The Effect of Curvature Estimations and Lighting Factor in the Edge Detection for 3D Rendered Images

Alaa Noori Mazher
Computer Engineering Department, University of Technology/Baghdad
Email: laa_19782002 @yahoo.com

Received on: 29/5/2012 & Accepted on: 15/8/2012

ABSTRACT

This paper presents new approach in the edge detection of 3D images (for non-photorealistic rendering). This new approach uses curvature estimations with lighting effect on triangle mesh as a new factor in edge detection process. New algorithm to detect edges on 3D curved surface with results is presented. In the proposed technique, the curved surface is generated by using Bezier curves and those generated curves are connected together using a special fashion to generate triangle mesh. Depending on those generated curves, the direction of surface becomes defined. The curvature is then calculated at defined points on each curve. The estimated curvature and lighting effect determine if there is edge can be detected or not. The size of the step of jumping between two defined areas determines the accuracy of detecting edge. The algorithm is tested on curved-surface which on it the edges are hardly to detect.

Keywords: Triangle Mesh, Curvature, Edge detection.

تأثير حسابات المنحنى وعامل الاضاءه في اكتشاف الحواف للصور ثلاثية الابعاد المعاد

الخلاصة

يقدم هذا البحث اسلوب جديد في اكتشاف الحافة للصور ثلاثية الابعاد للاعادة من نوع non – photorealisitic هذا الاسلوب يستخدم حسابات المنحنى مع تأثير الاضا ِة على الشبكه المثلثيه كعامل لاكتشاف الحافة. الخوارزمية الجديدة لإكتِشاف الحافاتِ على السطحِ المقوس بالنَتائِجِ مُقَدَّمةٌ . في التقنيةِ المُقتَرَحةِ، السطح المقوس يولد من منحنى بزاير وهذه المنحنيات تربط مع بعضها باسلوب خاص لتوليد الشبكه المثلثيه. حسابات المنحنى مع عامل الاضاءه هي التي تحدد اكتشاف الحافه او لا. حجم خطوةِ القفز بين منطقتين مُعَرَّفتينِ تُقرّرانِ دقةَ إكتِشاف الحافةِ. الخوارزميةَ مُجرّبةٌ على السطحِ المقوس الذي عليه الحافات بالكاد أَنْ تكتشفَ .

https://doi.org/10.30684/etj.30.16.2
2412-0758/University of Technology-Iraq, Baghdad, Iraq

INTRODUCTION

With the development of 3D scanning technologies and modeling tools, raw meshes can be composed of thousands to millions of polygons. Real-time rendering of such a large amount of data is always a challenge. Some researches [1] are directed to propose mesh simplification algorithms to decrease the complexity of models while maintaining similarity with the original models. This helps many other algorithms such as edge detection algorithms to decrease time which is spent in processing (detection process). Other researches are directed to propose a new detectors or edge detection algorithm with many tradeoffs such as time, and accuracy of the proposed technique to define and detect edges.

Recently many research deal with the Normal computations in edge detection algorithms for 3D Rendered images as in [2, 3].

By experience, calculations of Normals consume more time than any other factors; Normals contain the Cross-product in their computations. Detecting edges using Normals have been proposed by most researches.

There are few researches related to image rendering in non-photorealistic rendering field, edge detection process is very important to the rendering especially in the smoothing and fairing techniques that may be applied to 3D object's surface. Many researches deal with the problem of edge detection based on triangle mesh that can be constructed for creation of object. Almost of these researches depend on normal calculations or curvature estimations as condition to detect edges.

Normals can be described the surface nature (geometrical properties) according to the lighting fall on it, the use of Normals became very common in edge detection. After applying the main condition (which is based on Normals) by it the edge can be detected, the most time which is spent to detect edges was used in Normal calculations and this time increases when mesh becomes dense. This important reason was the first step to think about new approach in edge detection by using tangent line as new 3D edge detector.

In this paper, new approach for detecting edge without using Normal vectors is proposed, tested and applied on a surface which includes many significant properties.

THE SUGGESTED ALGORITHM

Firstly, the surface is generated from Bezier curve function; this function is used to create triangle mesh by which the surface is constructed. The groups of curves must be generated, after that those curves must be connected to gether by choosing set of points with equal distance between each other and connect the point 0 of curve1 with point 1 of curve 2 (i.e. in zigzag fashion) as in figure (2.a).

Step 1: Choose point P0 which belongs to the surface.

Step 2: Add small value to the p value, such that the added value is not exceeding the defined rate: the x and y values respectively of light co ordinate * 0.1 (by experience).

Step 3: Use function of curve (which is used in creation of surface) and P0 which is defined above for calculating the curvature K0 at point p0 by finding:

The second derivatives of the curve function at the selected point p (x,y) in term of x. or in term of y by using the defined functions 1 and 2 previously.

Step 4: repeat the step 1 to calculate the curvature K1 in term of p1, this new point can be chosen randomly but with a close distance to the previous point and must be not far away from it.

Step 5: Find the difference between the two calculated curvatures k0, k1, and compare with predefined threshold.

If dif>threshold then there is an edge its direction represents the direction of the tangent line.

Step 6: GOTO step1 to choose another points and start again to detect new edge if it existed.

Step 7: End.

DEFINITION OF EDGE

Edge is defined as significant local change of intensity in an image. Typically it occurs on the boundary between two different regions in an image.

There are significant reasons for the changing in intensity; these reasons can be divided into two types: geometric events and non-geometric events.

Geometric events are represented by either object boundary or surface boundary. Discontinuity in depth and/or surface color and texture is related to object boundary, whereas discontinuity in surface orientation and/or surface color and texture is related to surface boundary. For non-geometric events, the change of specularity or shadows-from other objects or from the same object- or inter-reflections (reflectance), or illumination boundaries, all of these reasons are considered non-geometric events [4].

The most common methods to find these changes are:

1-Highlight maxima and minima in the first derivative of an image.

2-Finding zero crossing in the second derivative.

The basic edge detection operator is a matrix area gradient operation that determines the level of variance between different pixels. The edge detection operator is calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the value of this matrix area is above a given threshold, then the middle pixel is classified as an edge.

In computer vision, edge detection is traditionally implemented by convolving the signal with some form of linear filter, usually a filter that approximates a first or second derivative operator.

EDGE CLASSIFICATION IN 3D IMAGES

Typically, edges in 3D image are different from edges of 2D image but each of which meet at the same point. Theoretically, the two types are the same, but in 2D the edge appearance depends on intensity change of light, and in 3D the edge appearance depends on the reason by which the intensity is changed. This reason is discontinuity in depth, i.e. geometrical reason. The recent algorithms of edge detection in 3D image use this concept in detection process. 3D objects can be obtained by new topic in computer graphics is called non-photorealistic rendering (NPR). The edge can be classified into three types as follows [5, 6, and 7]:

Assume that 3D scene objects are represented by polygonal meshes:

(a) A silhouette edge is an edge adjacent to a polygon facing toward the camera (front-facing) and one polygon facing in the opposite direction (back facing). The silhouettes on curved surfaces vary smoothly with change in viewpoint

(b)A border edge is an edge to exactly one polygon.

(c) A crease edge is an edge between front-facing (or back- facing respectively) polygons whose

Directional angle is above some threshold. Angle called dihedral angle defines the intensity of

Crease edge, i.e. They correspond to edges shared by two front-facing polygons, whose normal

Vectors make an angle q within the limits (min,max) specified by the user; we assume that 0 <= min<= q<= max<= 180.

Edge detection of 3d image has important application to surface segmentation and adaptive surface smoothing as in [8].

CURVATURE DEFINITION

The curvature K of a curve y=f(x) at any point p on it is defined to be the rate of change of the direction of the curve at p, that is, of the angle of inclination t of the tangent line at p, with respect to the arc length s. (see fig 1) intuitively, the curvature tells us how fast the tangent line is turning. Thus, the curvature is large when the curve bends sharply [9, 10].

As formulas for the curvature, we get:

$$K = \frac{dt}{ds} = \underset{\Delta s \to 0}{lim} \frac{\Delta\tau}{\Delta s} = \frac{\frac{d^2y}{dx^2}}{\left(1+\left(\frac{dy}{dx}\right)^2\right)^{3|2}} \qquad \ldots\ldots. (1)$$

$$K = \frac{-\frac{d^2y}{dx^2}}{\left(1+\left(\frac{dy}{dx}\right)^2\right)^{3|2}} \qquad \ldots\ldots. (2)$$

K is sometimes defined so as to be positive. If this is assumed, then the sign of k should be ignored in what follows.

MESH GENERATION (SURFACE CONSTRUCTING)

There are many methods are used in mesh generation. In this paper the triangle mesh is generated in a special manner by using Bezier curve function. The first curve is generated by using Bezier function:

The formula for a degree three Bezier curve is [11]:

$$Q(u)=B0(u)p0+B1(u)p1+B2(u)p2+B3(u)p3 \qquad \text{........ (3)}$$

Where the four functions Bt(u) called blending functions, are scalar-valued. The blending functions Bt(u) are clearly degree polynomials. Indeed, when their definitions are expanded they are equal to;

$$B0(u)=(1-u)3,\ B2(u)=3u2(1-u) \qquad \text{…….. (4)}$$

$$B1(u)=3u(1-u)2,\ B3(u)=u \qquad \text{……….(5)}$$

These defined points of appearance rate control the shape of generated curve. One curve is generated for each part of surface. And all of those curves represent the whole surface in 3d space.

Firstly, the surface is generated from Bezier curve function; this function is used to create triangle mesh by which the surface is constructed. The groups of curves must be generated, after that those curves must be connected together by choosing set of points with equal distance between each other and connect the point 0 of curve1 with point 1 of curve 2 (i.e. in zigzag fashion) as in figure 2 .

CURVATURE ESTIMATION WITH LIGHTING EFFECT ON A TRIANGLE MESH AS 3D EDGE DETECTOR

To detect edge of 3D object, suggested edge detector is presented. In this paper the curvature estimation is used with lighting effect on triangle mesh (curved surface) to detect edge. Its effect in detecting edge is evident on a triangle mesh. The curvature estimation has an important property in determining orientation of surface and speed of curved surface. Curvature calculations have strong relationship with normal plane. As result the curvature has a strong relationship with the lighting factor on the surface (Normal line is perpendicular on tangent plane and tangent line is perpendicular on normal plane).

To detect edge on triangle mesh , the lighting factor (lighting direction value is added to the estimate value of curvature by adding small threshold to the x, and y co ordinates of the selected point p, then the curvature in defined points on the surface (triangle mesh) must be calculated. Increasing in curvature lead to make a large difference between those calculated angles as a result this leads to detect edge. The comparison is done by using difference of those angles via calculate the curvature of a defined part of surface (defined part of surface represents one or more generated curve) above defined threshold.

The jumping step from one point p=(x,y) to another determines the accuracy of calculating curvature and as a result detecting the edge. Where the step size from point to another at the same curve (from area to area) is very important. If the jumping step is large as result the edge detection routine may exceed the area in which the edges may be existed (decreasing in accuracy, decreasing in processing time), whereas if the step size is small as a result many edges may be detected in neighbors areas i.e. no edge can be existed and is not detected (as a result increasing in processing time) or the result may be negative i.e. the changing in curvature may be not clear in very small neighbored areas, as a result no edge can be detected.

In the suggested algorithm, Bezier curve function is used to define the parts of surface.

The suggested algorithm works independently of mesh type which is used to generate surface. To make the algorithm do work generally; the surface must be created by using matrix of closed vertices and this matrix is generated by one function of curve with small shifting in x values in order to create surface as group of closed curves, each curve represents set of vertices and the whole group of sets represent the surface. Then the surface is constructed by connect each curve with it neighbor. As result the related mesh to that surface can be seen as triangle mesh or even rectangular mesh. The proposed algorithm can detect two types of edge: border and crease edges that reflect important surface information in 3D data .

IMPLEMENTATION AND APPLICATION

In this section, we describe the implementation of our algorithm and highlight some applications of our edge detection algorithm. Fig.2 illustrates those applications.

The proposed technique is implemented in Visual C++ with graphical interface OpenGL for rendering 3D scene (for non-photorealistic rendering). OpenGL (GL for Graphic Library) is a software interface to graphics hardware. This interface consists of about 120 substantially different functions that allow graphics programmer to specify the objects and operations needed to produce interactive three-dimensional applications [12, 13, and 14].

To make Visual C++ works correctly with graphical interface, OpenGL/GLUT must be set up on windows firstly by downloading GLUT [10]. GLUT is (OpenGL Utilities Toolkit) which contains functions to create and manage windows.

Four different applications are illustrated in figure.(2) in the appendix A**.**

CONCLUSIONS AND FUTURE WORK

In this paper, 3-D edge detector is proposed; it can detect two important types of edges: creases and borders. The algorithm is implemented in VC++ with OpenGL (Graphical interface to render 3D object). The proposed algorithm works independently of type of mesh which is used in the creation of surface, where the curvature and light directions are used in this algorithm (geometry of surfaces).

By the proposed technique the edges can be detected even in the dark area because the proposed technique uses the factor of the changing in depth of surface – the changing in the depth causes the difference in the intensity of light from an area to an area on the surface (i.e. edge creation)-. As future work, the tangent line can be investigated in detecting of silhouette by using specific criteria. Border edge can be detected by process which is recently called relief edge, and this could lead to what is called image understanding (feature extraction and find its relationship with lighting factor and other important factors), where the proposed algorithm of the new edge detector detects the border edge but in a few areas of surface, it needs to be completed by defining the direction of detected edges by the proposed algorithm.

REFERENCES

[1] Hoover, A., Jean-Baptiste, G. & Jiang, X. [1996], an Experimental Comparison of Range Image Segmentation Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(7).

[2] Timothy J. Baker, Identification and preservation of surface features, Dept. of MAE, Princeton University, Princeton, NJ, USA, Baker@tornado.princeton.edu, 2004.

[3] Sun,Y. D.Lpage, J.K. Paik, and others, Triangle mesh based edge detection and its application to surface segmentation and adaptive surface smoothing, © 2002, IEEE Paper.

[4] Marr, D. E. Hildreth, Theory of Edge Detection, Processing of the Royal Sociaty of London. Series B, Biological Science, Vol.207, No.1167 (Feb.29, 1980) pp.187.217.

[5] Mario Costa Sousa, Przemyslaw Prusinkiewicz, A Few Good Lines: Suggestive Drawing of 3D Models, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, 2003.

[6] Hertzmann, A. introduction to 3 D Non-Photorealistic rendering: Silhouettes and outlines, media research laboratory, Department of Computer Science, New York University, available in http://www.mrl.nyu.edu/hertzmann

[7] Ramesh Raskar, Kar-Han Tan, and others, Non-photorealistic Camera:Depth Edge Detection and Stylized Rendering using Multi-Flash Imaging, Mitsubishi Electric Research Labs (MERL).

[8] Sun,Y. D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi, Triangle Mesh-Based Edge Detection and its Application to surface segmentation and Adaptive Surface Smoothing, Knoxville, TN 37996, USA, ,yiyong@iristown.engr.utk.edu

[9] Frank Ayres, Jr., Ph.D., Elliot Mendelson, Ph.D., Calculus, (fourth Ed.)The McGraw-Hill, Inc., 1999.

[10] Berkey and P. D. Blanchard, Calculus (3rd ed), Saunders, 1992.

[11] Jay Whitehead, Finding the Equation of a Tangent Line Using the First Derivative ", Student Learning Assistance Center (SLAC), Texas State University- San Marcos, March 1999.

[12]M.Woo,J.Neider,T.Davis,D.Shreiner.OpenGL®Programming Guide: The Official Guide to Learning OpenGL, ver. 1.2, 3rd. ed., Addison-Wesley, 1999

[13] OpenGL, M. W, Programming Guide, the Red book, Fifth edition.

[14] OpenGL 1.2 Reference Manual, Second Edition (October 2000), © Copyright International Business Machines Corporation 1994, 2002

APPENDIX A

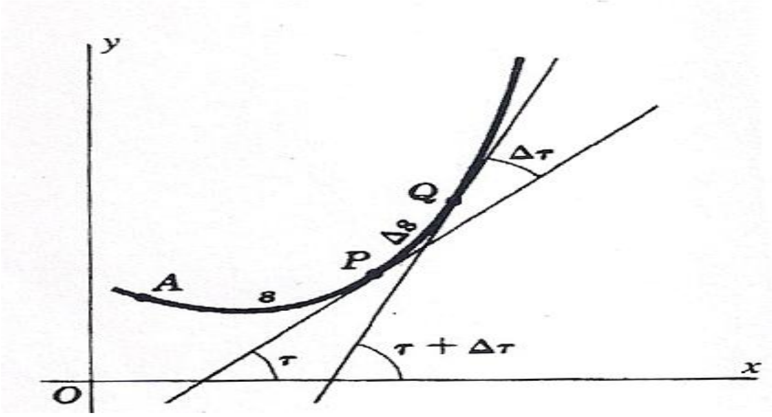


Figure (1)

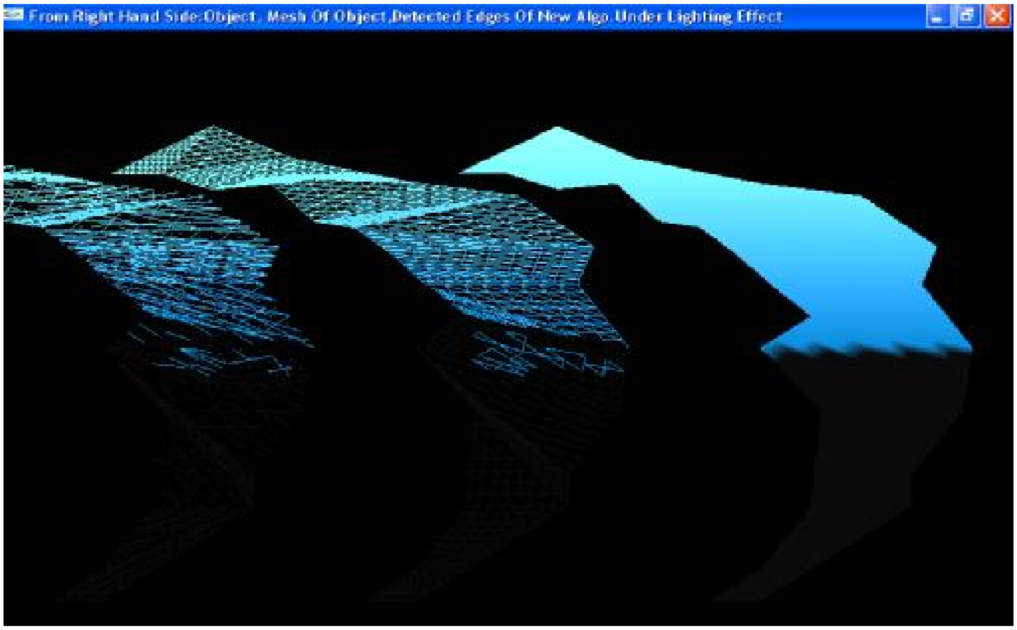
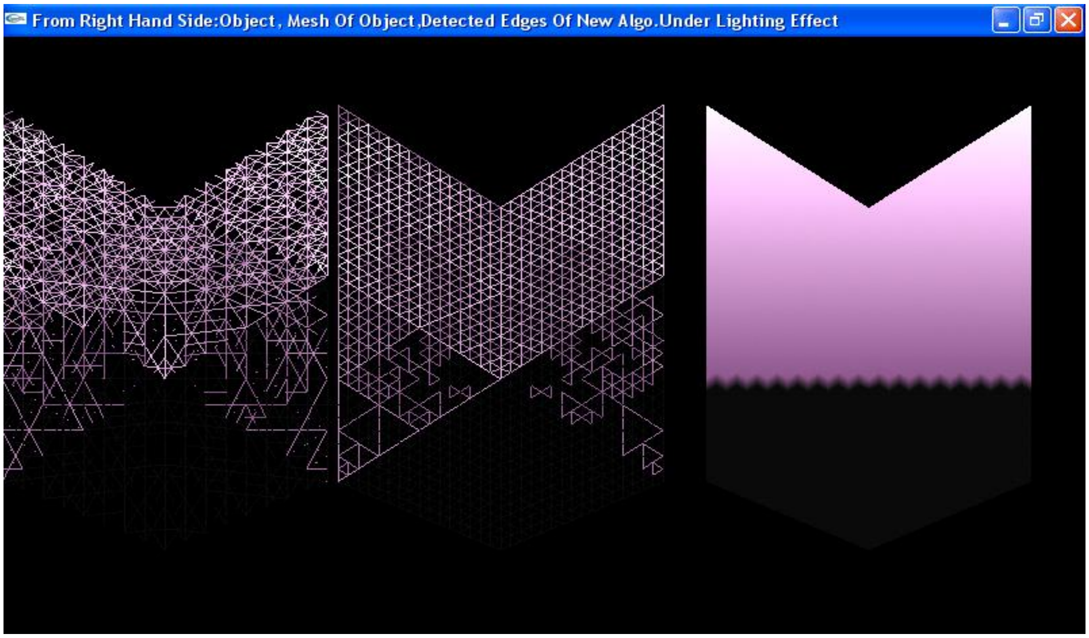

From Right Hand Side:Object, Mesh Of Object,Detected Edges Of New Algo Under Lighting Effect

Figure (2 a)



From Right Hand Side:Object , Mesh Of Object ,Detected Edges Of New Algo.Under Lighting Effect

Figure (2b)

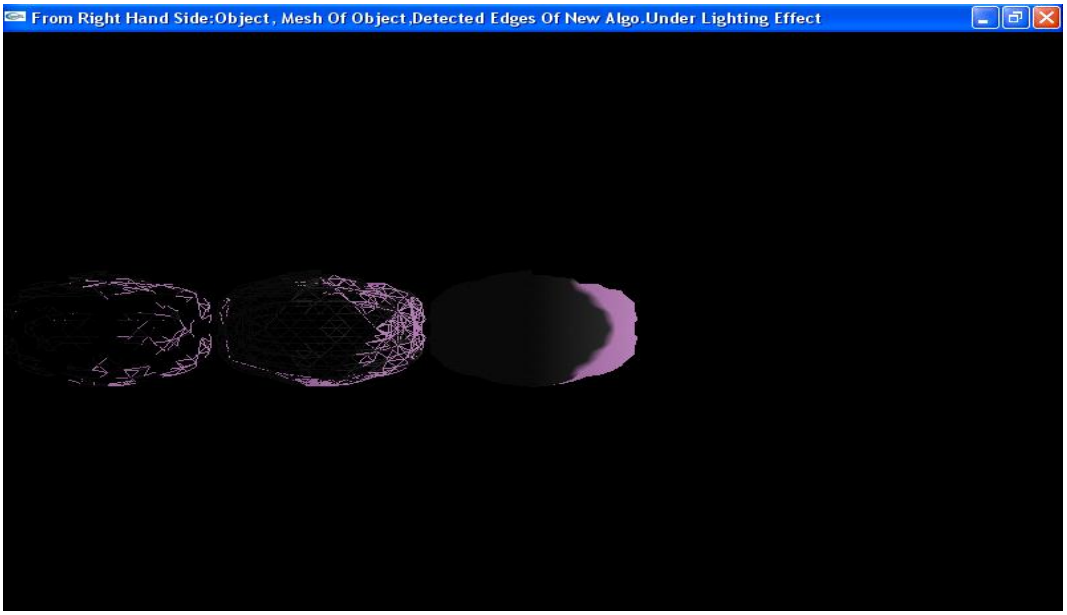

From Right Hand Side:Object, Mesh Of Object,Detected Edges Of New Algo.Under Lighting Effect

Figure (2C)

Figure. (2): three different applications of 3D curved surfaces with the same threshold values. (a) 3D curved surface with light position [5.0, 1.9, 1.0, and 10.0]. Border and crease edges are detected by the new algorithm. (b) and (c) Another 3D curved surface which is created by using the same function of the surface in (a) but with different variables had been given to x, y, and z. borders and creases edges are seen clearly. Detected borders and creases are seen clearly.