Alignment of 3-D Scanning Data for Polygonal Mesh based on Modified Triangulation

Masud An Nur Islam Fahim, Sakib Mostafa, Jarin Tasnim, Dr. A. B. M. Aowlad Hossain
Khulna University of Engineering & Technology Khulna-9203, Bangladesh
mostofa21fahim@gmail.com, sakibmostafa11@gmail.com, jarintasnim.j@gmail.com, aowlad0403@ece.kuet.ac.bd

***Abstract*— Alignment of 3-D scanning data is important in holography, gesture based gaming, statuette modelling, tomography and so on. All of this sector are advancing day by day on the basis of computational geometry. One of the most complex open problems in the sector of combinatorial and computational geometry is to deal with convex hull. This type of problem commonly arises in the several cases like meshing, 3D data alignment etc. For generating 3D model first, we have to compute the triangulation step. In this study, a divide and conquer based way of triangulation is presented. Furthermore, detecting the point of interest in the vornoi region as well as discarding of other points is also an important feature of the study. The outcome of the study shows effective reconstruction with legit triangulation, less computation time for determining convex hull and a smoothed polygon through proper edge flipping.**

Keywords—Delaunay Triangulation; 3D scanning data; reconstruction; mesh; convex hull.

I. INTRODUCTION

3-D modelling is a very demanding technological manner is being practiced by not only the engineering people but also in the sector arts and medical science. Historical place reconstruction and modelling, Bone sculpture modelling, Tomography, Virtual reality and so on are the generic practices of three-dimensional imaging.

The progress in 3D reconstruction techniques using uncalibrated or calibrated multi-view images enable people to comprehends 3D scene information from the real world. In most cases, the reconstructed 3D data are represented by a set of 3D points and are then converted to mesh structures via post-processing. The mesh data are then used for rendering, image texturing, deformations and other manipulations to obtain the realistic 3D model. However, constructing well-structured 3D meshes is a challenging task. Several complexities are being inherited within the computation. Various methods have been proposed to construct 3D meshes though researches on efficient 3-D meshing still demanding. General triangulation based approaches [1] often provide significantly poor performances in constructing 3D surface meshes when the given depth points fail to satisfy the theoretical concerns. Moreover, methods which provides functional utilities through voxel representation as an intermediate stage [2, 3] produce 3D meshes whose resolutions fails to maintain the quality of fine reflections of the given point sets. In addition to these technical difficulties, the scarcity of the depth points causes more difficulty in representing exact 3D shapes, including object boundaries and sharp edges [4]. There are several ways of solving these problems by flipping the edges in the triangular mesh [5]. 3D sound field based researches provides prominent results which is only applicable for specific triangulation cases [6]. In general, triangulation method is based on mapping and proper connections between the points [7]. Projecting the three-dimensional scatter points into two-dimensional tangent plan to implement triangulation and then filling the exact depth information into each vortex is basically known as the mapping method [8]. When the two-dimensional optimal triangulation mapped into three dimension, it will result in a large number of long and narrow triangles [9] and will be structured as over-segmented region. It has been shown that functionality of region growing method is the representation of direct triangulation algorithm. Initially it generates a primary triangle called seed triangular and using some triangulation rules to construct new triangle which based on the initial triangle edge [10]. Computational repetitions of the execution will give the final triangular mesh. The advantage of this method is easy to implement, fast execution and less time-consuming. The disadvantage is that when scattered points of surface had huge amounts of data or had a complex topology, the triangulations are overlap each other easily [11].

In this study, we plan to develop a method of triangulation of 3D co-ordinates in order reconstruct the mesh emphasizing on efficient edges computation as well as gaining better performance in boundary points determination from large datasets. Convex hull defining is the procedure of the setting the shape of the acquired dataset from the laser scanner or other means. Initially acquired dataset has been sampled into number of groups for determining the convex hull. After that connection between all the boundary points has been done by joining them through edges. Triangulation procedure can be started from any of the points of point sets. In the meantime, some constraints have been maintained for conserving the fineness and regularity of the total triangulation process such as changing the orientation of connecting edge if it's necessary to create a proper triangle generically known as edge flipping.

This paper is organized as follows. The methodology section provides the brief algorithmic representation of the calculation of the convex hull, triangulation constraint, and triangle determination process. Then the constructed meshes from various datasets, requirement of computation time and

978-1-5386-1023-7/17/$31.00 ©2017 IEEE

graphical views are shown, compared and discussed in the result analysis section. Finally, a conclusion is drawn mentioning the findings, limitation and future perspective of this work.

II. Methodology

The whole procedure begins with collecting the point sets from any rigid object by scanning it with a laser scanner.

Acquired point sets contains the geological information about the object as well as the color information. Defining this point sets through a variable P and considering an incident plane for doing any kind of operation which is known as Euclidian plane. Now dealing with the boundary points of the acquired point set is a challenging part due to larger calculation and this study shows emphasizing on a simpler method to deal with it easily. Again, determining the right edge for the triangulation is the fundamental part of the algorithm. This scheme is planning to define the legal edges more accurately than the conventional algorithm. Doing the triangulation with mentioned features properly, a mesh output has been achieved which has been shown in result analysis section in a comparative approach. Considering the acquired point set P as follows:

$p \coloneqq \{p_1, p_2, ..., p_n\}$ where *P* is a set of n distinct points in the Euclidian plane. Now the Voronoi diagram will be achieved by partitioning the plane into non-overlapping convex net. Mathematically, let $p \coloneqq \{p_1, p_2, ..., p_n\}$, where $x_i \neq x_j$ for $x \neq y$ then the Voronoi region $\{V(p_i)\}$ be defined as :

$$\{V(p_i)\}=\{P: |p - p_i| \leq |p - p_i|, \forall_j \neq i \qquad (1)$$

Then the Voronoi diagram of P can be defined as:

$$\{V(p_i)\}=\{V(p_1), V((p_2)), \ldots V((p_n))\} \qquad (2)$$

Now for a given Voronoi diagram, if each pair of sites of P whose Voronoi polygons share an edge is joined by a straight-line segment, then the result is a triangulation of the convex hull of the original n sites which is known as the Delaunay triangulation [12]. If a triangle T_k in the triangulation is named the Delaunay triangle, then the Delaunay triangulation can be written as:

$$T=\{T_1, T_2, T_3...., T_{nt}\} \qquad (3)$$

Maximal planar subdivision with vertex set P. Before going to triangulation of the points in the plane following observation has taken under consideration in a constructive manner. Now before going to triangulation some basic properties need to be conjunctional with the total working steps for preserving the practical outcome of the scheme.

Following observations has been maintained precisely to ensure proper triangulation from the input data set.

1. Let P be a set of n points in the plane, not all collinear, and let k denote the number of points in P that lie on the boundary of the convex hull of P. Then any triangulation of P has $2n - 2 - k$ triangles and $3n - 3 - k$ edges.

2. Let C be circle (figure), a line intersecting C in points a and b, and p, q, r, and s points lying on the same side of. Suppose that p and q lie on C, that r lies inside C, and that s lies outside C. Then

$\measuredangle arb > \measuredangle apb = \measuredangle aqb > \measuredangle asb$.

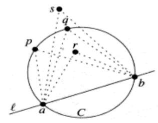

Fig. 1. Graphical Representation of Thales Theorem

3: A triangulation τ of P is legal if and only if τ is a Delaunay triangulation.

4: Any angle-optimal triangulation of P is a Delaunay triangulation of P. Furthermore, any Delaunay triangulation of P maximizes the minimum angle over all triangulations of P.

Now after obtaining the co-ordinates of the scanning data we have performed the alignment by following this triangulation steps:

Input: A set P of n+1 points in the plane.

Output: A Delaunay triangulation of P.

1. Initialize with a sampling of S samples of the points in the plane and let's say $S_{p_x} = \{S(p_{n1}), S(p_{n2}), ..., S(p_{nn})\}$ where S_{p_x} contains the total points in the plane which has been sampled into n groups. This step will be followed twice in horizontal and vertical axis direction. For vertical manner, we will obtain S_{p_y}.

2. Compute the boundary points by only determining the initial and final samples in both horizontal and vertical direction.

3. Determine the boundary points in the samples just by computing the maximum distances from fundamental axis.

4. If p_0 is the far most point then we can begin triangulation with p_{-1}, p_{-2}..

5. Begin the triangulation with a bigger triangle
   - Choosing of a point will be done in random fashion, say p_0 from *P*.
   - Now we will find the triangle Δ that p_0 lies in.
   - Distribution into subdivision of the Δ into smaller triangles that have P_0 as a vertex.
   - Edge flipping will be executed until all edges are legal.
   - Repeat steps 2-5 until all points have been added to T.

.

Now this is the regular triangulation algorithm which is generically known as Delaunay triangulation. Now we will introduce a divide and conquer based algorithm for this study.

Let's considering an equation of a line is for a two dimensional case is

$$L_1 \equiv y_1 = \frac{d}{dx} x_1 + C;$$

where y_1 showing the entity about the y axis and $\frac{d}{dx} x_1$ denoting the information about slope and entity about x axis and c stands for any constant. So we could write this equation for four straight lines which will be used for denoting a bounded region and they are

$L_1 \equiv y_1 = m_1 x_1 + C$, $L_2 \equiv y_2 = m_2 x_2 + C$, $L_3 \equiv y_3 = m_3 x_3 + C$, $L_4 \equiv y_4 = m_4 x_4 + C$

Now, we will consider a set of random points inside the region of this. So, we will find the Euclidian distance between points and the boundary lines in terms of shortest. This approach will help us to find out the boundary points of the data set. So after that having with the boundary points we will try find out voronoi diagram just by connecting those points. Let's denote this initial voronoi diagram as V_1. After this we will temporarily remove this V_1 out of the field and we will have another data set just like before with a less number of data points. Now we will repeat the step of voronoi diagram making step for dividing the dataset into another region and this loop will be continued till the last data points. Now we have a set of voronoi diagrams say V={ V_1, V_2 ,V_3,…, V_n}

So this dataset is been partitioned into n numbers of voronoi sets and now we will do the conquering portion by connecting those points by triangulating them. So following steps will cover the total version of the proposed algorithm.

Algorithm: Proposed D&C algorithm we followed in this paper

Input: A random point set either from a laser scanner or a Microsoft Kinect sensor, P_0

Output: The minimum triangulation of the data set **Tri**(P_0)

1. Create the boundary lines

   $L_1 \equiv y_1 = m_1 x_1 + C$, $L_2 \equiv y_2 = m_2 x_2 + C$, $L_3 \equiv y_3 = m_3 x_3 + C$, $L_4 \equiv y_4 = m_4 x_4 + C$

2. If l={ l_1, l_2 ,l_3, …, l_n} ∊ $L_{1\text{-}4}$ & p = {p_1, p_2, p_3, …, p_n} ∊ P_o
3. Then, compute $D(l,p) = \sqrt{\sum_{i=1}^{n} (l_i - p_i)^2}$
4. Connect the primary points and generate the V_i; where i=0.
5. Remove V_i .
6. Continue steps 2 to 4 to the last point or last set of points.
7. Let V={ V_1, V_2 ,V_3,…, V_n} the set of graph G(v,e) where v stands for vertices and e stands for the edges.
8. Choosing of pivotal edge will by random selection algorithm.
9. Identification of edges will be done by considering weight consideration comparing with the pivotal edge. They are less weighted than pivot edge and e' is the set of it.
   - Now Fun_Con {G'(v,e')} will be done and steps for are P ← { } +> Set P is sum of edges
   - **for** each vertex *v* in V
   - **do** MAKE-SET(*v*)
   - sort e' into incremental order by weight *w*
   - **for** each (*u*, *v*) taken from the sorted list
   - **do if** FIND-SET(*u*) = FIND-SET(*v*)
   - **then** P ← P ∪ {(*u*, *v*)}
   - UNION(*u*, *v*)
   - **return** P
10. If the member of connected component in fun_con output is exactly one
11. Then it's the desired connection
12. Else
13. Construct {G''(v'',e'')} ; where each connected component is a vertex in v''.
14. If fun_con1 can be stated as follows:
    - Make a queue (Q) with all the vertices of G (V);
    - For each member of Q set the priority to INFINITY;
    - Only for the starting vertex (s) set the priority to 0;
    - The parent of (s) should be NULL;
    - While Q isn't empty Get the minimum from Q – let's say (u); (priority queue);
    - For each adjacent vertex to (v) to (u)
    - If (v) is in Q and weight of (u, v) < priority of (v) then
    - The parent of (v) is set to be (u)
    - The priority of (v) is the weight of (u, v)

fun_con1 {G''(v'',e'')} will give the desired output

As a result of executing these steps allows us to determine the proper triangulation, voronoi diagram, less computation to initialize the convex hull. Obtained results have been shown in result analysis portion.

III. Result Analysis and Discussion

We have used reliable datasets from various sources; Bunny datasets, Dragon datasets, Laughing Buddha datasets have been collected from the Stanford University Graphics Lab., Horse dataset has been collected from Georgia Tech University CS website. Following diagrams are mesh reconstruction of the collected datasets based on the proposed work scheme.

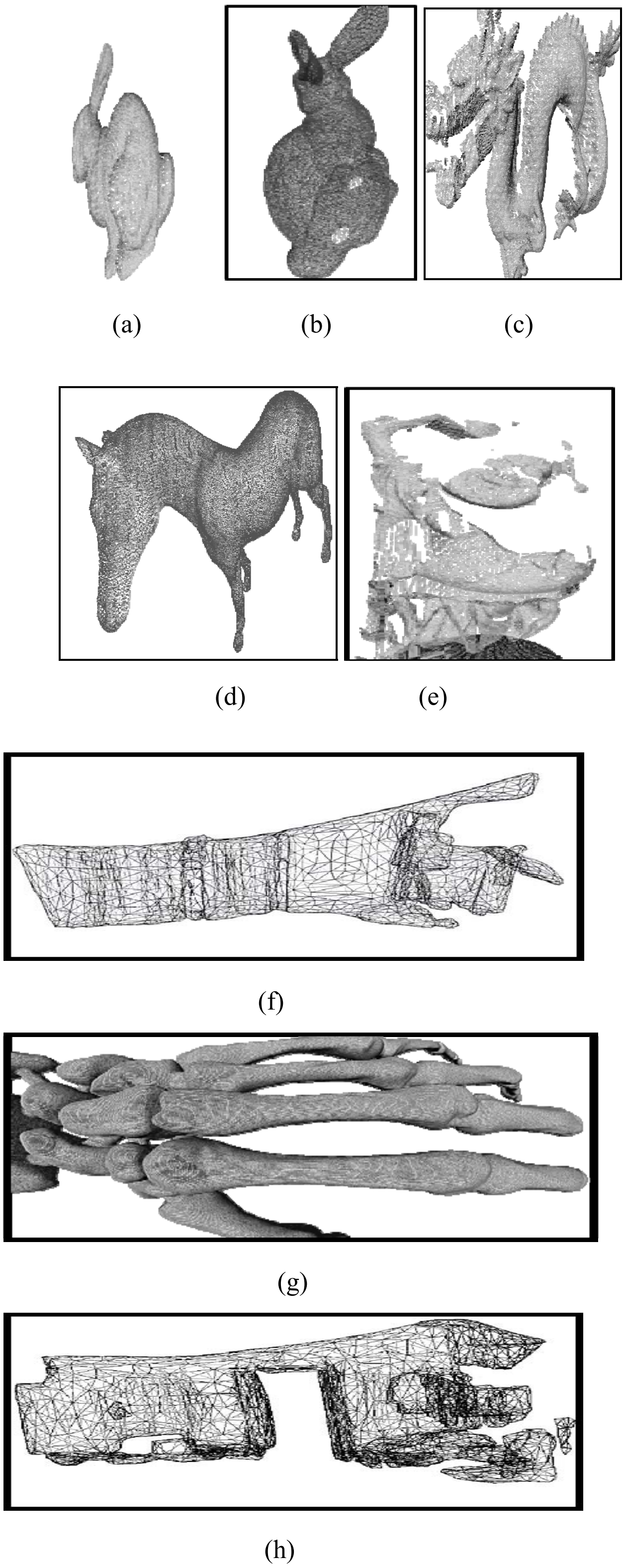

Fig. 2. a) Bunny back view set 1, b)Bunny back view set 2,c) Dragon set 1, d) Horse set 1,e) Laughing Buddha set 3, f) boundary wall, g) Hand bone, h) Door wall reconstructed by proposed study.

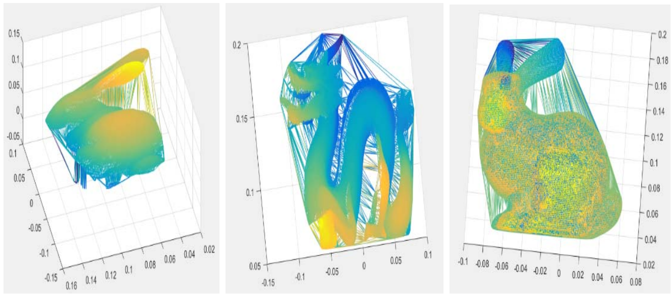

Fig. 3. Similar dataset is being triangulated with traditional triangulation algorithm.

From comparative perspective, several decisions can be made. From both of the figure we can see the graphical quality of the traditional algorithm. Also, we can see the presence of numerous amounts of extra triangulations have been done because of less efficient computation of convex hull. Also, it can be taken measure that proposed work is not absolute to provide perfect triangulation; some irregularities have been presented in the output due to lack of proper scanning and shading complexity. Our aim is to provide better simulated workflow in future by upgrading this scheme. Also for maintaining a smoothed output we are planning to do mesh smoothing on this work. Without any doubt this extra triangulation causes both extra time consumption and visual degradation. All of the tests have been performed in standard CPU.

Following figure holds the information about triangulation time for each of the tested datasets and a comparative analysis of timing information between traditional triangulation algorithm and proposed algorithm.

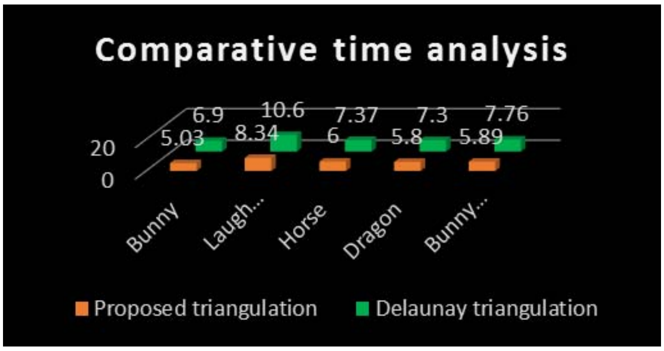

Fig. 4. Time comparison between the proposed work scheme and generic Delaunay triangulation

Since this works deals with the reconstruction of meshes, analysis of vertices and faces are necessary since a polygon mesh is a defined collection of vertices and faces. In this analysis we will show degree of accuracy about collecting information. Then we have compared this information with the original data and popular Delaunay algorithm as following in the upcoming picture.

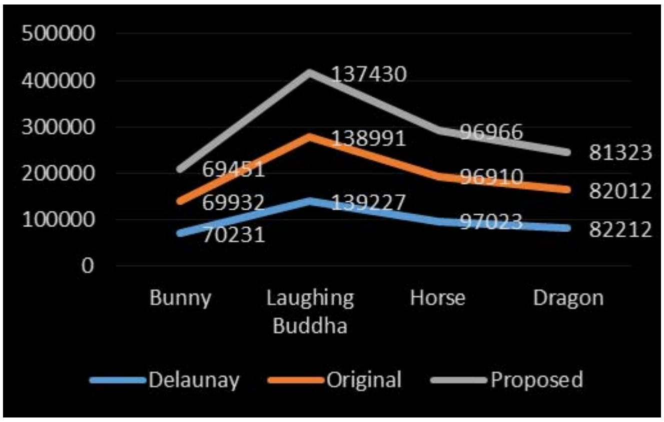

Fig. 5. Faces collected throughout the whole process.

From the following figure in the above, we can easily determine the difference between proposed algorithm and comparing algorithm. Here, we can see the number of faces provided by the existing Delaunay algorithm is slightly greater than the original data and number of data provided by the proposed scheme is little less than the original data; since the primary purpose is creating minimal amount of triangulation. Now we will see another graph about the number of vertices and without any confusion we can say that, proposed algorithm has provided similar type of performance with insignificant error.

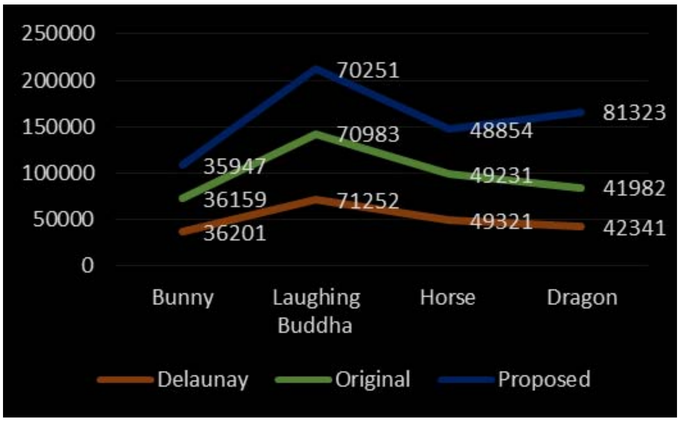

Fig. 6. Vertices collected throughout the whole process.

IV. CONCLUSIONS

For faster reconstruction of the three-dimensional scanning data, a modified algorithm has been proposed. According to the study input data set has been divided into number of samples in order to find the boundary points and provides a smooth calculation of convex hull. Then triangulation has been started through the points in a permutation manner. The legalization of proper triangulation been regulated by several constraints. During the iteration steps, necessary edge flipping has been done to compute the legit edges and properly reconstruct the legal triangle. The future work of this study is to compute the smoother meshes by performing the Laplacian smoothing for a better three-dimensional modelling system.

REFERENCES

[1] Jean-Daniel Boissonnat and Steve Oudot, “Provably good sampling and meshing of surfaces,” Graph. Models, vol. 67, no. 5, pp. 405–451

[2] William E. Lorensen and Harvey E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” ACM SIGGRAPH, vol. 21.

[3] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe,“Poisson surface reconstruction,” in Proc. of Symp. On Geometry Processing,

[4] IMAGE UNPROJECTION FOR 3D SURFACE RECONSTRUCTION: A TRIANGULATION-BASED APPROACH’ Min-H. Sung, Hwasup Lim, Hyoung-Gon Kim and Sang Chul.

[5] Atsutada Nakatuji, Yasuyuki Sugaya, and Kenichi Kanatani, “Mesh optimization using an inconsistency detection template,” in IEEE ICCV, 2005, pp. 1148–1153.

[6] .Quiet Area Detection in 3D Sound Field Simulation via Delaunay Triangulation ;Xiaoyan Liang1, Zhe Fan1, Ge Lin1,2,∗, Xiaonan Luo1,2014 5th International Conference on Digital Home.

[7] Maur P. Delaunay triangulation in 3D[R]. Technical Report, Departmen.of Computer Science and Engineering, University of West Bohemia, Pilsen, Czech Republic, 2002.

[8] Flick U. Mixing Methods, Triangulation, and Integrated Research[J]. Qualitative Inquiry and Global Crises, 2011: 132.

[9] Biniaz A, Dastghaibyfard G. A faster circle-sweep Delaunay triangulation algorithm [J]. Advances in Engineering Software, 2012, 43(1): 1-13.

[10] Vieira M, Shimada K. Surface mesh segmentation and smooth surface extraction through region growing[J]. Computer aided geometric design, 2005, 22(8): 771-792.

[11] New Triangulation Method for Surface Scattered Points; Xuehe Zhang , Ge Li and Jie Zhao Zhenxiu Hou;roceedings of 2014 IEEE International Conference on Mechatronics and Automation August 3 - 6, Tianjin, China

[12] Computational Geometry Algorithms and Applications; Mark de Berg • Otfried Cheong Marc van Kreveld • Mark Overmars; Springer 3rd edition,2014.