In many algorithms we try to triangulate the surface which are not realizable in R3 plain. Klein bottle and the real projective plain are two examples. Some algorithm tries to generalize data in high dimensional using **Delaunay Triangulation** theorem but its worst case complexity is in exponential. This makes its limited to only low dimensions.

**Freeman incremental algorithm** is only algorithm which can triangulate surface from high dimensional point cloud.

Most algorithms are designed to reconstruct manifold surfaces and those who reconstruct non-manifold surfaces, they only take point cloud in R3.

This algorithm was largely motivated by our desire to construct a surface from a point

cloud dataset describing molecular conformations of cyclo-octane, known from first principles to have two degrees of freedom. Of course the algorithm is not restricted for use on molecular conformation data.

**Algorithm**

This algorithm can reconstruct non-manifold and manifold surfaces.

For manifold surfaces it is a slight modification of Freeman incremental algorithm. This algorithm uses MDS ( Multi- Dimensional Scaling ) for neighbourhood distance. It doesn’t use angle and tangent space parameters like Freemans algorithm

In case of Non-Manifold surfaces, it uses pre-processing before applying Triangulation method.

In the first pre-processing step, it identify the non-manifold regions by local dimension estimation. The manifold regions of the surface are assumed to the two-dimensional (2D) and the non-manifold regions are assumed to be three dimensional(3D). The 3D regions are then fitted locally with two intersecting planes using a degenerate quadratic surface.Using the two planes, it splits the 3D neighborhood into two 2D neighborhoods.

**Finding Non-Manifold Regions In Point Clouds**

It finds the region by portioning the data sets by local dimension and for estimating local deminsion it uses PCA **( Principal Component Analysis** ) at each point.

(<https://medium.com/@aptrishu/understanding-principle-component-analysis-e32be0253ef0>)

(<http://www.statisticssolutions.com/principal-component-analysis-pca/>)

Let data set X= {xi} and and E (epslon ) as neighbour of xi by set Bi ={ xj: |xj-xi|<E}

To perform the PCA it do following steps:

Xm = mean of the points belong to Bi.

Bm = Bi-Xm.

We use SVD on Bm to find = U *Σ* V^T where *Σ* is a diagonal matrix containing the singular values *σ j* and those singular values gives idea about local dimension. ( In my downloads )

**Fitting Plane In Data Set**

Let our surface is found to be non-manifold ( 3D) and it has double curve i.e. it has self intersection where intersection points belong to two tangent spaces.

Formally, the singular-value decomposition of an {\displaystyle m\times n} real or complex matrix {\displaystyle \mathbf {M} } is a factorization of the form {\displaystyle \mathbf {U\Sigma V^{\*}} }, where {\displaystyle \mathbf {U} }is an {\displaystyle m\times m} real or complex [unitary matrix](https://en.wikipedia.org/wiki/Unitary_matrix), {\displaystyle \mathbf {\Sigma } } is a {\displaystyle m\times n} [rectangular diagonal matrix](https://en.wikipedia.org/wiki/Rectangular_diagonal_matrix) with non-negative real numbers on the diagonal, and {\displaystyle \mathbf {V} } is an {\displaystyle n\times n} real or complex unitary matrix. The diagonal entries {\displaystyle \sigma \_{i}} of {\displaystyle \mathbf {\Sigma } } are known as the [**singular values**](https://en.wikipedia.org/wiki/Singular_value) of {\displaystyle \mathbf {M} }. The columns of {\displaystyle \mathbf {U} } and the columns of {\displaystyle \mathbf {V} } are called the **left-singular vectors** and **right-singular vectors** of {\displaystyle \mathbf {M} }, respectively.

The singular-value decomposition can be computed using the following observations:

* The left-singular vectors of **M** are a set of [orthonormal](https://en.wikipedia.org/wiki/Orthonormal) [eigenvectors](https://en.wikipedia.org/wiki/Eigenvectors) of **MM**∗.
* The right-singular vectors of **M** are a set of orthonormal eigenvectors of **M**∗**M**.
* The non-zero singular values of **M** (found on the diagonal entries of **Σ**) are the square roots of the non-zero [eigenvalues](https://en.wikipedia.org/wiki/Eigenvalues) of both **M**∗**M** and **MM**∗.

**Pre-Processing**

After fitting planes ( neighbourhood) into dataset means we have now a manifold surface ( 2D) and we have obtained fitted neighbourhood. After fitting Bi , split Bi into three subset :

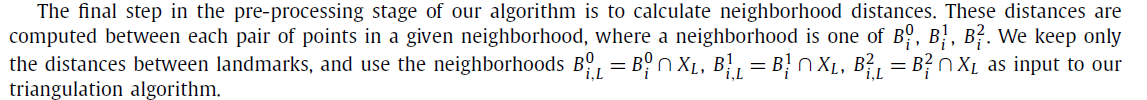
* One contains the points new the intersection (Bi0)
* Bi1 contains the points closest to one plane
* Bi2 contains the points closest to other plane

For pre processing we subsample a set from dataset known as landmark data set. We first take points from the intersection and then from rest of the set. Subsampling imrproves the quality of the theorem.

**Condition of XL** Each point in Xl must be at distance dl (Euclidian ) from other point in Xl

**Final** step in pre processing is to calculate the neighbourhood distance.

**Neighbourhood Distance:** It is the distance between the neighbourhood points which can from one of Bi0 Bi1 and Bi2.



So pair wise distance and landmark neighbourhood are input to triangulation theorem. But the triangulation cannot be tuned except by altering the various pre-processing parameters. Before describing the triangulation algorithm.

**Reconstruction**

