Recap

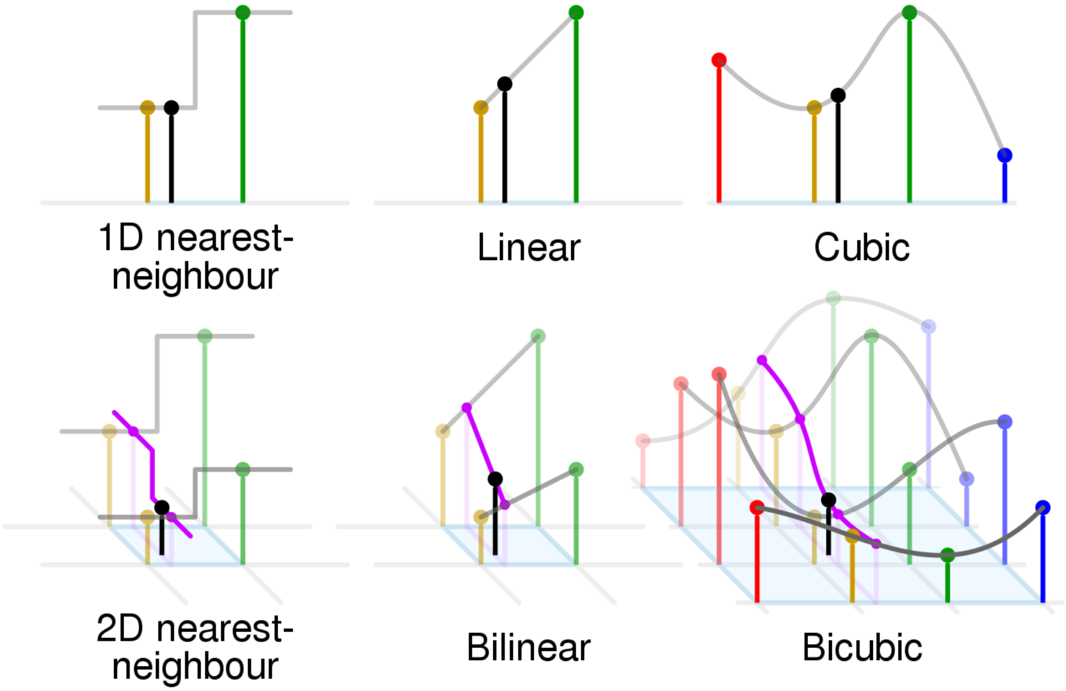
- UAS-LiDAR system and data collection
  - Velodyne-16 LiDAR
  - GNSS & IMU
  - LiDAR file format
- A brief overview of LiDAR data preprocessing steps
- Algorithm details
  - noise/outlier removal
  - ground filtering
  - rasterization

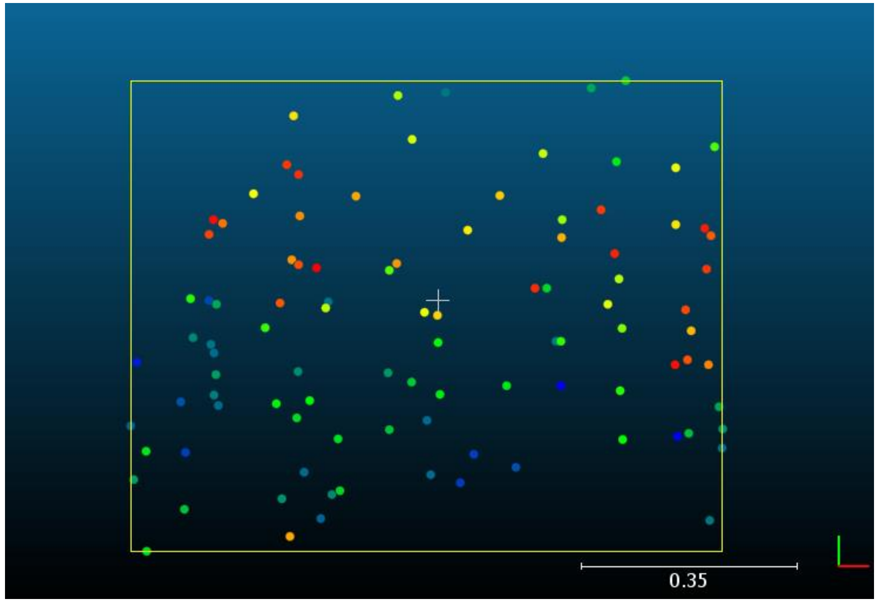
RIT
College of Science
Chester F. Carlson Center for Imaging Science

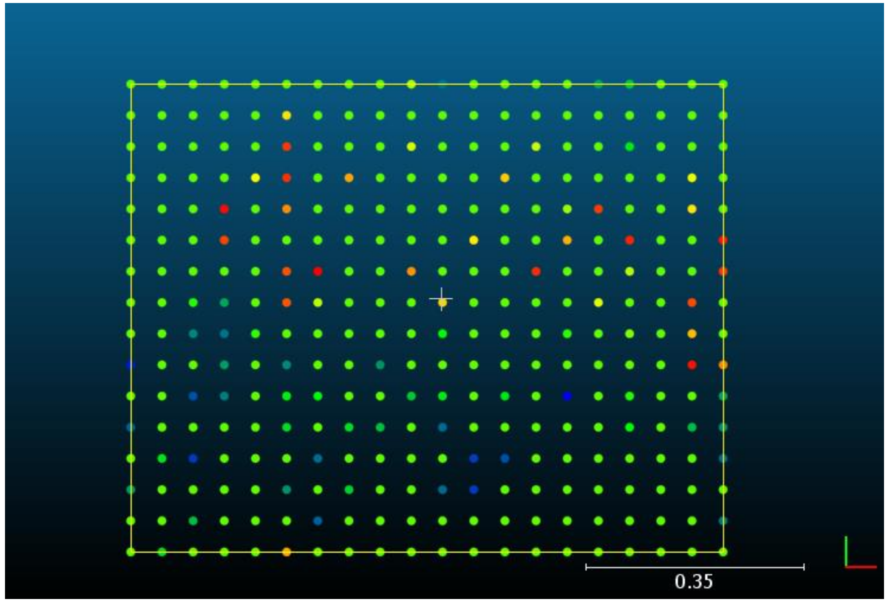
Outline – week 13

1. Interpolation methods
2. Down-sampling methods
3. PCA for point cloud
4. Surface normal
5. Iterative Closest Points (ICP) registration

Interpolation methods

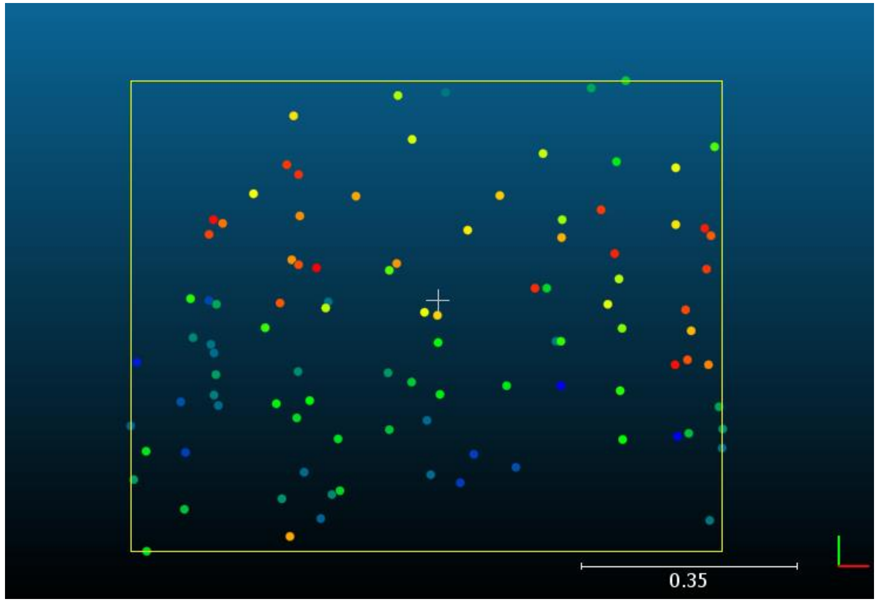




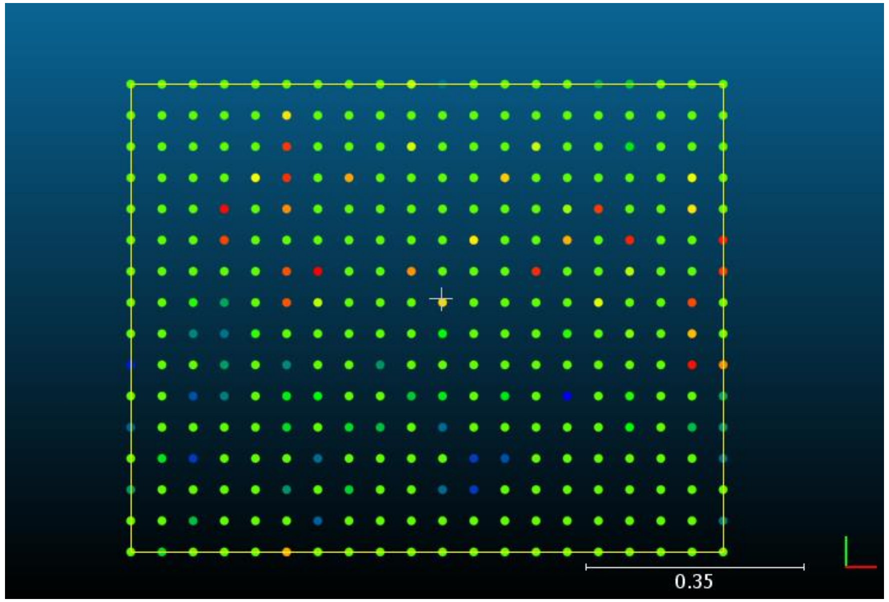


ref: https://en.wikipedia.org/wiki/Multivariate_interpolation

Interpolation methods



- **Inverse Distance to a Power** when you do not specify a smoothing factor
- **Kriging** when you do not specify a nugget effect
- **Nearest Neighbor** under all circumstances
- **Radial Basis Function** when you do not specify an R^2 value
- **Modified Shepard's Method** when you do not specify a smoothing factor
- **Triangulation with Linear Interpolation**
- **Natural Neighbor**



Surfer 15

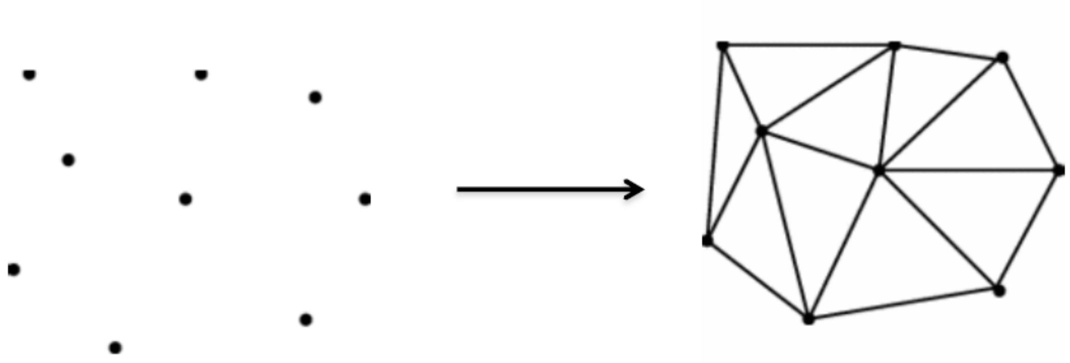
ref: Surfer15 Full User’s Guide

RIT
College of Science
Chester F. Carlson Center for Imaging Science

Interpolation methods

There are ***three common methods*** used for interpolation:

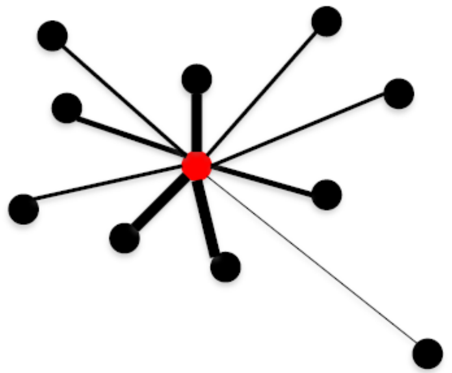
1. **Triangulation with interpolation**



Each triangle is considered to be a plane (*TIN* = triangulated irregular network)

Interpolation is used to find values in the triangle

2. **Inverse distance weighting**



- Each point in the grid is a weighted average of the points
- The weight is set to 1 / D, where D is the distance between the point on the grid and the lidar return

Modified from Jan's slides

Interpolation methods - IDW

2. Inverse distance weighting

-- Weighted Averaging

Assumes value of an attribute *z* at any unsampled point is a distance-weighted average of sampled points lying within a defined neighborhood around that unsampled point. Essentially it is a weighted moving average

$$\hat{z}(x_0) = \sum_{i=1}^{n} \lambda_i \cdot z(x_i) \qquad \sum_{i=1}^{n} \lambda_i = 1$$

where λ_i are given by some weighting function

Modified from Jan's slides

Interpolation methods - IDW

Inverse Distance to a Power Math

The equation used for Inverse Distance to a Power is:

$$\hat{Z}_j = \frac{\sum_{i=1}^{n} \frac{Z_i}{h_{ij}^{\beta}}}{\sum_{i=1}^{n} \frac{1}{h_{ij}^{\beta}}}$$

$$h_{ij} = \sqrt{d_{ij}^2 + \delta^2}$$

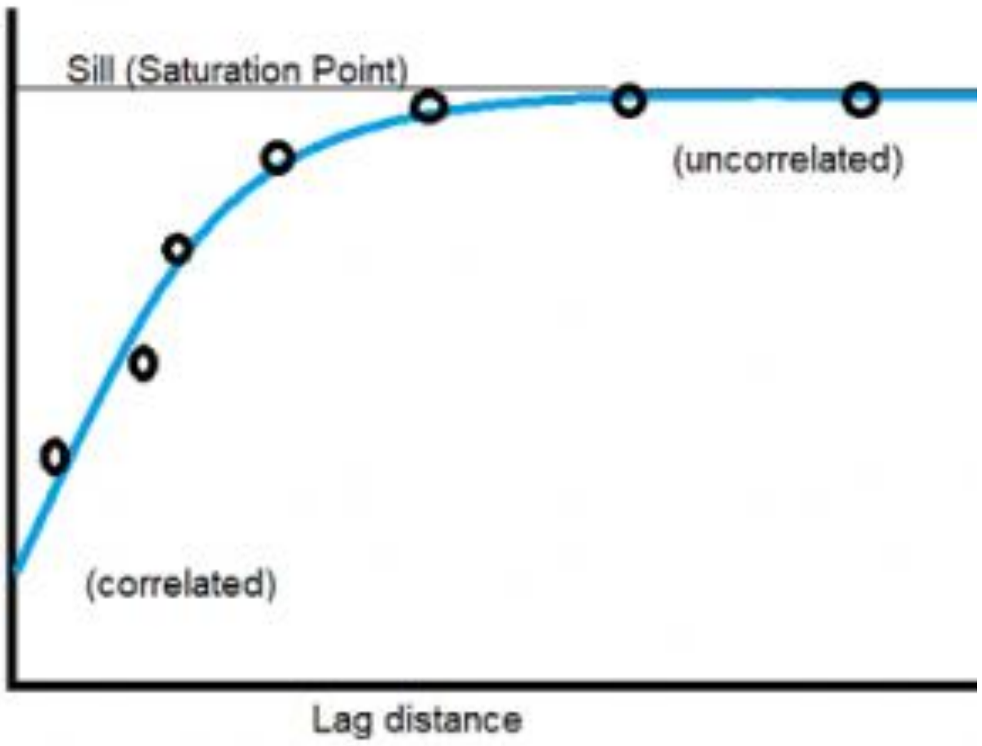
where:

h_{ij}	is the effective separation distance between grid node "j" and the neighboring point "i."
$\hat{Z}_j$	is the interpolated value for grid node "j";
Z_i	are the neighboring points;
d_{ij}	is the distance between the grid node "j" and the neighboring point "i";
β	is the weighting power (the *Power* parameter); and
δ	is the *Smoothing* parameter.

ref: Surfer15 Full User’s Guide

Interpolation methods - Kriging

3. Kriging - Variogram



src: https://www.statisticshowto.com/variogram/

Interpolation methods - Kriging

Prediction with Kriging

- Developed variogram used to estimate distance weights for interpolation
  - Interpolated values are the sum of the weighted values of some number of known points where weights depend on the distance between the interpolated and known points
- Weights selected so that estimates are:
  - Unbiased
  - Minimum variance
  - Selected based on the theoretical semi-variogram you choose

$$\hat{Z}(s_o) = \sum_{i=1}^{N} \lambda_i Z(s_i)$$

$where:$

$Z(s_i) = measured\_value\_at\_i$

$\lambda_i = unknown\_weight\_at\_i$

$s_o = prediction\_location$

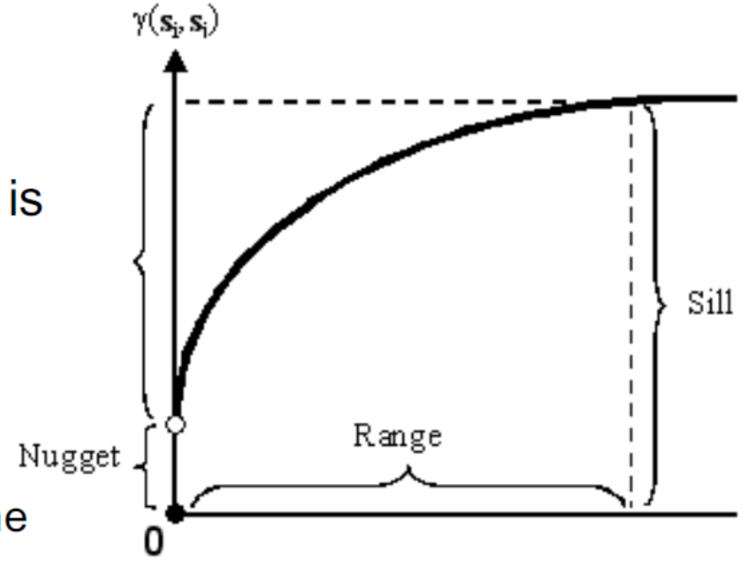
$N = number\_of\_measured\_values$

Modified from Jan's slides

Interpolation methods - Kriging

The "kriging" semi-variogram

- Depicts spatial autocorrelation of measured sample points
- Sill (***c***): Upper limit, where the semi-variance levels off
- Range (***a***): Distance at which the sill is reached
- Nugget (***c$_o$***): Intersection with the y (semi-variance) axis
  - A non-zero values indicated that repeated measurements at the same point yield different values
    - Noise
    - Sub-grid cell variation
    - Measurement error
- Partial sill
  - Sill - nugget



$$s^2 = \frac{\sum (x_i - \bar{x})^2}{N-1}$$

$$\gamma(h) = \frac{\sum (y(x) - y(x+h))^2}{2N}$$

The above equation is used in 2D, what about 3D?

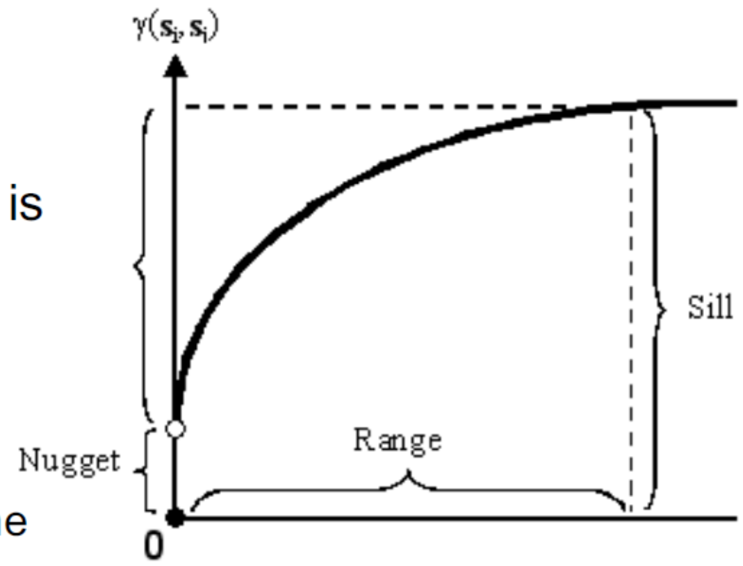
$$\gamma(\Delta x, \Delta y) = \frac{?}{2N}$$

Modified from Jan's slides

Interpolation methods - Kriging

The "kriging" semi-variogram

- Depicts spatial autocorrelation of measured sample points
- Sill (c): Upper limit, where the semi-variance levels off
- Range (a): Distance at which the sill is reached
- Nugget (c_o): Intersection with the y (semi-variance) axis
  - A non-zero values indicated that repeated measurements at the same point yield different values
    - Noise
    - Sub-grid cell variation
    - Measurement error
- Partial sill
  - Sill - nugget



$$s^2 = \frac{\sum (x_i - \bar{x})^2}{N-1}$$

$$\gamma(h) = \frac{\sum (y(x) - y(x+h))^2}{2N}$$

The above equation is used in 2D, what about 3D?

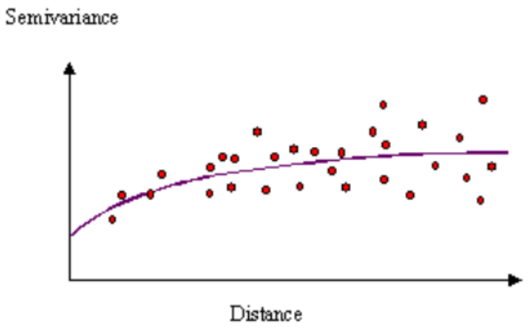
$$\gamma(h) = \frac{\sum (Z_{x+\Delta x, y+\Delta y} - Z_{x,y})^2}{2N}$$

Modified from Jan's slides

Interpolation methods - Kriging

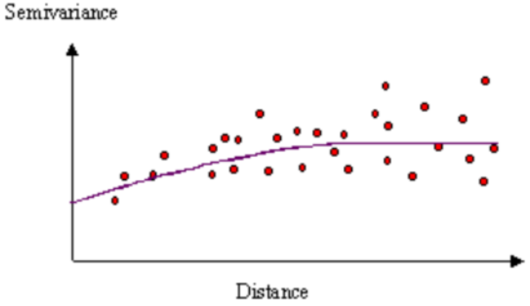
Theoretical semivariograms

Exponential → $\gamma(h) = c_0 + c\left(1 - e^{-\frac{h}{a}}\right)$

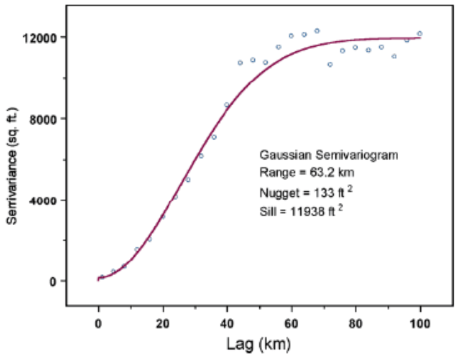


Spherical → $\gamma(h) = c_0 + c\left[\frac{3}{2}\frac{h}{a} - \frac{1}{2}\frac{h^3}{a^3}\right]$ For h <= a

$\gamma(h) = c_0 + c$ For h > a

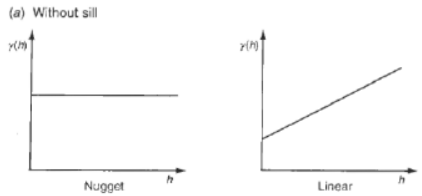


Gaussian → $\gamma(h) = c_0 + c\left(1 - e^{-\frac{h^2}{a^2}}\right)$



From Fortin and Dale - *Spatial Analysis*

Linear → $\gamma(h) = c_0 + bh$



- Power
- Circular

Discussion - Which one to use, if:

a) steep slopes in your terrain?

b) flat terrain?

c) flat terrains with vegetations on it?

Modified from Jan's slides

Interpolation methods - Kriging

	A	B
Count	15251	15251
Average	100.00	100.00
Standard Deviation	20.00	20.00
Median	100.35	100.92
10 Percentile	73.89	73.95
90 Percentile	125.61	124.72

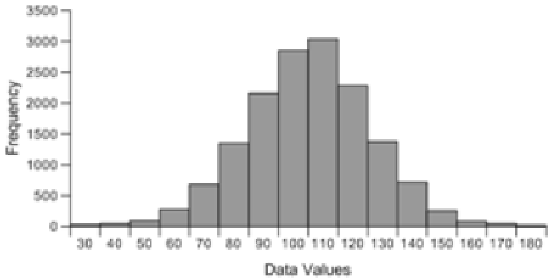


*Figure 1.1 Data Set **A** Histogram*

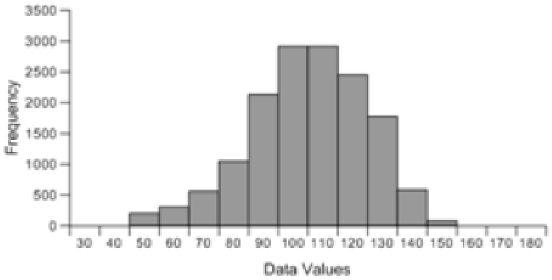


*Figure 1.2 Data Set **B** Histogram*

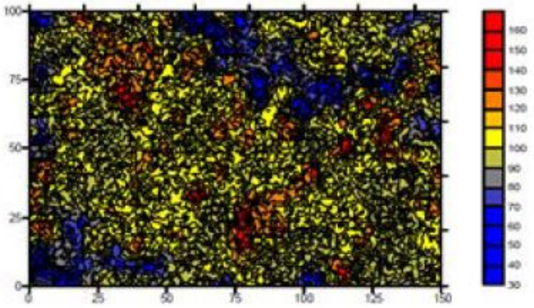


*Figure 1.3 Data Set **A** Contour Plot*

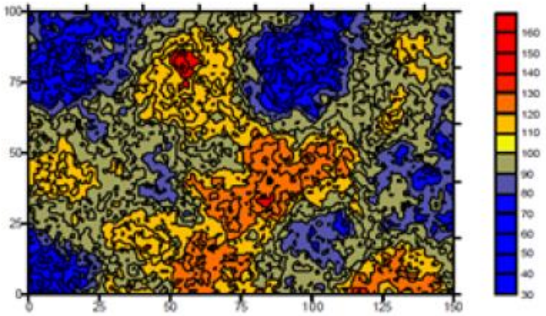


*Figure 1.4 Data Set **B** Contour Plot*

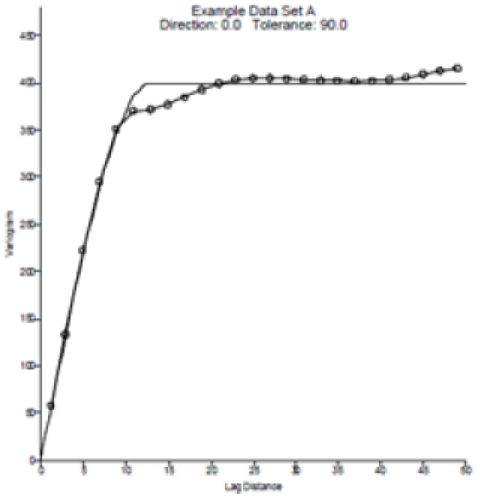


*Figure 1.5 Data Set **A** Variogram and Model*

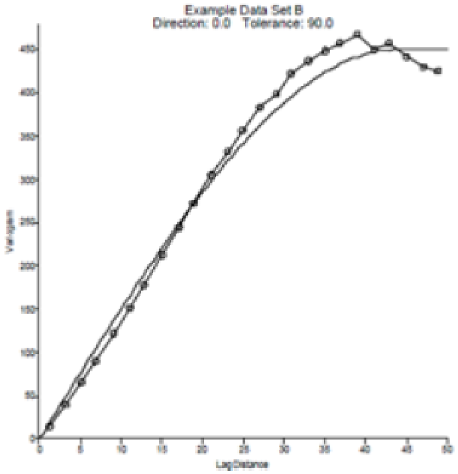


*Figure 1.6 Data Set **B** Variogram and Model*

The kriging algorithm incorporates four essential details:

a) the spacing
b) the inherent length scale of the data
c) the inherent trustworthiness of the data
d) natural phenomena

ref: Surfer15 Full User's Guide, p284, p315-317

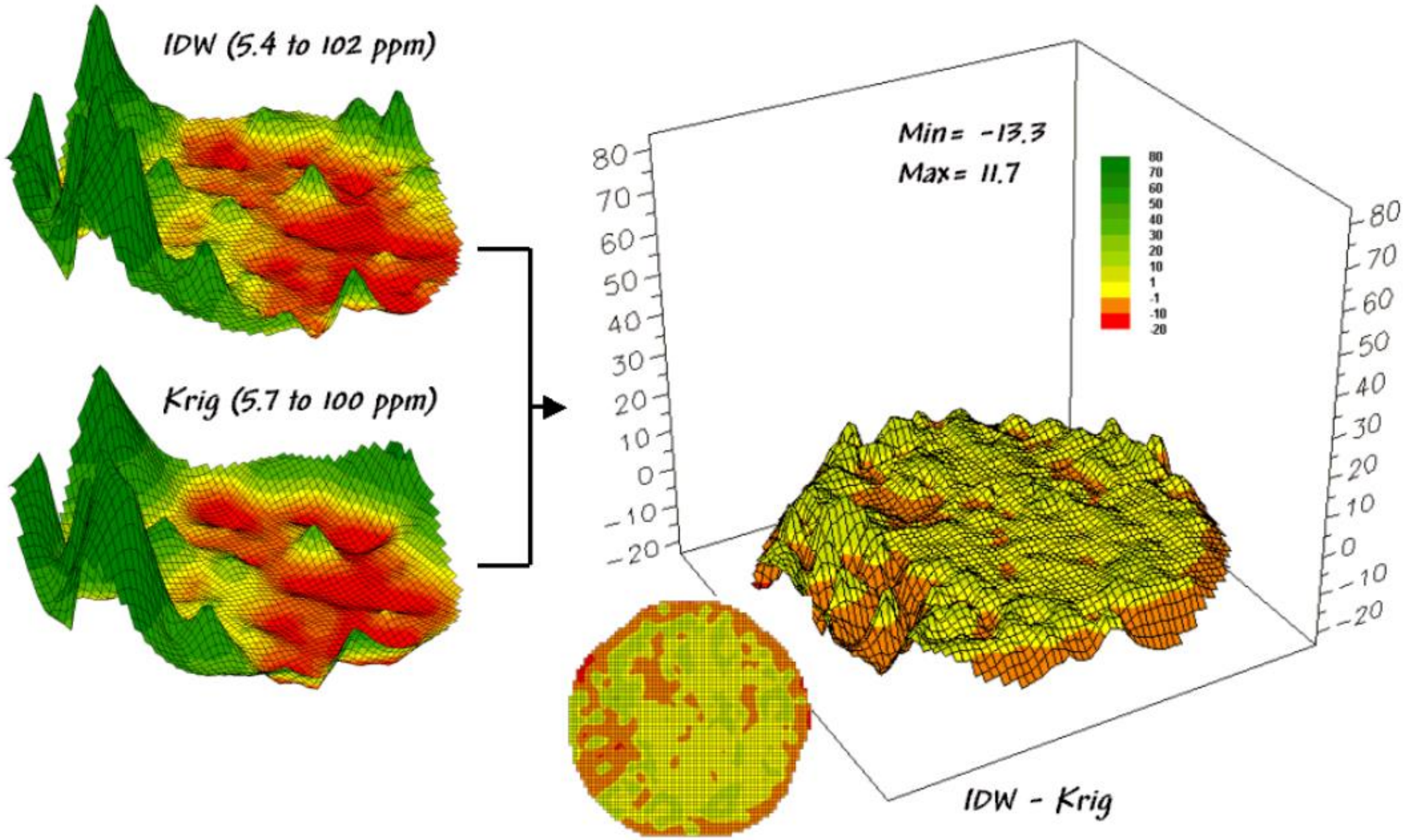
RIT
College of Science
Chester F. Carlson Center for Imaging Science

Interpolation methods

Spatial Interpolation

(*comparing IDW and Kriging results*)

Comparison of the IDW and Kriging interpolated surfaces shows small differences in in localized estimates

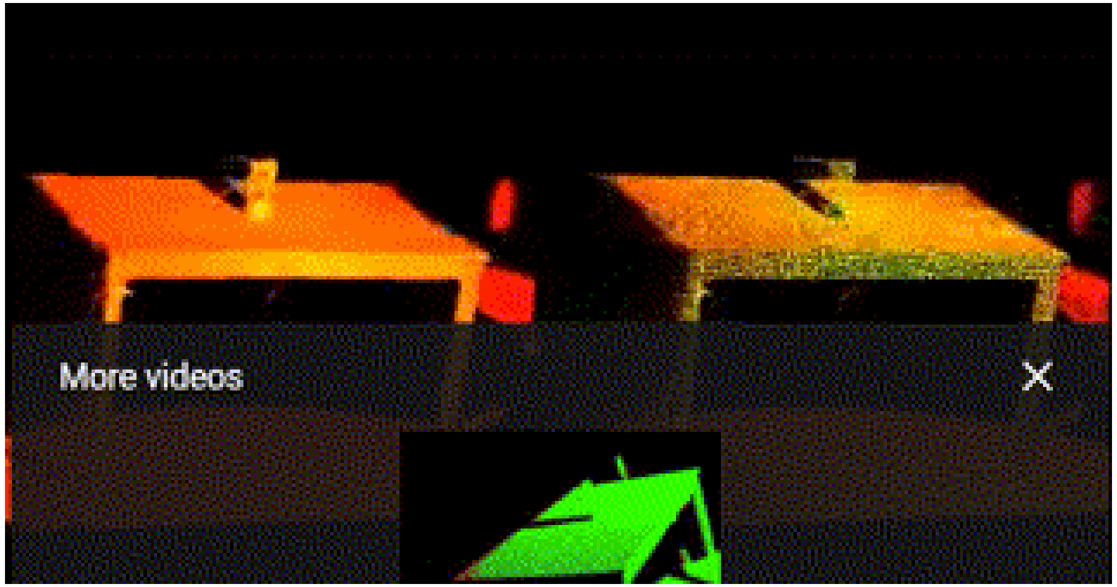


(Berry)

Modified from Jan's slides

Down-sampling methods

- Voxel Grid
- Farthest Point Sampling (FPS)
- Normal Space Sampling (NSS)



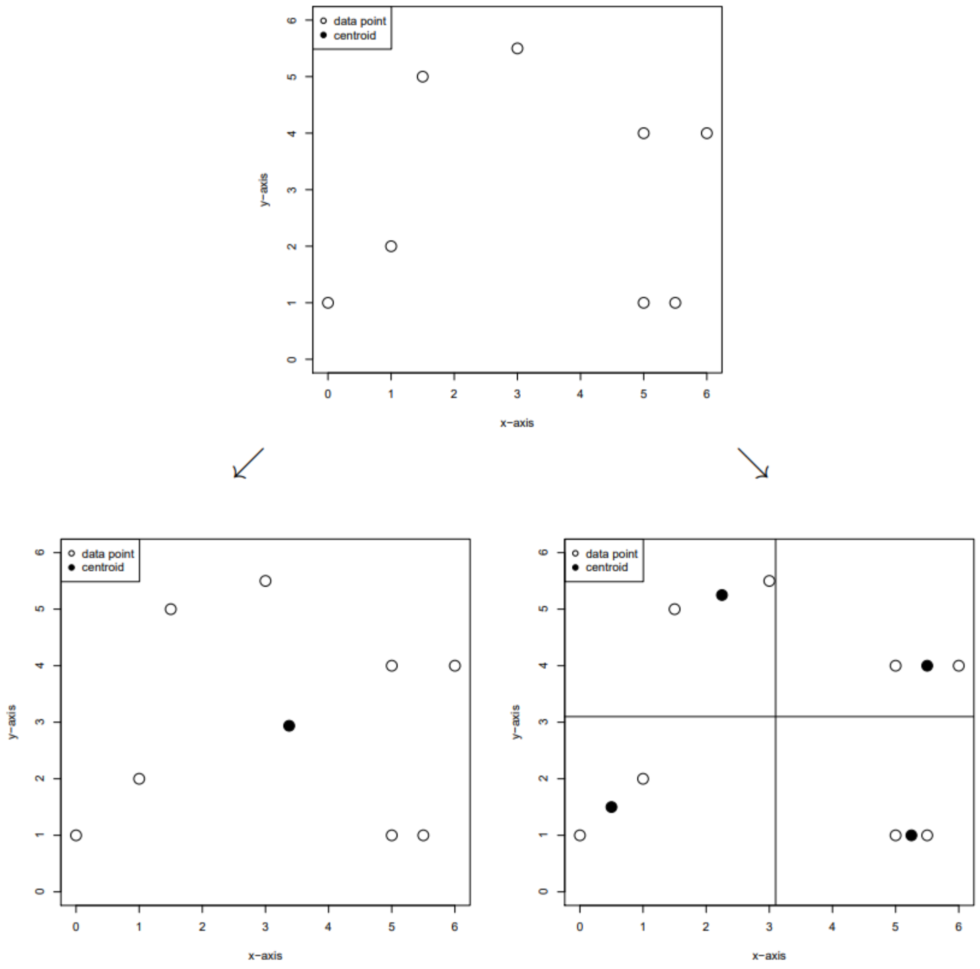
src: https://pointclouds.org/documentation/tutorials/voxel_grid.html

Down-sampling – Voxel Grid

1. Build a voxel grid that contains the point cloud
2. Take one point in each cell

Q: How to take one point?

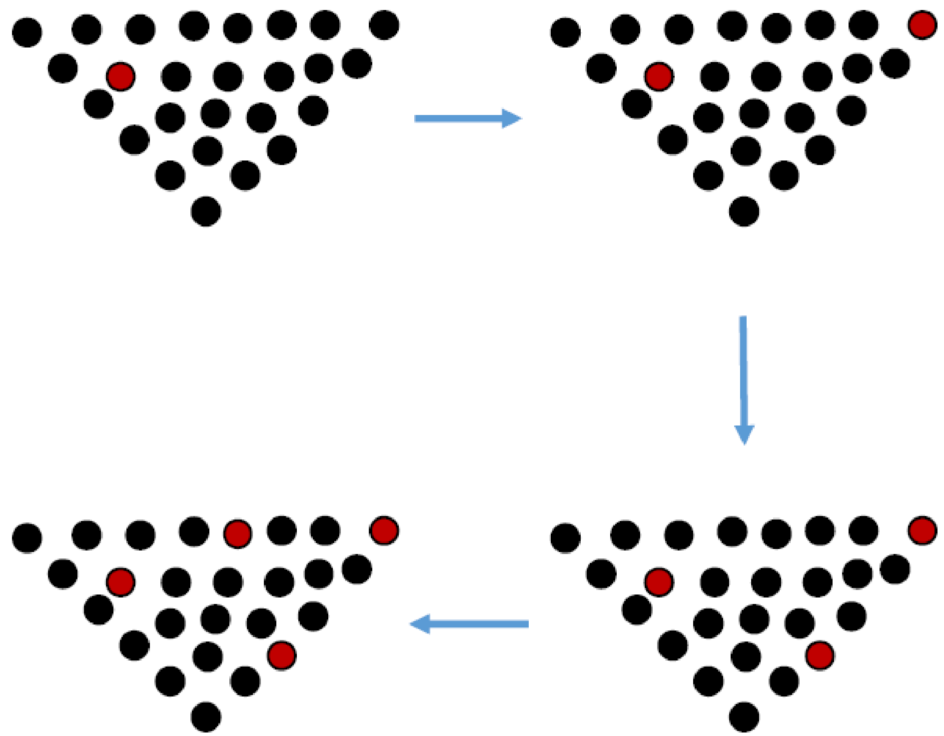
1. Centroid
   a) For coordinates, compute the average in the cell
   b) For other attributes, voting / average
   c) More accurate but slower
2. Random selection
   a) Randomly select a point in the cell
   b) Less accurate but faster



RIT
College of Science
Chester F. Carlson Center for Imaging Science

Down-sampling – Farthest Point Sampling (FPS)

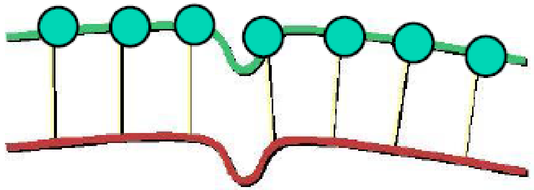
1. Randomly choose a point to be the first FPS point
2. For each point in the original point cloud, compute its distance to the nearest FPS point
3. Choose the point with the largest value, add to FPS set
4. Iterate steps 2 and 3 until we get the desired number of points



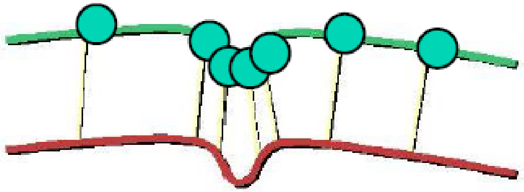
MOST USED IN DEEP LEARNING.

Down-sampling – Normal Space Sampling (NSS)

1. Construct a set of buckets in the normal space
2. Put all points into bucket according to the surface normals
3. Uniformly pick points from all buckets until we have the desired number of points



Uniform Sampling

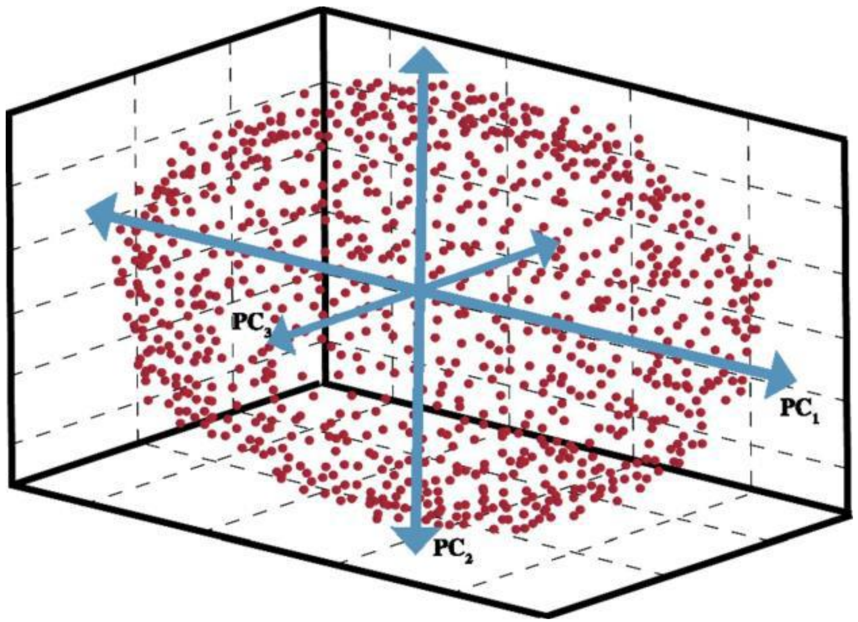


Normal Space Sampling

PCA for point cloud

Applications:

- Dimensionality reduction
- Surface normal estimation
- Key point detection
- Feature description
- Segmentation



RIT
College of Science
Chester F. Carlson Center for Imaging Science

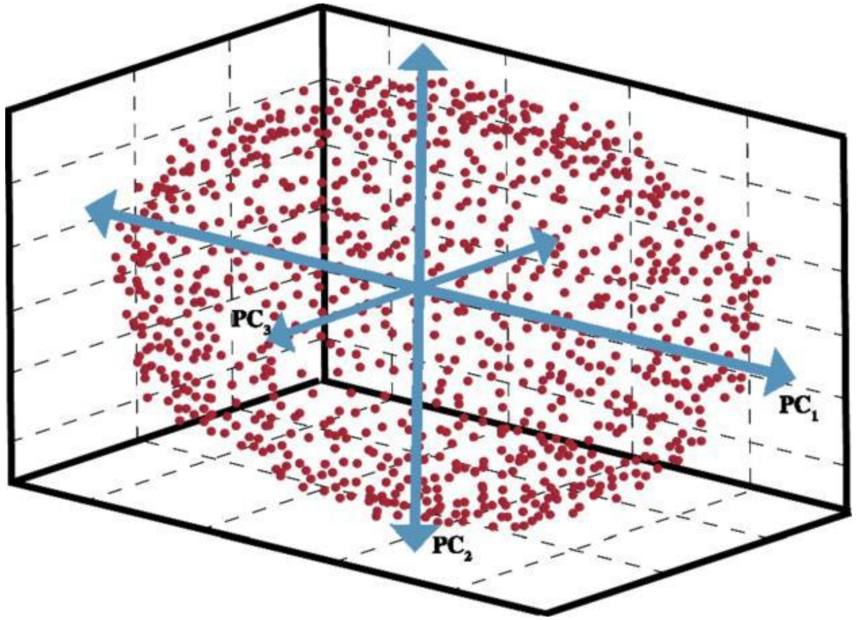
PCA for point cloud

1. Normalized by the center

   $$\tilde{X} = [\tilde{x}_1, \cdots, \tilde{x}_m], \tilde{x}_i = x_i - \bar{x}, i = 1, \cdots, m$$
   $$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i.$$

2. Compute SVD $H = X\tilde{X}^T = U_r\Sigma^2 U_r^T$

3. The principle vectors are the columns of U_r
   (Eigenvector of X = Eigenvector of H)



PCA for point cloud

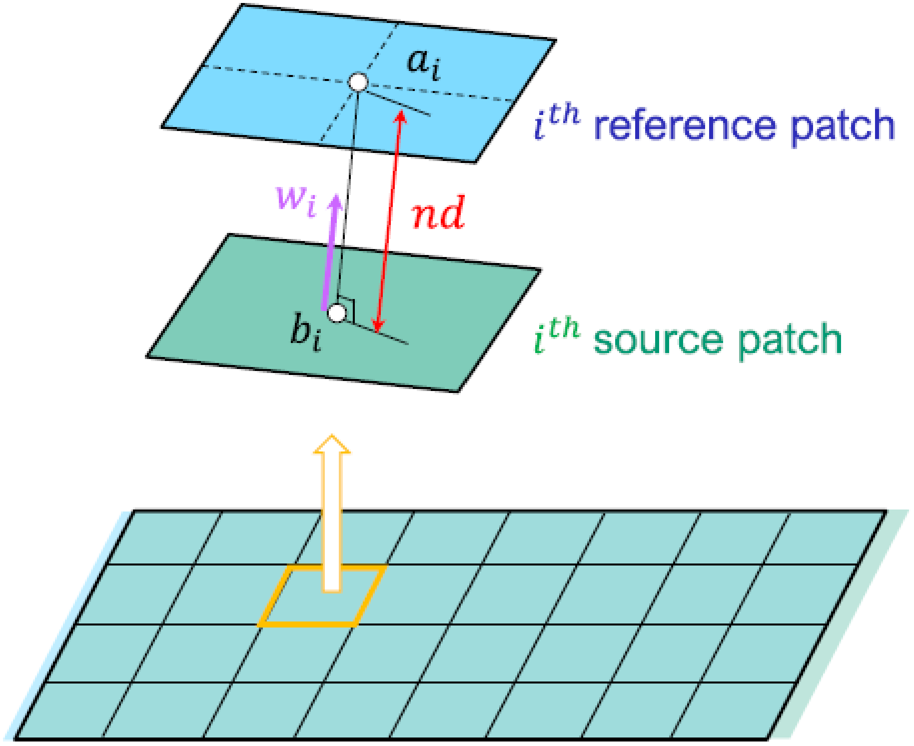


Fig. 7. Illustration of the extraction of terrain planar patches and the discrepancy estimation using conjugate terrain patches.

(Lin and Habib, 2020)

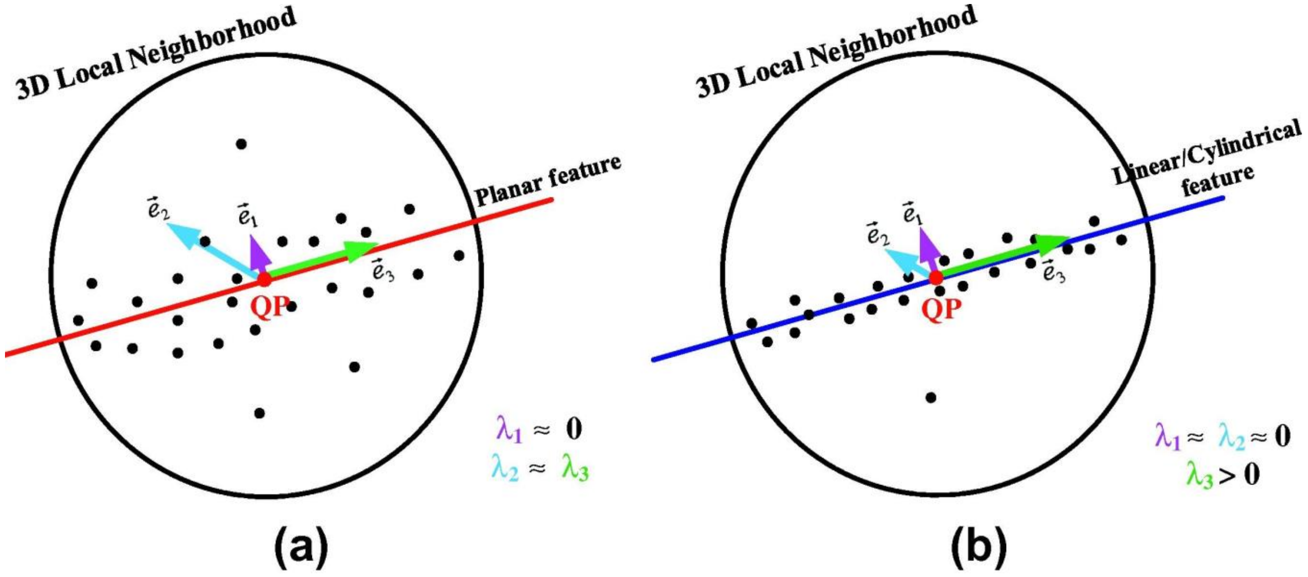
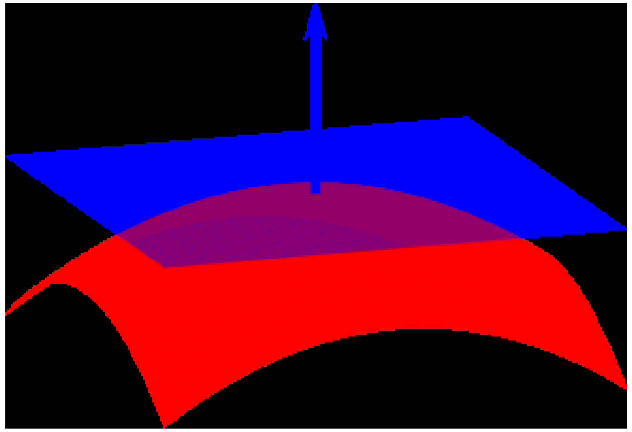
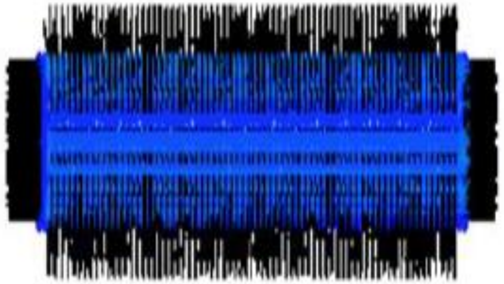


Fig. 2. Established local neighborhood including a: (a) planar feature, and (b) linear/cylindrical feature. (Lari and Habib., 2013)

Surface Normal





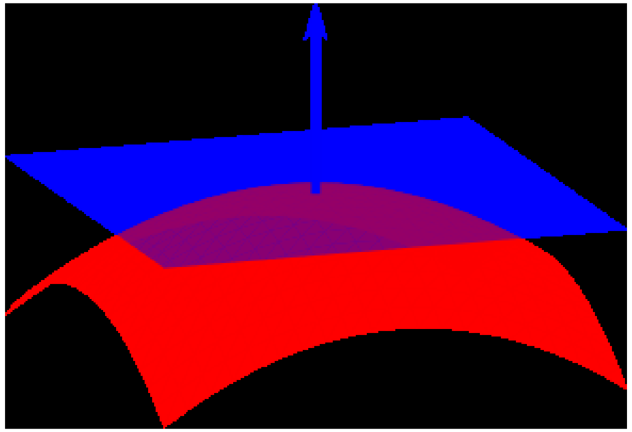
Surface normal on surface

- The vector perpendicular to the tangent plane of the surface at a point P

Applications

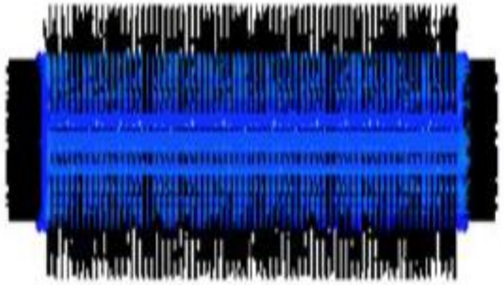
- Segmentation / Clustering
- Plane detection
- Point cloud feature for applications like Deep Learning

Surface Normal



How to compute:

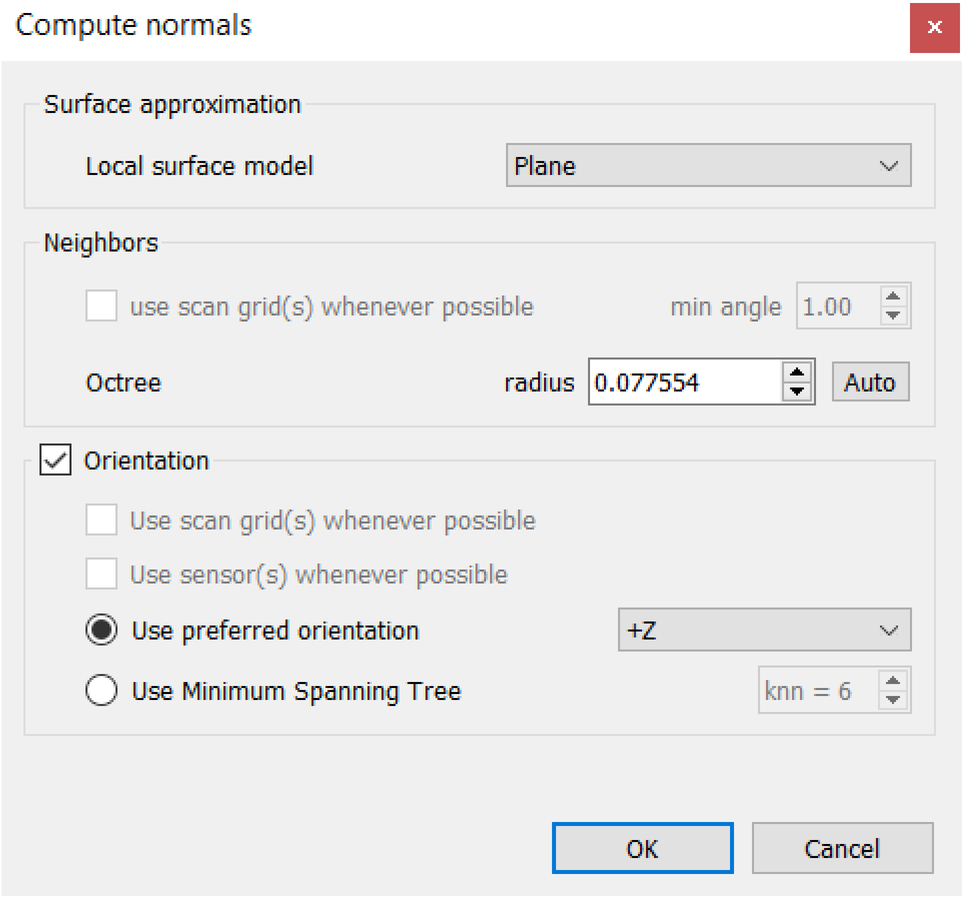
1. select a query point P
2. find the neighborhood that defines the surface
3. PCA
4. Normal -> the least significant vector
5. Curvature -> ratio between eigen values
   $\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$



Surface Normal

How CloudCompare computes normals:

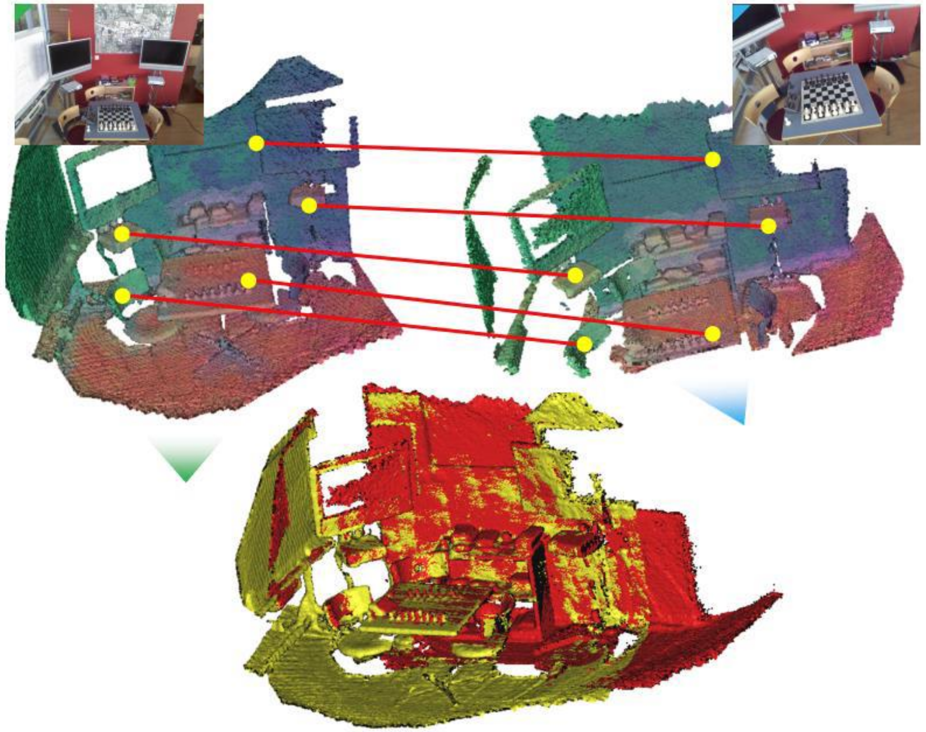
- Local surface model
  - (best fit) plane - robust to noise but very bad with sharp edges and corners
  - 2D triangulation - weak to noise but good with sharp edges
  - quadric - very good for curvy surfaces



Edit > Normals > Compute (dialog of v 2.12 alpha)

Point Cloud Registration

- Find a transform to align two point clouds
- A transform consists of
  - Rotation R
  - Translation t
- Method 1 -Iterative Closest Point (ICP)?
  - ICP requires proper initial guess
  - Low overlapping ratio
- Method 2 –Detect and match features
  - No initialization required
  - Works for low overlapping ratio



ref: Deng et al., 2018; Bellekens et al., 2015

Point Cloud Registration

- Find a transform to align two point clouds
- A transform consists of
  - Rotation R
  - Translation t

$$T = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

How to calculate the parameters? - SVD

Based on the point correspondences, the cross correlation matrix M between the two centered point clouds can be calculated, after which the eigenvalue decomposition is obtained as follows:

$$M = USV^{\intercal} \tag{6}$$

The optimal solution to the least-squares problem is then defined by rotation matrix R as:

$$R_t^s = UV^{\intercal} \tag{7}$$

and the translation from target point cloud to source point cloud is defined by:

$$\tilde{\mathbf{t}} = \tilde{\mathbf{c_s}} - R_t^s \tilde{\mathbf{c_t}} \tag{8}$$

ref: Bellekens et al., 2015

RIT

College of Science
Chester F. Carlson Center for Imaging Science

Point Cloud Registration

- Find a transform to align two point clouds
- A transform consists of
  - Rotation R
  - Translation t
- Method 1 -Iterative Closest Point (ICP)?
  - ICP requires proper initial guess
  - Low overlapping ratio
- Method 2 –Detect and match features
  - No initialization required
  - Works for low overlapping ratio

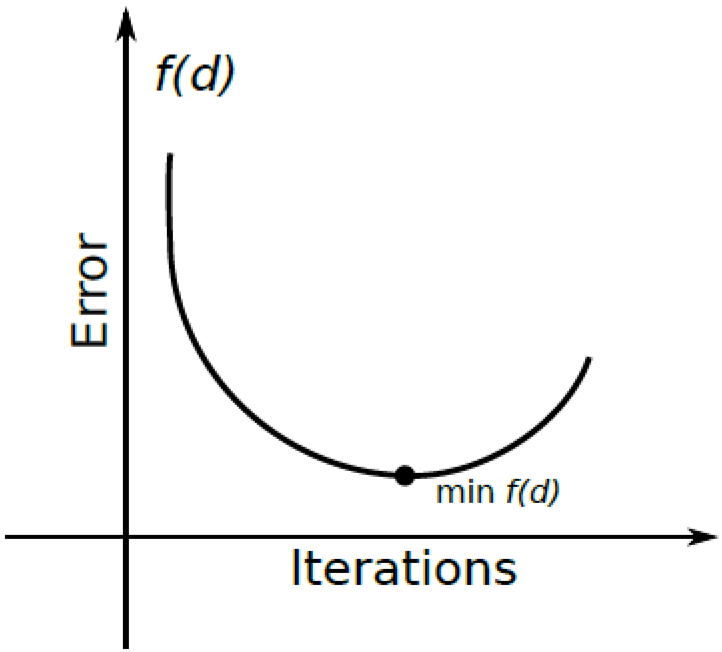


Figure 1. ICP Least square approach.

ref: Bellekens et al., 2015

Iterative Closest Points (ICP) registration

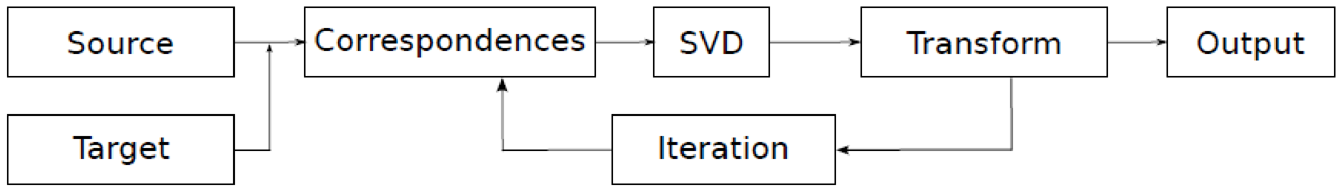


Figure 3. ICP overview scheme.

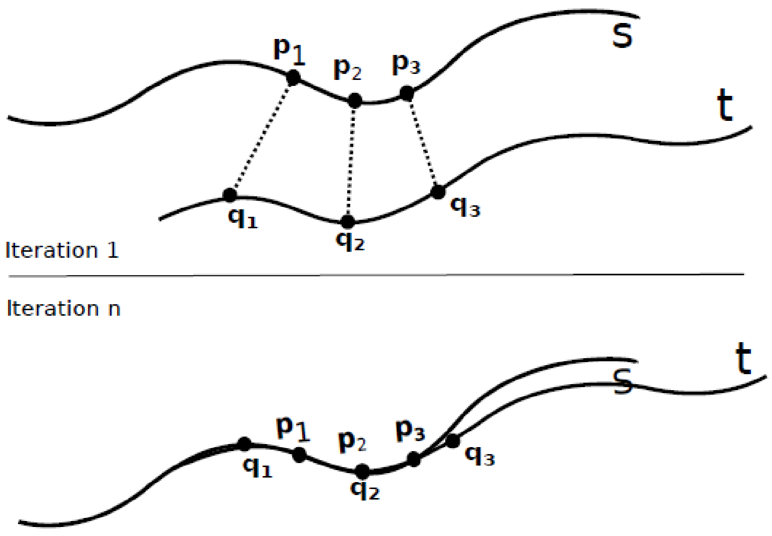


Figure 4. ICP alignment based on a point to point approach.

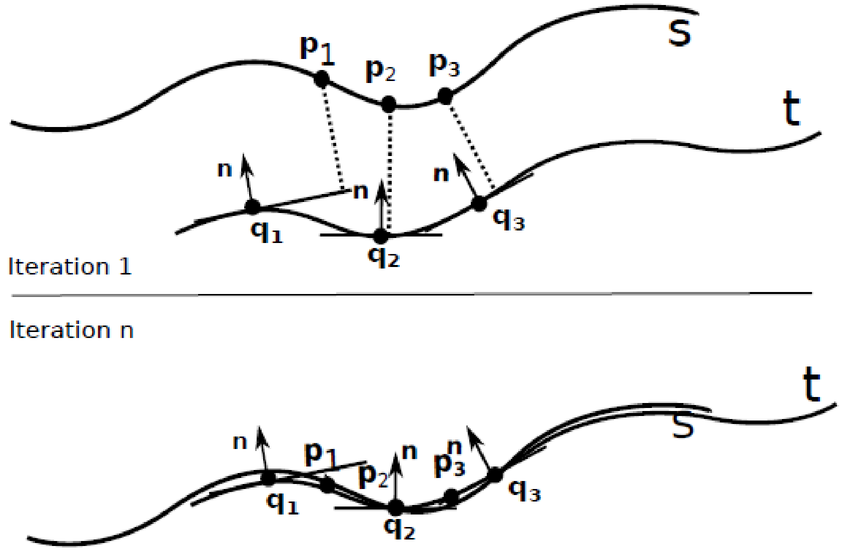
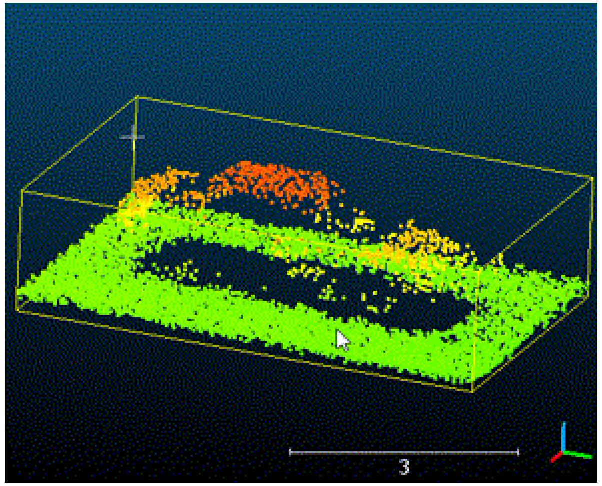


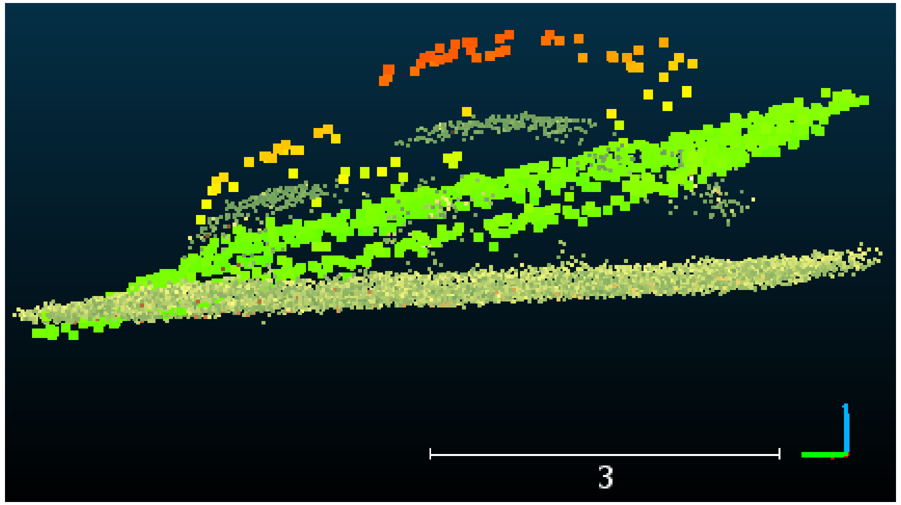
Figure 5. ICP alignment based on a point to surface approach.

ref: Bellekens et al., 2015

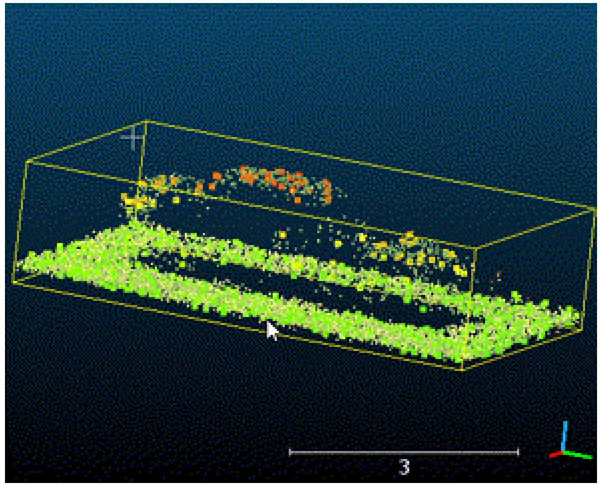
Iterative Closest Points (ICP) registration



Car – original



Car – to be registered



Car – registered

Lab for Week 13

Interaction between 3D Point cloud and 2D image mosaics:

- Lab 1: Colorize your LiDAR point cloud using 2D image mosaics;
- Lab 2:
  1) 2D segmentation using spectral angle mapper;
  2) sampling the segmentation result using the 3D point cloud.

SfM point cloud

- Principles of SfM
- SfM – LiDAR registration
- Comparison between LiDAR and SfM point cloud

Reference

- Alsadik, B.; Remondino, F. Flight Planning for LiDAR-Based UAS Mapping Applications. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 378. https://doi.org/10.3390/ijgi9060378
- Kidd, John Ryan, "Performance Evaluation of the Velodyne VLP-16 System for Surface Feature Surveying" (2017). Master's Theses and Capstones. 1116. https://scholars.unh.edu/thesis/1116
- Juan Guerra-Hernández, Diogo N. Cosenza, Luiz Carlos Estraviz Rodriguez, Margarida Silva, Margarida Tomé, Ramón A. Díaz-Varela & Eduardo González-Ferreiro (2018) Comparison of ALS- and UAV(SfM)-derived high-density point clouds for individual tree detection in Eucalyptus plantations, International Journal of Remote Sensing, 39:15-16, 5211-5235, DOI: 10.1080/01431161.2018.1486519
- Klápště, Petr, Michal Fogl, Vojtěch Barták, Kateřina Gdulová, Rudolf Urban, and Vítězslav Moudrý. "Sensitivity analysis of parameters and contrasting performance of ground filtering algorithms with UAV photogrammetry-based and LiDAR point clouds." *International Journal of Digital Earth* 13, no. 12 (2020): 1672-1694.
- Zhang, Wuming, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xie, Xiaoyan Wang, and Guangjian Yan. "An easy-to-use airborne LiDAR data filtering method based on cloth simulation." *Remote Sensing* 8, no. 6 (2016): 501.
- Lari, Zahra, and Ayman Habib. "An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data." *ISPRS Journal of Photogrammetry and Remote Sensing* 93 (2014): 192-212.
- Bellekens, Ben, Vincent Spruyt, Rafael Berkvens, Rudi Penne, and Maarten Weyn. "A benchmark survey of rigid 3D point cloud registration algorithms." *Int. J. Adv. Intell. Syst* 8 (2015): 118-127.
- Deng, Haowen, Tolga Birdal, and Slobodan Ilic. "Ppfnet: Global context aware local features for robust 3d point matching." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 195-205. 2018.