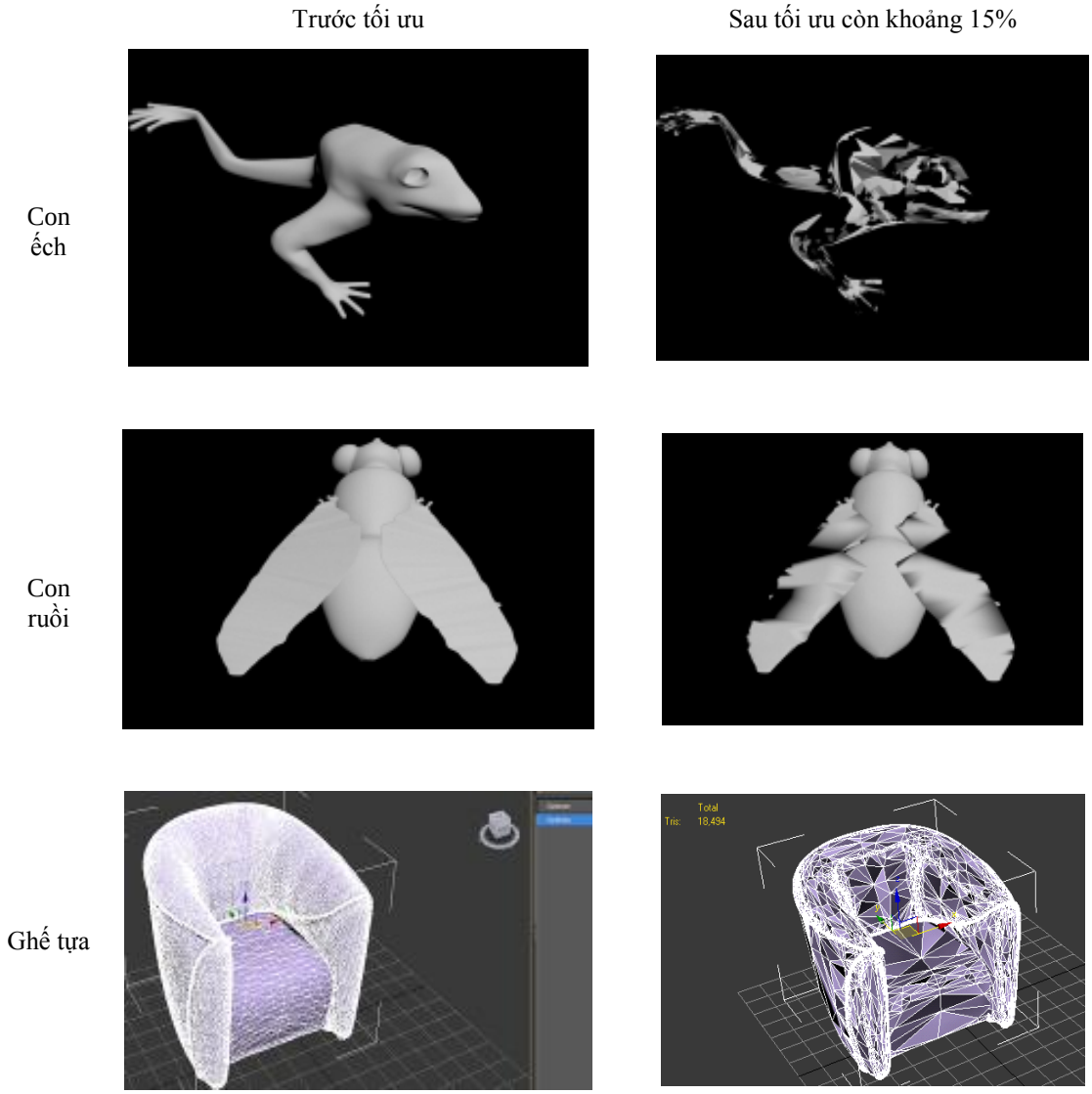
Cập nhật ($v_0$):

% Else break;

}

Until Mục tiêu cần giảm là đạt.

***C. Một số lỗi xảy ra khi rút gọn lưới mô hình bằng lệnh optimize(tối ưu) trong 3Ds max***



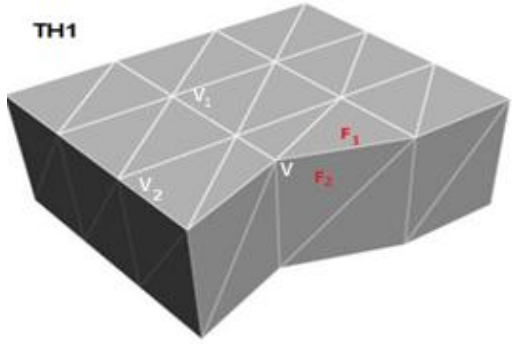

**Hình 5.** Một số lỗi xảy ra khi rút gọn lưới trong 3Ds Max

***D. Thuật toán đề xuất***

Dựa vào công thức (1), chúng tôi đã thử nghiệm cho một số mô hình sẽ thấy rõ một số lỗi khi chúng ta thực hiện rút gọn lưới. Để khắc phục tình huống trên, chúng tôi xin đưa ra một số giả thiết sau. Chúng tôi vẫn sử dụng phương pháp Gassian Curvature để rút gọn lưới của mô hình, nhưng chúng tôi nhận thấy một số trường hợp ngoại của các điểm trên bề mặt mô hình không thể áp dụng theo phương pháp này.

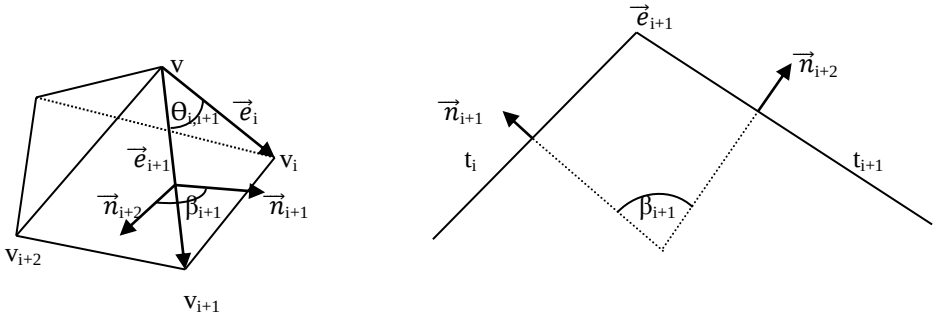
Trường hợp thứ nhất, trong *hình 6* những điểm $v_2$ nằm trên đường thẳng, mà đường thẳng này là giao của hai mặt phẳng vuông góc với nhau thì tổng góc xung quang điểm đó là $360^o$, và những điểm v là giao của ba mặt phẳng có tổng các góc xung quanh lớn hơn $360^o$, khi đó công thức (1) được thay thế bằng công thức (2):

$$\overline{K} = |2.\pi - \sum_{i,j=1}^{n} \theta_{i,j}| \quad (2)$$




**Hình 6**. Góc tại đỉnh v

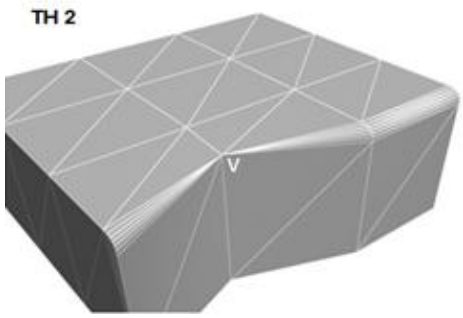
Do đó, đỉnh v tại vị trí này sẽ được phân biệt với đỉnh $v_1$ đó là véctơ pháp tuyến của hai mặt phẳng chứa hai tam giác kề nhau có chung đỉnh v không được lớn hơn một giá trị λ cho phép.



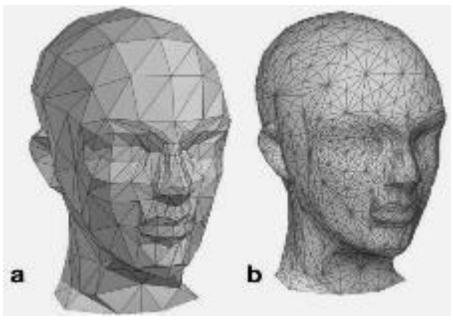

**Hình 7.** Góc tại đỉnh v và góc giữa hai mặt phẳng kề nhau

Chúng xác định các cạnh $\vec{e}_{i+1}$ là cạnh nối giữa v và $v_{i+1}$, góc giữa hai cạnh kề nhau liên tiếp $\theta_{i,i+1} = ∟(\vec{e}_i, \vec{e}_{i+1})$. Tam giác giữa $\vec{e}_i$, $\vec{e}_{i+1}$ được đặt tên $t_i = \Delta(v, v_i, v_{i+1})$, véctơ pháp tuyến tương ứng là $\vec{n}_{i+1} = \frac{\vec{e_i} \times \vec{e}_{i+1}}{||\vec{e_i} \times \vec{e}_{i+1}||}$ . Trên cạnh $\vec{e}_{i+1}$ góc giữa hai véctơ pháp tuyến của hai tam giác $t_i$ và tam giác liền kề $t_{i+1}$ là $\beta_{i+1} = ∟(\vec{n}_{i+1}, \vec{n}_{i+2})$.

Trường hợp thứ hai, tại một điểm đang xét mà góc quanh điểm khi áp dụng công thức: $\overline{K} = |2\pi - \sum_{i=1}^{n} \theta_i|$ nhỏ hơn một ngưỡng α cho phép và góc giữa hai véctơ pháp tuyến của hai tam giác kề nhau nhỏ hơn một giá trị λ cho trước. Tuy nhiên, số đường kết nối đến điểm v là rất nhiều hay bậc của đỉnh v là cao. Trong trường hợp này, chúng tôi sẽ giữ lại đỉnh v có số lượng cạnh kết nối đến nó là nhiều hơn số lượng *m* cho phép.

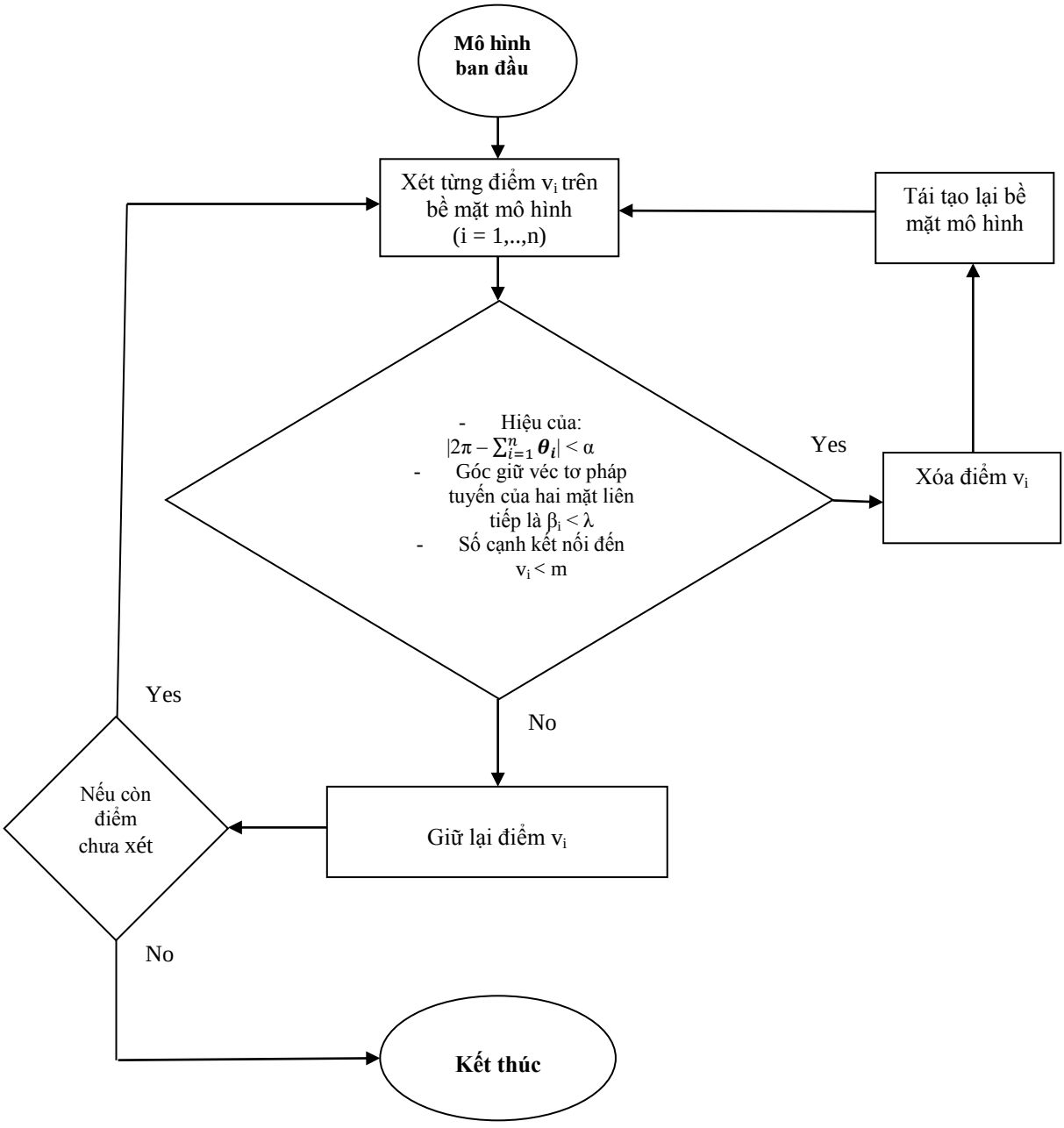



**Hình 8.** Đỉnh O với nhiều cạnh kết nối




**Hình 9.** Mô hình sau và trước tối ưu

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




**Hình 10.** Sơ đồ khối việc xóa điểm

Một tập tam giác lưới M bao gồm một tập các đỉnh $V = \{v_i\} \subset IR^3$, được kết nối bởi một tập các cạnh $E = \{e_j = (v_{j1}, v_{j2})\}$ và một bộ tam giác $T = \{t_k = \Delta(v_{k1}, v_{k2}, v_{k3})\}$ . Với $v \in V$ là một đỉnh của một tam giác lưới M và để $v_1$, ..., $v_n$ là các đỉnh lân cận của v.

Thuật toán đề xuất chi tiết:

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

CalcErrorMetric (v) = $|2\pi - \sum_{i=1}^{n} \theta_i|$;

Sort_Vertexlist ();

Repeat:

{

Lấy $v_0$ (một đỉnh trong lưới tam giác);

% If $((v_0.\text{errormetric} < \alpha)$ and $(\measuredangle(\vec{n}_{i-1}, \vec{n}_i) < \delta)$ and $(\vec{e}_i ; i<m))$

Tìm hàng xóm của ($v_0$);

Loại bỏ ($v_0$);

Lưới tam giác (Nghb ($v_0$));

Cập nhật ($v_0$):

% Else break;

}

Until Mục tiêu cần giảm là đạt .

## III. KẾT QUẢ THỰC NGHIỆM

### *A. Kiểm tra các mô hình đầu vào, lựa chọn công cụ*

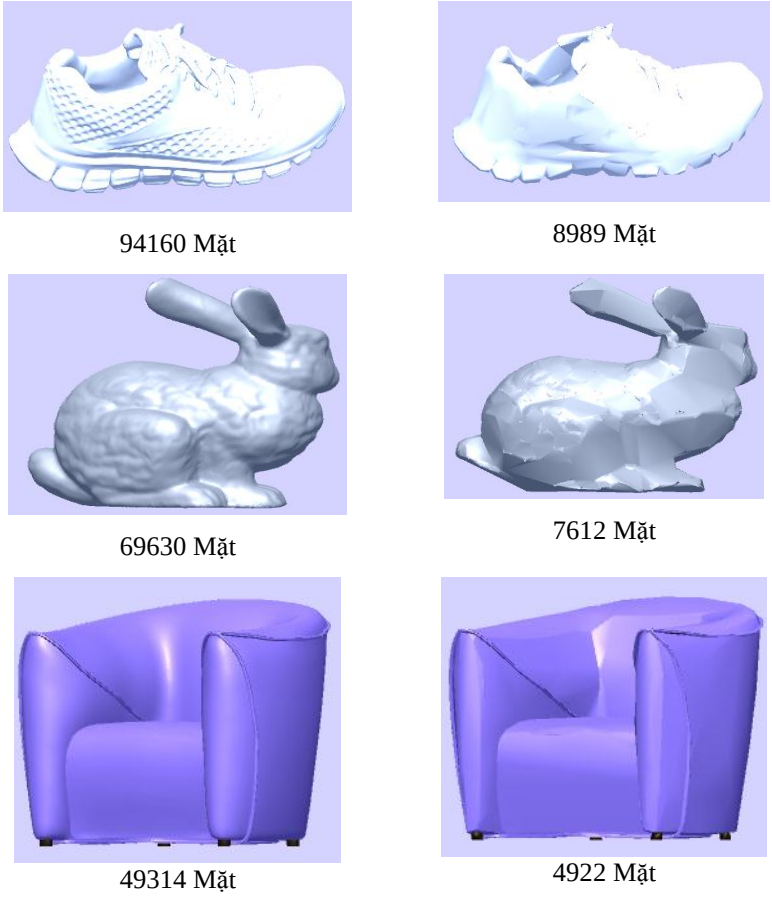
Các mô hình để test là các mô hình dưới dạng 3D, có số lượng lưới bề mặt lớn. Một số mô hình là đã được thử nghiệm cho các bài thử nghiệm trong các bài toán rút gọn lưới trước đó, ví dụ như mô hình con thỏ, một số mô hình còn lại chúng tôi tự tạo ra bằng các phần mềm 3D hoặc siêu tập trên internet...

Dựa trên những nghiên cứu về các kỹ thuật rút gọn lưới mô hình đã có, chúng tôi sử dụng phương pháp độ cong bề mặt Gaussian 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 tổng góc để xóa điểm. 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 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, hoặc được mô hình hóa bởi các phần mềm 3D nhưng có số lượng lưới lớn. Các tham số của từng mô hình là số lượng điểm, cạnh sẽ suy ra được số lượng mặt. Tuỳ vào mô hình khác nhau mà việc tính toán rút gọn lưới bề mặt có thể thay đổi trong chương trình.

### *B. Một số kết quả rút gọn mô hình trên chương trình thực nghiệm*

Rút gọn bề mặt lưới với số lượng lưới còn lại là khoảng 10%




**Hình 11.** Hình ảnh mô hình trước và sau tối ưu với tham số tối ưu là khoảng 10%

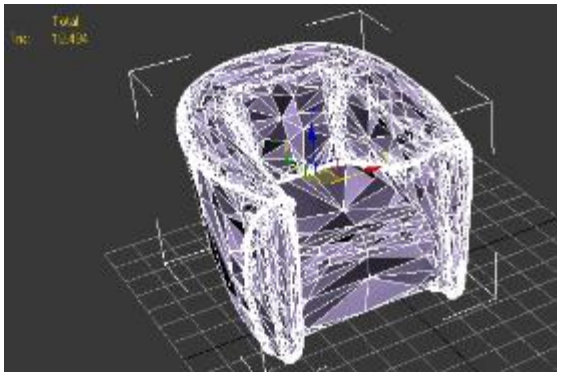
Với tham số tối ưu là khoảng 10% thì sự khác biệt rõ giữa mô hình trước và sau khi thực hiện tối ưu. Tuy nhiên nhiều chỗ của mô hình bị méo mó nặng.

So sánh kết quả chạy trên chương trình với kết quả chạy bằng lệnh optimize trong 3Ds Max cho kết quả sau:

KQ thử nghiệm khoảng 10% lưới

4.922 Mặt

KQ trên 3Ds max với khoảng 30% lưới



18.494 mặt

**Hình 12.** So sánh với tối ưu trong 3Ds max

## IV. KẾT LUẬN

Trong nghiên cứu này, chúng tôi áp dụng phương pháp bề mặt cong Gassian Curvature tính tổng góc của một điểm trên bề mặt để giải quyết việc rút gọn lưới bề mặt. Chương trình kết quả này giúp người thiết kế 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 lưới bề mặt là thấp nhất có thể. Một công việc làm khiến các công ty tốn rất nhiều thời gian và tiền bạc để thuê các nhà thiết kế.

Bài toán rút gọn 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. Nhóm tác giả hi vọng báo cáo 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 vị đã quan tâm và bỏ thời gian đọc báo này.

## LỜI CẢM ƠN

Chúng tôi xin cảm ơn sự tài trợ từ đề tài "Phát triển hệ thống thông tin hình ảnh ba chiều hỗ trợ chẩn đoán từ xa" do Viện Công nghệ Thông tin, ĐHQG Hà Nội làm chủ trì cho nghiên cứu này.

## TÀI LIỆU THAM KHẢO

[1] 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.

[2] 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 BasisFunctions.*

[3] Enrique Valero, Antonio Adan(*2012), Automatic Construction of 3D Basic-Semantic Models of Inhabited Interiors Using Laser Scanners and RFID Sensors.*

[4] David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, Robert Huebner (2002),*Level of detail for 3D graphics*, Morgan Kaufmann.

[5] Nira Dyn, Kai Hormann, Sun-Jeong Kim, and David Levin(2000), *Optimizing 3D Triangulations Using Discrete Curvature Analysis*.

[6] Kobbelt, L., *Discrete fairing, in The Mathematics of Surfaces* VII,T.Goodman and R. Martin (eds.), Clarendon Press, Oxford, 1997, 101-131.

[7] Steven J. Owen, Matthew L. Staten, Scott A. Canann and Sunil Saigal(1998), *Advancing Front Quadrilateral Meshing Using Triangle Transformations*,

[8] 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.*

[9] David Bommes, Timm Lempfer, Leif Kobbelt (2011), *Global Structure Optimization of Quadrilateral Meshes.*

# A TECHNIQUE FOR 3D MESH SIMPLIFICATION IN THE APPLICATIONS OF VIRTUAL REALITY

**Ma Van Thu, Do Nang Toan, Le Son Thai, Do Hong Quan**

***ABSTRACT:*** *Virtual reality is a technology that uses 3D space modeling to simulate the real world or the human imaginary world. 3-D models in virtual reality can be static, dynamic and can interact with each other. The big problem here is that the models created to satisfy the aesthetic demands of humans often have a tremendous amount of grid surface. With such a model, some devices running virtual reality applications must meet the configuration and bandwidth of the internet. This paper demonstrates a technique for simplifying the mesh surface of a 3D model based on the "Gaussian Curvature" formula incorporating a number of constraints, so that the model after the retraction of the mesh remains secure in shape in the application.*