3D Photography Project

An Overview of 3D Photography and
The Computational Aspect of Triangle Mesh

By Muhammed Santa for 3D photography class, Prof. Ioannis Stamos, CUNY,Fall2003

Project Overview

In this project, although I concentrated on the computational geometry that involves the triangulation process, following topics are covered :

❖ 3D Photography overview :

3D photography is the process of using cameras and light to capture the shape and appearance of real objects. This process provides a simple way of acquiring 3D models of unparalleled detail and realism by scanning them in from the real world.

A variety of techniques are used to achieve 3D models through 3DP, but they all share some common steps of operation.I will give a brief overview of these steps involving 3D Photography.

❖ Dalaunay triangulation and Voronoi diagram :

Process of creating triangles from camera coordinates c(x) and c(y) of the given data sets is known as triangulation which is needed to determine the 3D coordinates of the point sets is part of meshing process. We do not need to know the projector coordinates p(x) and p(y) for triangulation but p(x) and p(y) are needed along with intrinsic parameters c,f,k of the device to determine the 3D coordinates.One common method used for triangulation is **Dalaunay Triangulation. A Voronoi diagram** is a dual of dalaunay triangles can be used in mesh generation**.** I will discuss the algorithm and computational complexity of Dalaunay triangulation and Voronoi diagram. I will also show example of some other triangulation methods.

❖ Results from triangulations and meshes:

some triangle meshes, dalaunay triangulations and voronoi diagrams are created from data sets.I have not implemented any of the methods yet, so used matlab functions to generate results. C code to read .ptx files and matlab codes are shown.

3D Photography overview

For all image base models, idea for achieving 3DP actually is recovering the shape of the objects, i.e. the 3d coordinates of the object and then reconstructing the object from those 3d data set.

Image formation process consists of a projection from a three-dimensional scene onto a two dimensional image. During this process the depth is lost. The three-dimensional point corresponding to a specific image point is constraint to be on the associated line of sight. From a single image it is not possible to determine which point of this line corresponds to the image point. If two (or more) images are available, then the three-dimensional point can be obtained as the intersection of the two line of sights. This process is called triangulation.

In 3DP usually three types of models are chosen:

- **Surface model:** The 3D surface is approximated by a triangular mesh
- **Volumetric model:** Use voxel depth map to create a surface.
- **Plenoptic model:** New (virtual) views are rendered from the recorded ones by interpolation in real time from internal image geometry.

3DP technique for all these models usually involve following steps:

- **3D Data Collaboration:** Calibration process, sampling etc.
- **3D Shape Recovery:** Coordinate recovery, registration etc.
- **3D Image Reconstruction:** triangulation, improved meshing etc.
- **3D model acquisition:** Rendering with texture map.

3D Photography overview

❖ 3D Data Collaboration:

3D Data: Three types of data usually considered: point data, volumetric data and surface data.Point data is simply point clouds offer no connectivity information.

Volumetric data(Voxels) are popular in medical imaging can see 'insight' of an object. Surface data types include polyhedral and usually correspondence to what we see., the most popular one is triangle.Range images offer depth information along with 3D coordinates and intensity.

Calibration Process: In order to know the relationship between an image point and its line of sight, we need to find following geometric and radiometric data.

- **Intrinsic:**Focal length, principal points, distortion.
- **Extrinsic:**Positions,orientation.
- **Radiometric**: mapping between pixel value and scene radiance,gamma values etc.

Targets in calibration:with full 3D(non planar) calibration target we can calibrate with one picture, but difficult to construct. With 2D(planar) calibration target we need multiple views but can be constructed accurately.

Some of the commonly used targets are: checker board,concentric coded circles, coded circles, 3D circles, color coded circles.

Sampling/Scanning :Good sampling rate(not necessarily high) and sampling from all possible directions are needed to acquire whole model and to minimize anti-aliasing.Both coherent(a point cloud) and incoherent(showing contour lines) scanning can be used.MCOP images and multiple view projections ensure good sampling.

3D Photography overview

- **3D shape recovery:**

  3d Coordinate Recovery: usually involves internal image geometry and camera information.Camera coordinates c(x) and c(y),projector coordinates p(x) and p(y) are needed along with intrinsic parameters c,f,k of the device to determine the 3D coordinates. This process also could be classified as 2D to 3D registration,may also includes classification and segmentation process.

  Global Registration: needs due to alignment process for multiple scans of the object.Error due to alignment should be distributed to all scans.Most popular techniques are pairwise alignment, pairwise ICPs.

  3D Shape Recovery from point correspondence : To find feature in one image, we search along the epipolar line of the corresponding image. Stereo images causes small baseline and ambiguity.Multiperspective images reduces ambiguity.Shapes from motion, offers easier feature tracking for far away views.Shapes from shading are created based on the surface reflectance values under a known point light could be achieved with one image but mathematically unstable.

3D Photography overview

❖ Image Reconstruction:

Splatting: Usually means to reproject one input pixel onto multiple output pixels.Consists of rendering each point using reconstruction kernel.

Meshing: Once we have a cloud of 3D points which have to be connected somehow in order to look like a surface. This is known as mesh creation. Process of reconstructing object surface by partitioning into smaller domain (polygon, polyhedron etc) might includes some of the following steps:

- **Triangulation:** A triangulated planar surface is needed for smooth patching, probably the first step of meshing. Usually uses Delaunay triangulation algorithm.
- **Volume Carving**: 3D isosurface representation technique to reconstruct 3D area with right resolution from volumetric data. Marching cube(for 3D data) or marching square (for 2D data)algorithms are widely used.
- **Polygon meshing/ mesh optimization**: can use triangulation or dividing into some higher-order polygons or polyhedrol to reconstruct object surface from 3d data.Catmull-clarkalgorithm is widely used in this purpose.
- **Surface Interpolation** : might needed to fill in the holes or to remove noises .Linear interpolation is used widely for isosurface generation and bilinear and trilinear interpolation could be used for better results. Delaunay triangulation or other surface interpolation methods like polynomial interpolation or spline interpolation widely used .

Smoothing/Unshading the mesh: Process is used to make smoothly joined patches might includes zippering process to put together the final object model from several data sets.some cases like for MCOP images zippering process might not be needed.

3D Photography overview

❖ 3D Model acquisition (last but not the least!) :

Usually **Image Base Rendering(IBR)** techniques are used in surface texture mapping.Unlike traditional 3D computer graphics in which 3D geometry of the scene is known,image-based-rendering(IBR) techniques render novel views directly from input image data.IBR can be classified in three categories:

Rendering with no geometry: techniques rely on characterization of the plenoptic function.might includes light field rendering or mosaicking.Light field rendering uses many images for filtering and interpolating (but, not to get geometric information) to generate new image of scene. Concentric mosaics reduces the data by using a circular camera path.

Rendering with implicit geometry:might includes lumigraph or view interpolation or view morphing. View interpolation generates novel views by interpolating optical flow between corresponding points.View morphing generates in-between camera matrices based on point correspondence of two original camera data.

Rendering with explicit geometry:might includes LDI,3D warping, view dependent texture mapping, texture map models. Layer Depth Images(LDI), 3D warping only uses depth information for a set of images for rendering.Multiple-Center-of-Projection (MCOP) images use one single image with internal epipolar geometry.

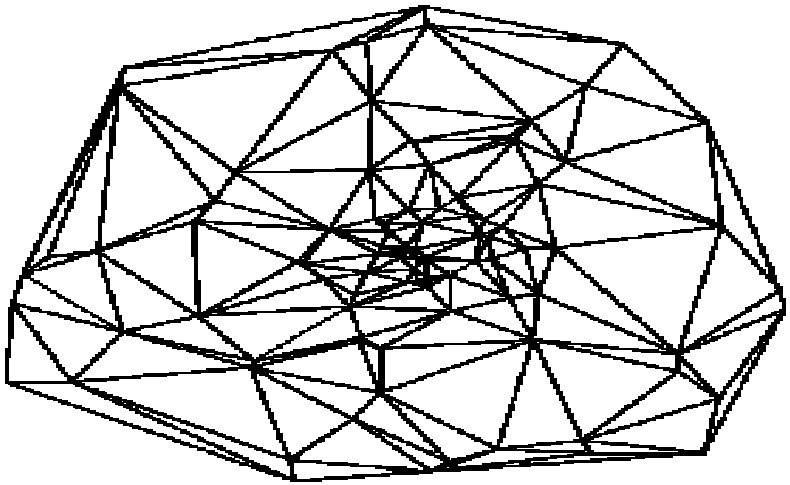
Other surface rendering techniques(Flat shadding, ray tracing etc.) and **Lighting techniques**(Diffuse, Speculer, Transparency etc.) is often used to improve rendering.

Delaunay Triangulation and Voronoi Diagram

Triangle meshing, a surface interpolation method uses triangulated surface as a means of reconstructing the surface from the Unstructured data sets. Most commonly uses algorithm to generate triangulated surface from an unstructured point set is Delaunay Triangulation algorithm.

Delaunay Triangulation:

The Delaunay triangulation of a point set is a collection of edges satisfying an "empty circle" property. A ***Delaunay*** *triangulation* of a vertex set is a triangulation of the vertex set with the property that no vertex in the vertex set falls in the interior of the circumcircle (circle that passes through all three vertices) of any triangle in the triangulation.



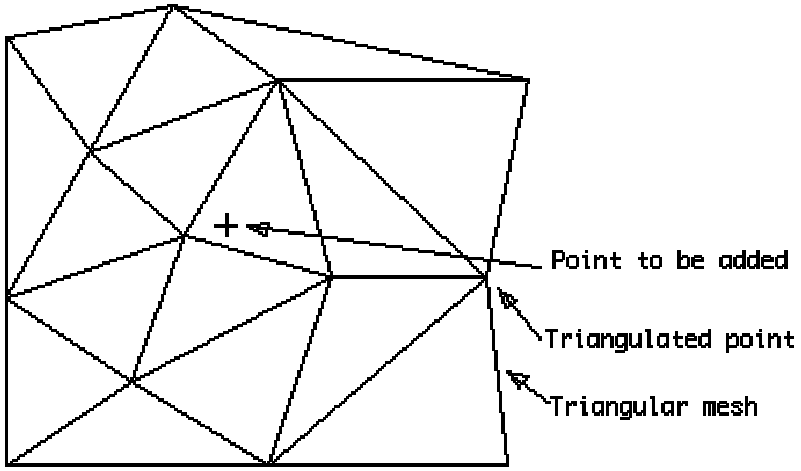
Delaunay Triangulation and Voronoi Diagram

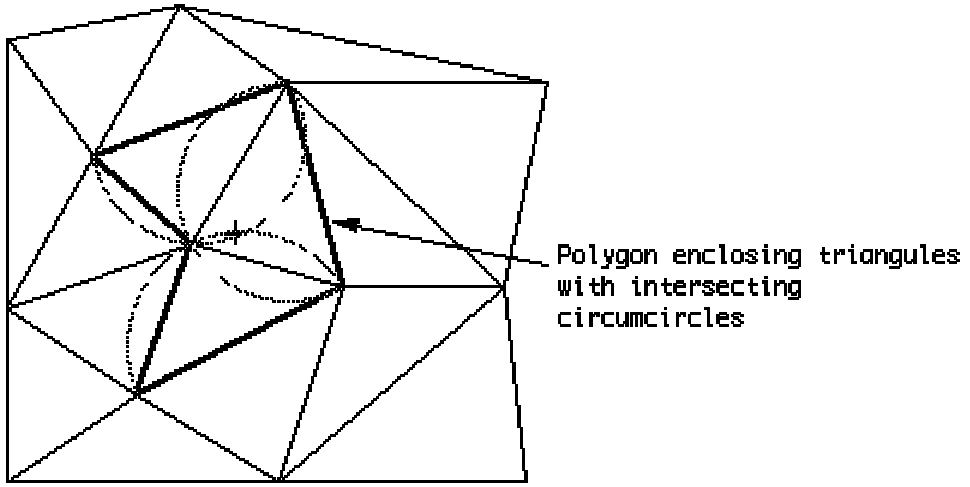
Algorithm Description Delaunay O(N^2) :

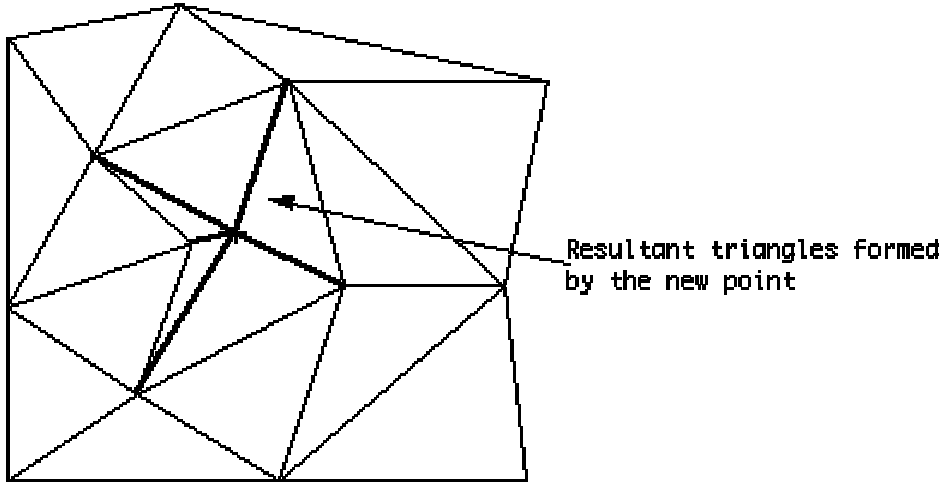
- Several other algorithms are available.
- Following algorithm was given by Paul Bourke runs in O(n^2) time, can be improved to O(n^1.5).

```
subroutine triangulate
input : vertex list output : triangle list
        initialize the triangle list
        determine the supertriangle
        add supertriangle vertices to the end of the vertex list
        add the supertriangle to the triangle list
        for each sample point in the vertex list initialize the edge buffer
                for each triangle currently in the triangle list
                calculate the triangle circumcircle center and radius
                if the point lies in the triangle circumcircle
                then add the three triangle edges to the edge buffer
                remove the triangle from the triangle list
                endif
                endfor
        delete all doubly specified edges from the edge buffer
        this leaves the edges of the enclosing polygon only
        add to the triangle list all triangles formed between the point
         and the edges of the enclosing polygon
         endfor
        remove any triangles from the triangle list that use the supertriangle vertices
        remove the supertriangle vertices from the vertex list
end
```

Delaunay Triangulation and Voronoi Diagram

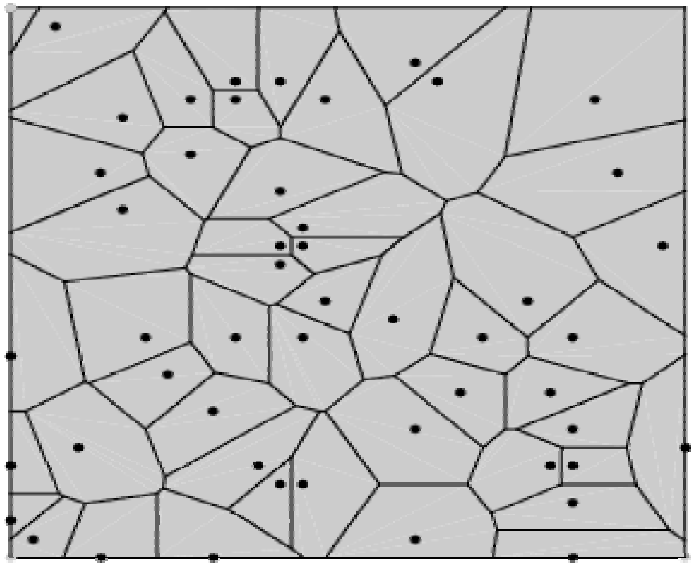






Delaunay Triangulation and Voronoi Diagram

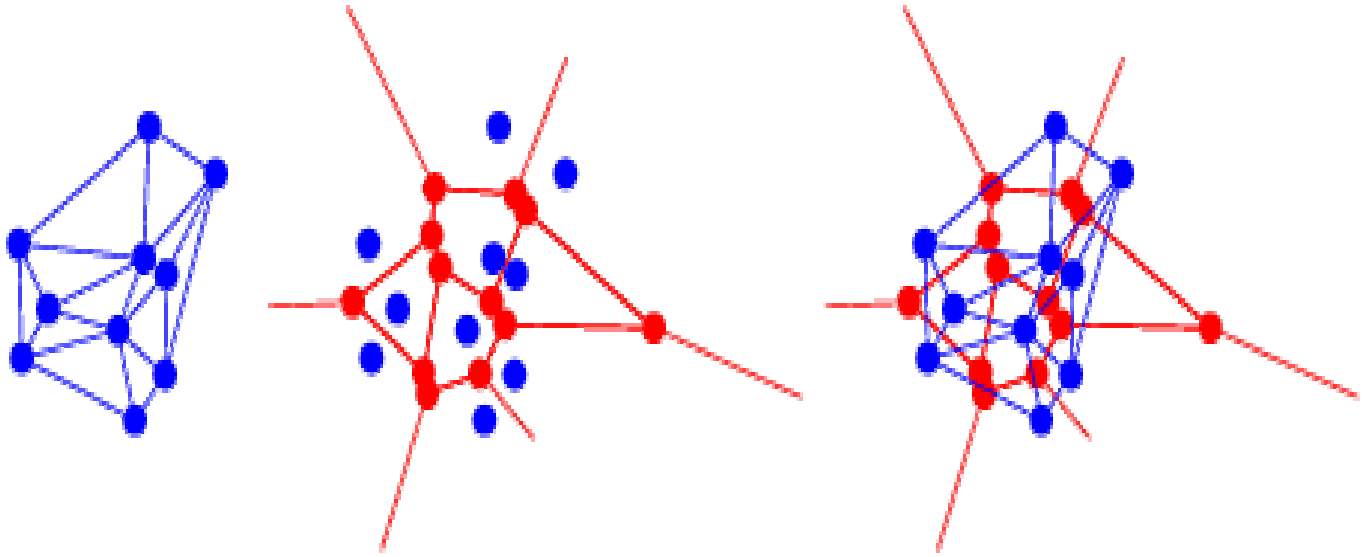
- A *Voronoi diagram* of a vertex set is a subdivision of the plane into polygonal regions (some of which may be infinite), where each region is the set of points in the plane that are closer to some input vertex than to any other input vertex. (The Voronoi diagram is the geometric dual of the Delaunay triangulation.)



Delaunay Triangulation and Voronoi Diagram

- **Voronoi Diagram algorithm:** Since the Delaunay Triangulation is in hand, it is easy to compute the Voronoi diagram, accomplished in O(N log(N)).
- **Pseudo code :** while (delaunay triangles){
  compute center **px,py** and radius **rad** of incoming delaunay triangle
  A = bx - ax,
  B = by - ay,
  C = cx - ax,
  D = cy - ay,
  E = A(ax + bx) + B(ay + by)
  F = C(ax + cx) + D(ay + cy)
  G = 2(A(cy - by) - B(cx - bx)),
  px = (DE - BF)/G,
  py = (AF - CE)/G.
  rad = dist((px,py) to A or B or C)
  }
  check to see if each of the edges are already added to data structures;
  if(already added)
  add the voronoi center of this triangle
  else
  create a new voronoi Node and add the information display the vornoi edges, vertices, and circles
  end

Delaunay Triangulation and Voronoi Diagram



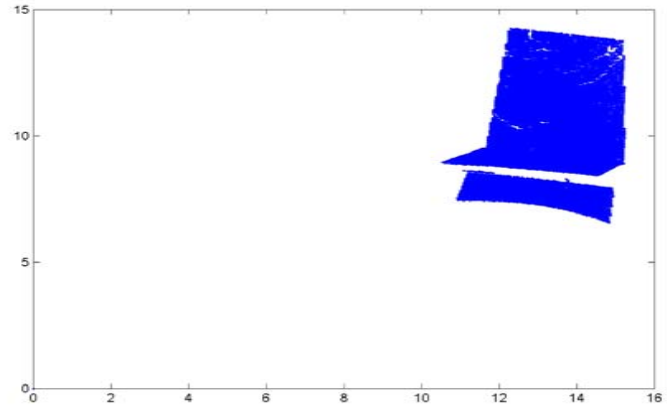
Delaunay triangulation

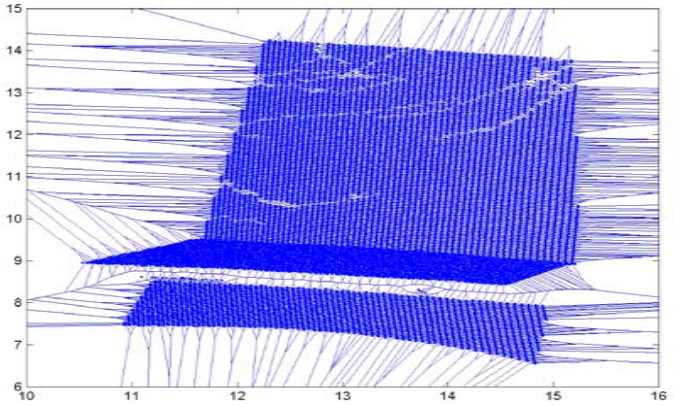
Voronoi diagram

Delaunay and Voronoi

Results from triangulations and meshes:

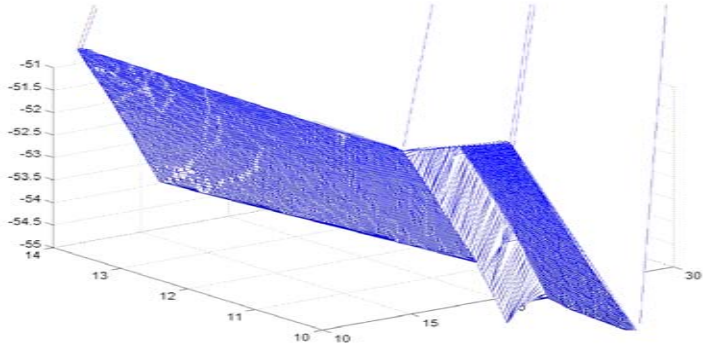
- Results from small_example.ptx data
- Top(plot from small_example.ptx)
- Bottom(Vornoi diagram/triangulation)

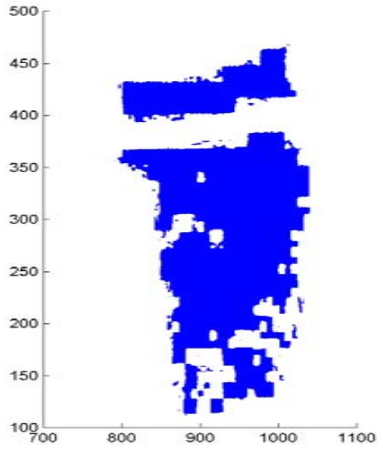


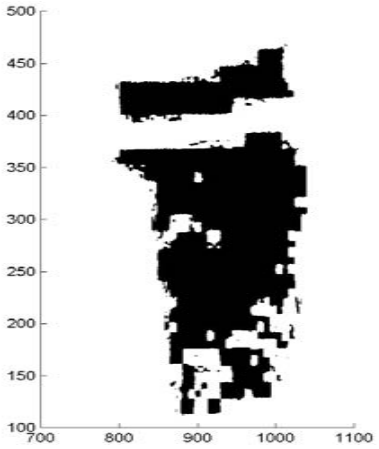


Results from triangulations and meshes:

- Results from small_example.ptx data
- Top(mesh generated from small_example.ptx)
- Bottom(data after smoothing mesh)

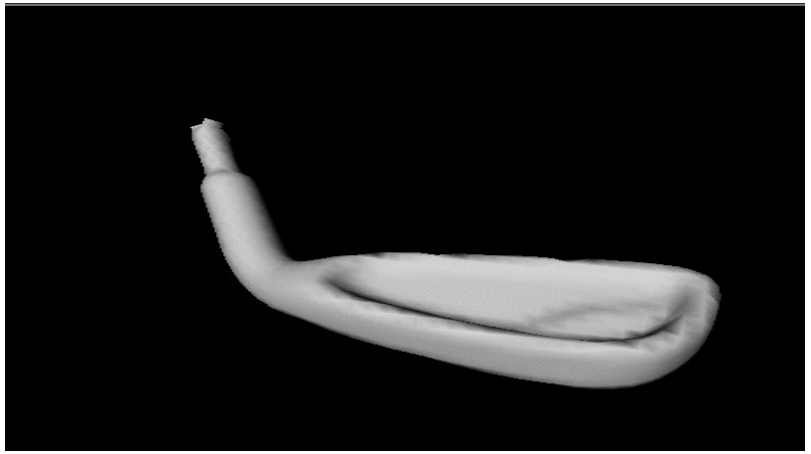


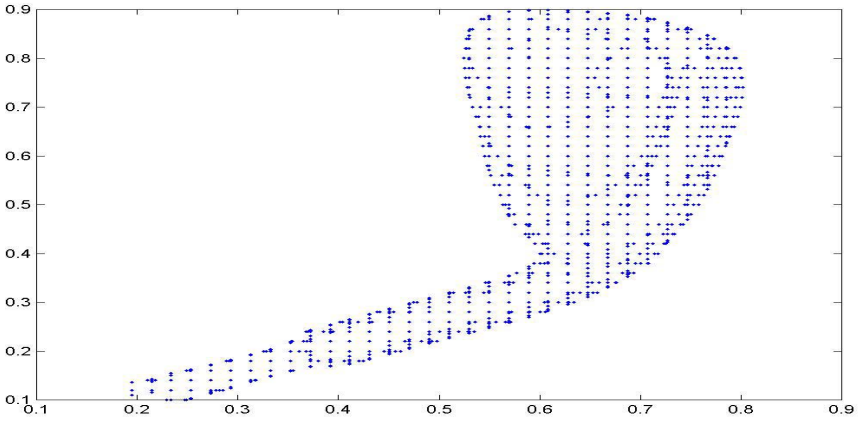




Results from triangulations and meshes:

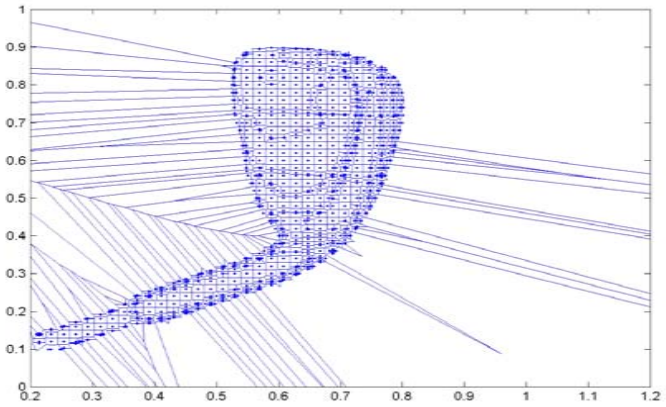
- Results from club.ptx data
- Top(club image)
- Bottom(data plotted from club image)

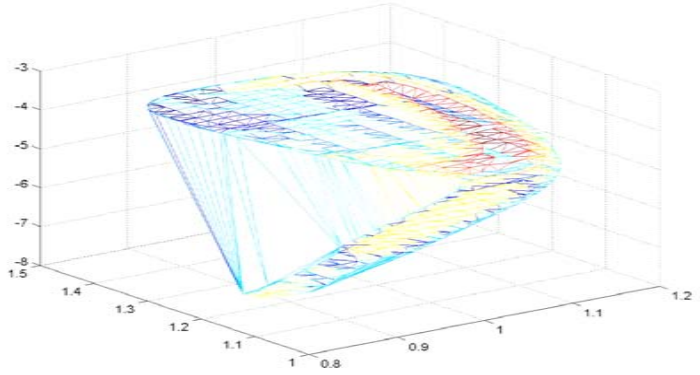




Results from triangulations and meshes:

- Results from club.ptx data
- Top(vornoi triangulation for club)
- Bottom (Trimash for club)





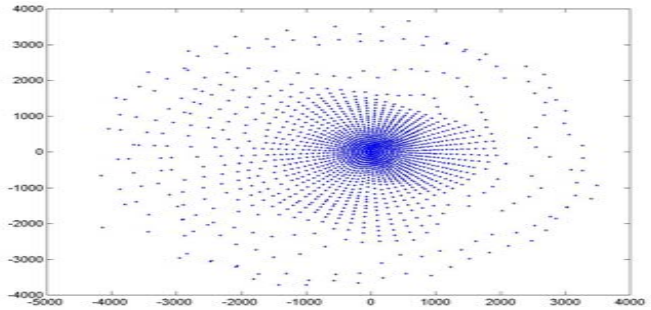
Results from triangulations and meshes:

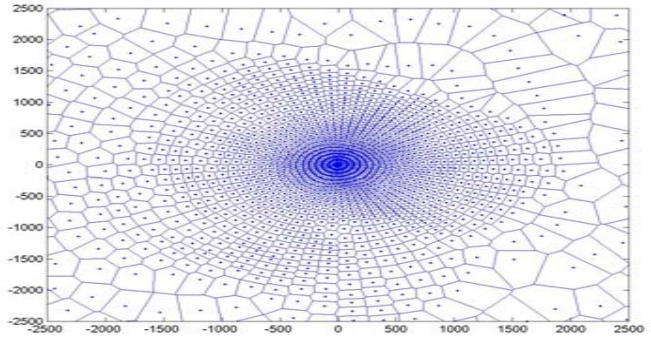
Results from xray.xyz data;

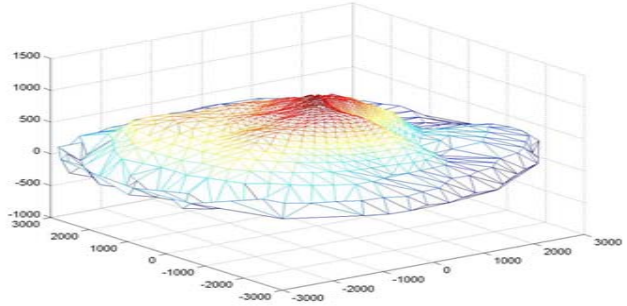
Top (xray points)

Middle(vornoi diagram)

Bottom(trimesh generated)







Matlab codes:

- Matlab code for small_data(range data) example:
- v1=load('c:\3dPhoto\small_example.xyz');
- xdata=v1(:,1); ydata=v1(:,2); zdata=v1(:,3);
- plot(xdata,ydata,'.');
- tri=delaunay(xdata,ydata);
- trimesh(tri,xdata,ydata,zdata);axis([10 30 10 14 -55 -51]);
- voronoi(xdata,ydata,tri);axis([10 16 6 15]);
- [x,map]=imread('c:\3dPhoto\small_example.jpeg');
- xsmooth=smooth3(x,'box',3);
- view(3); subplot(1,2,2)
- p=patch(isosurface(xsmooth,.5));
- subplot(1,2,1)
- p=patch(isosurface(xsmooth,.5),'FaceColor','Blue','EdgeColor','none');
- reducepatch(p,.15); rotate3d on;

Matlab codes:

- mash/triangulation (from club.pts)
- Matlab code to generate triangulation and mesh
- v1=load('c:\3dPhoto\club.xyz');
- xdata=v1(:,1);
- ydata=v1(:,2);
- zdata=v1(:,3);
- plot(xdata,ydata,'.');
- tri=delaunay(xdata,ydata);
- trimesh(tri,xdata,ydata,zdata);axis([.8 1.2 1 1.5 -8 -3]);
- voronoi(xdata,ydata,tri);axis([.2 1.2 0 1.0]);
- [x,map]=imread('c:\3dPhoto\club_trimesh.jpeg');
- xsmooth=smooth3(x,'box',3);
- view(3);
- subplot(1,2,1)

C codes:

- Here is a code in C that reads a ptx file.
- typedef struct {
  int width,height,ptCount;
  float *ptxX, *ptxY, *ptxZ, *ptxR;
 }pointCloudS, *pointCloudP;

```
int loadPtxFile(char *filename, pointCloudP ptCloud)
{FILE *fp;
 int n, i, width, height, count;
 fp=fopen(filename, "r");
 if(!fp) {UI_printf("Error: couldn't open %s.\n",filename);
  return 0;}
 ptCloud->ptCount=0;
 fscanf(fp,"%d\n%d",&width, &height);
 n = width * height;
 ptCloud->width = width;
 ptCloud->height = height;
 ptCloud->ptxX = (float*)malloc(sizeof(float)*n);
 ptCloud->ptxY = (float*)malloc(sizeof(float)*n);
 ptCloud->ptxZ = (float*)malloc(sizeof(float)*n);
 ptCloud->ptxR = (float*)malloc(sizeof(float)*n);
 count=0;
 while(!feof(fp) && count < n) {
  fscanf(fp,"%f %f %f %f", &(ptCloud->ptxX[count]), &(ptCloud->ptxY[count]), &(ptCloud->ptxZ[count]),
   &(ptCloud->ptxR[count])); count++;}
 ptCloud->ptCount = count;
fclose(fp);
 return 1;}
```

conclusion

- 3D photography is an emerging field which uses lot of application from image processing, computer graphics, computer vision and computational geometry.

- Triangulation is a good and simple surface interpolation method in 3DP although higher order polygons or polyhedrons are usually used in practice for fewer surface divisions.