Survey of Methods in Computer Graphics:

Robust Moving Least-squares Fitting with Sharp Features

S. Fleishman, D. Cohen-Or, C. T. Silva

SIGGRAPH 2005.

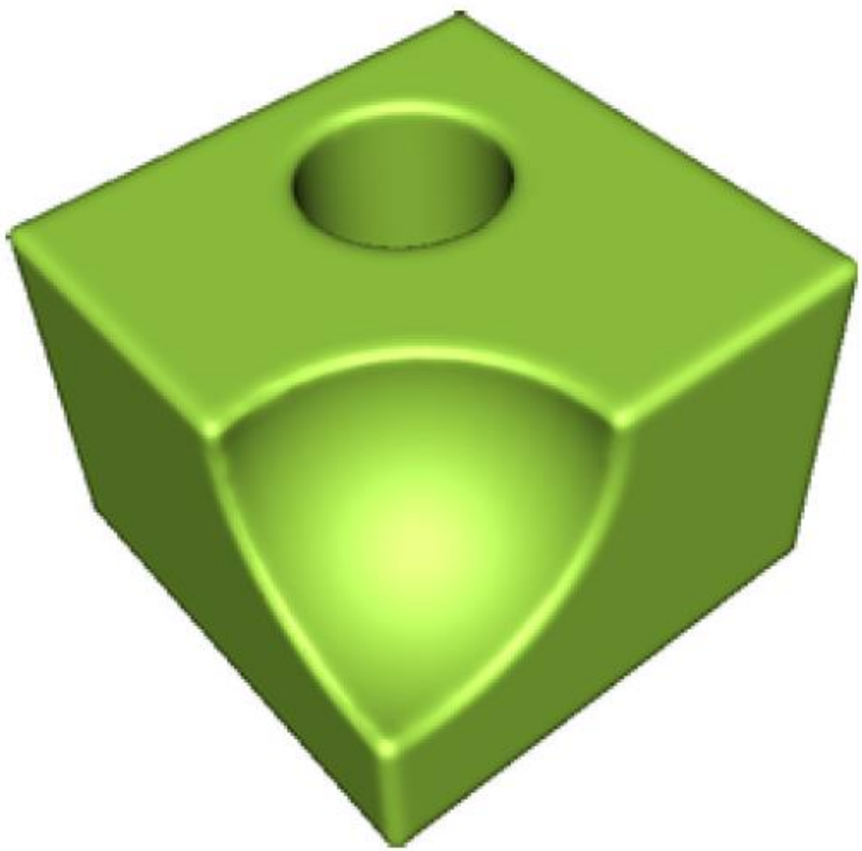
Method Pipeline

Input:
- Point to project on the surface.
- Cloud of points sampled form the surface.
- Orientation **not** required?

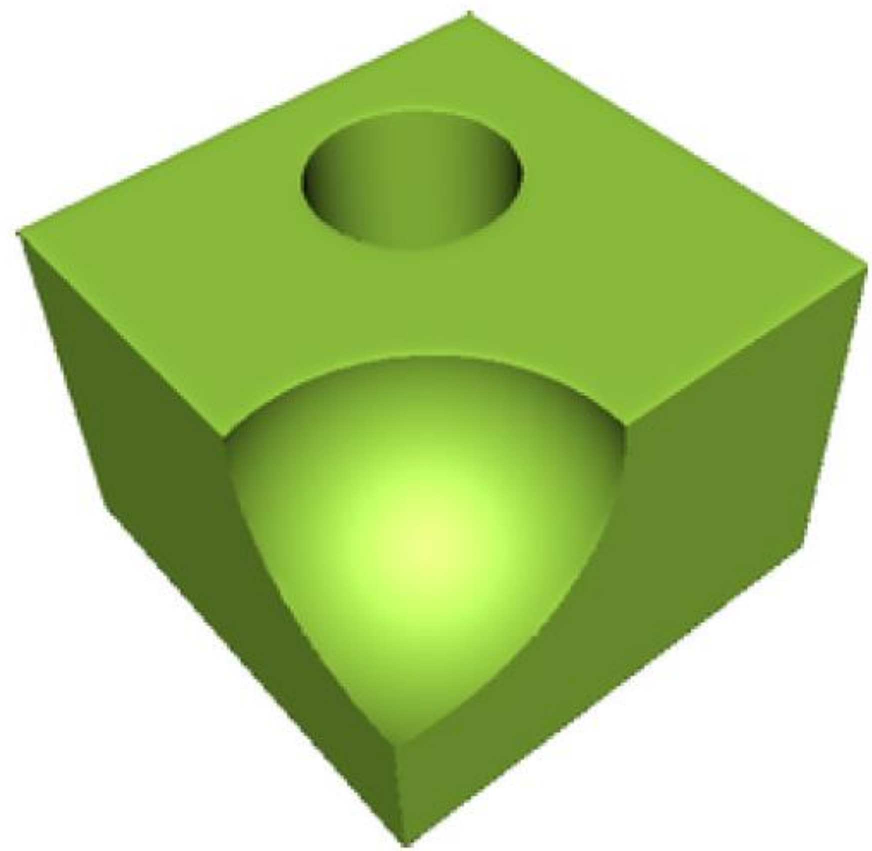
Output:
- Point projected on a Piecewise Smoothn MLS surface.

Guarantees:
- Piecewise Smoothness.
- Manifold?

Smoothness vs Sharpness



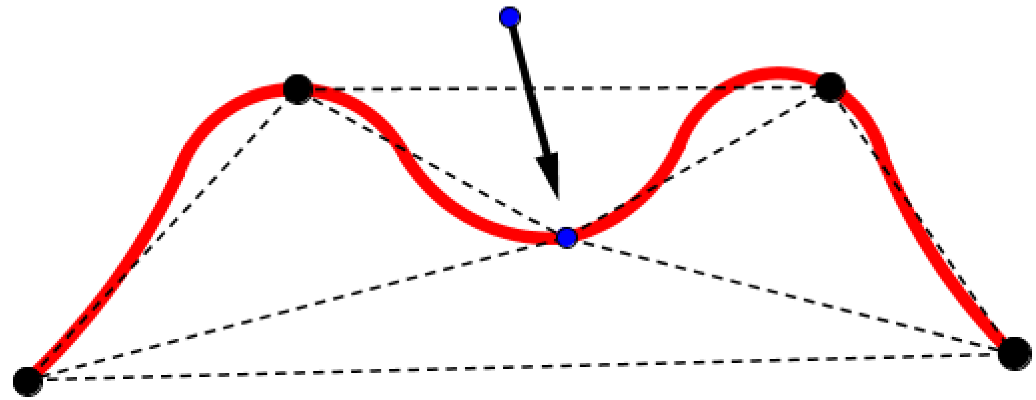
Global MLS
(Levin, Alexa et.al)



Piecewise MLS
(Fleishman et.al)

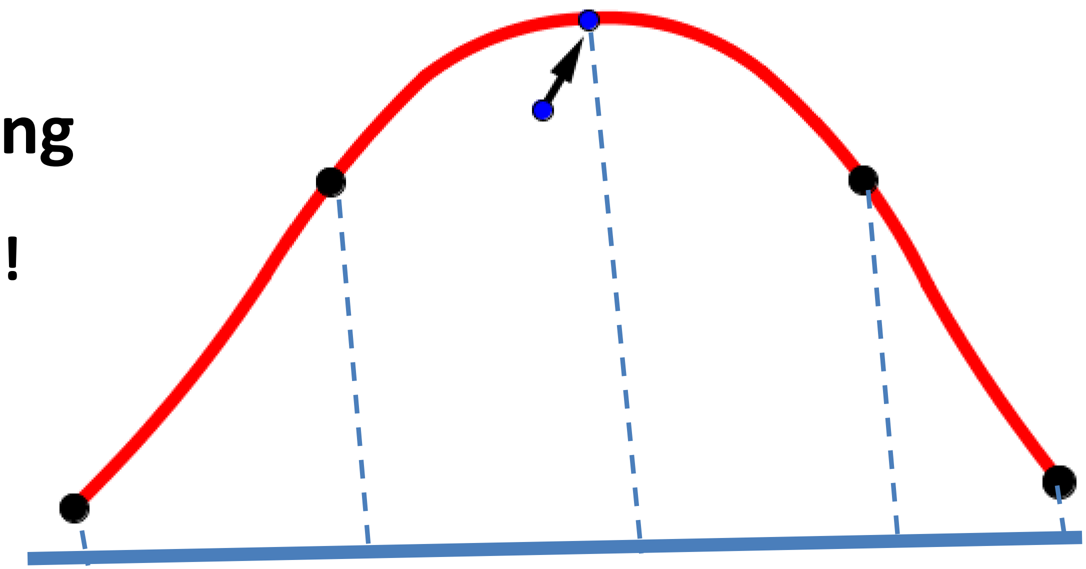
Point Reallocation

1) Smoothing



2) Polynomial Fitting

Avoid Shrinking!



Sharp Features

Interscetion of smooth surfaces

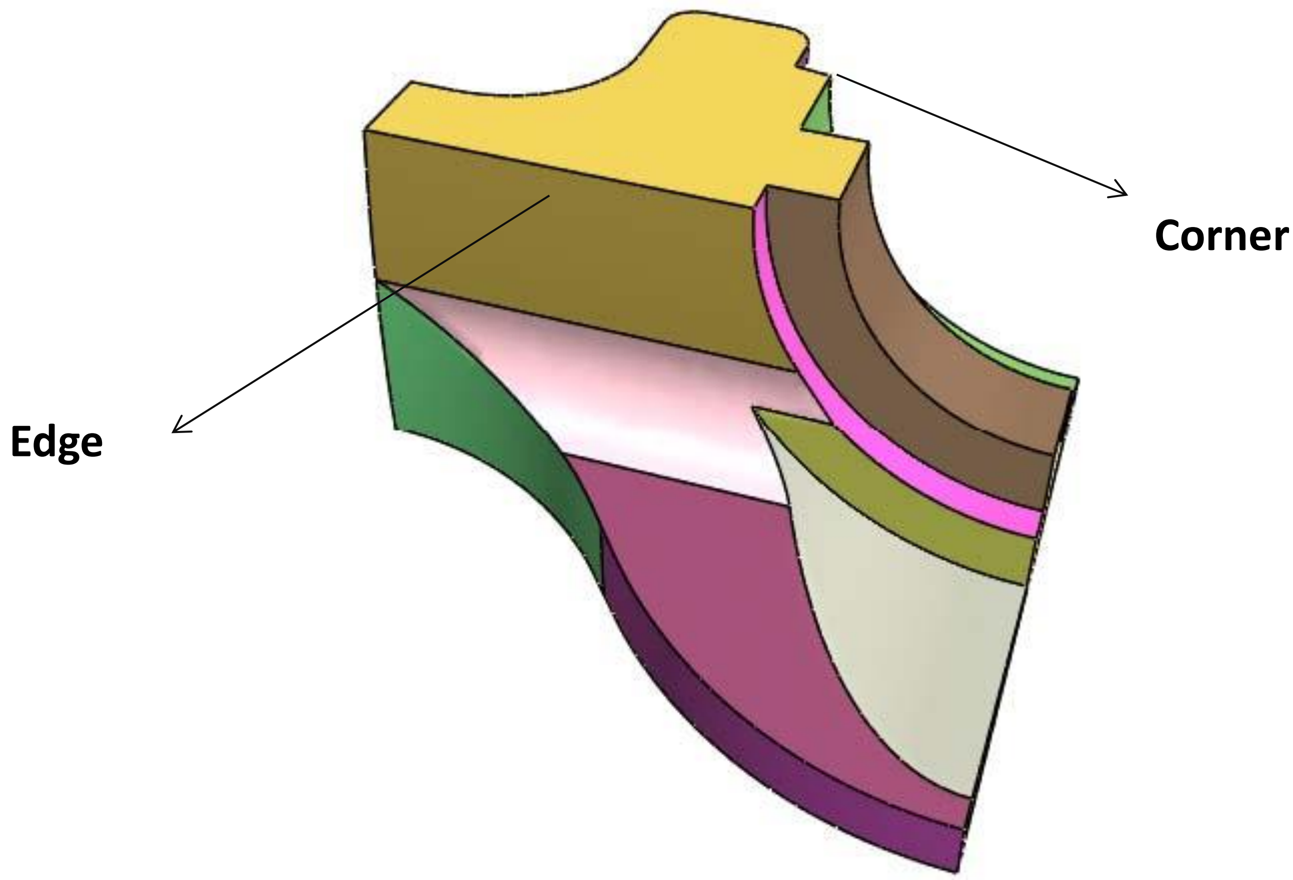
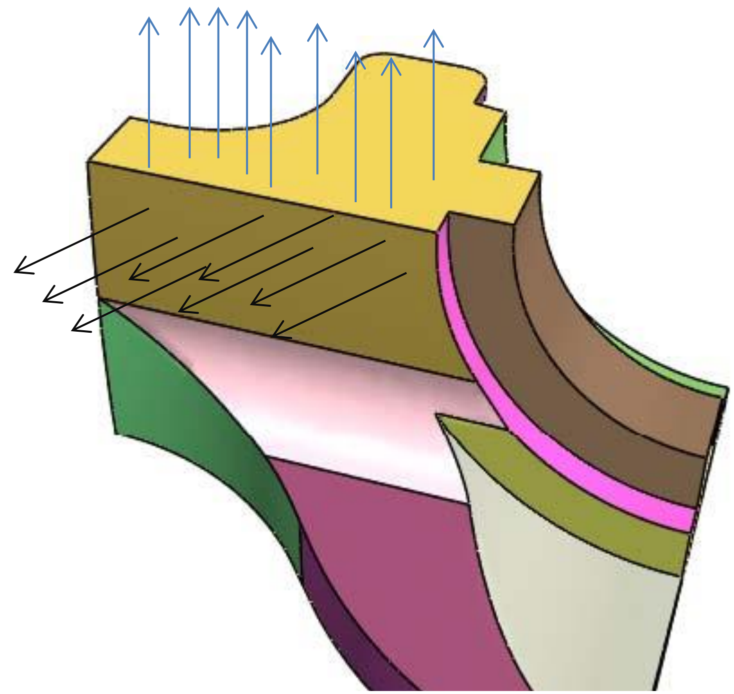


Image from:

http://i.cs.hku.hk/GraphicsGroup/projects/segmentation/segmentation.php

1) Different surfaces can be distinguished computing normals in noise free data sets.



2) In noisy data sets normal computation near sharp features is not accurate.

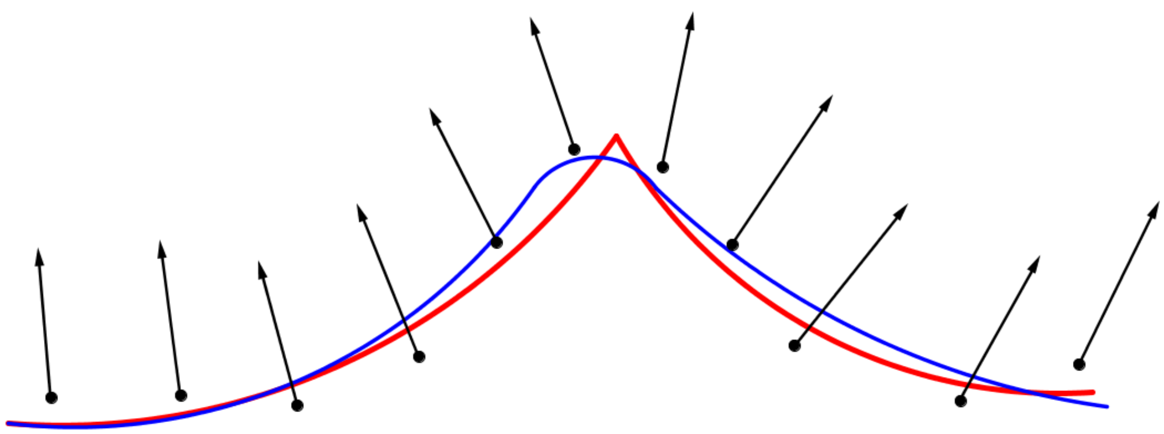


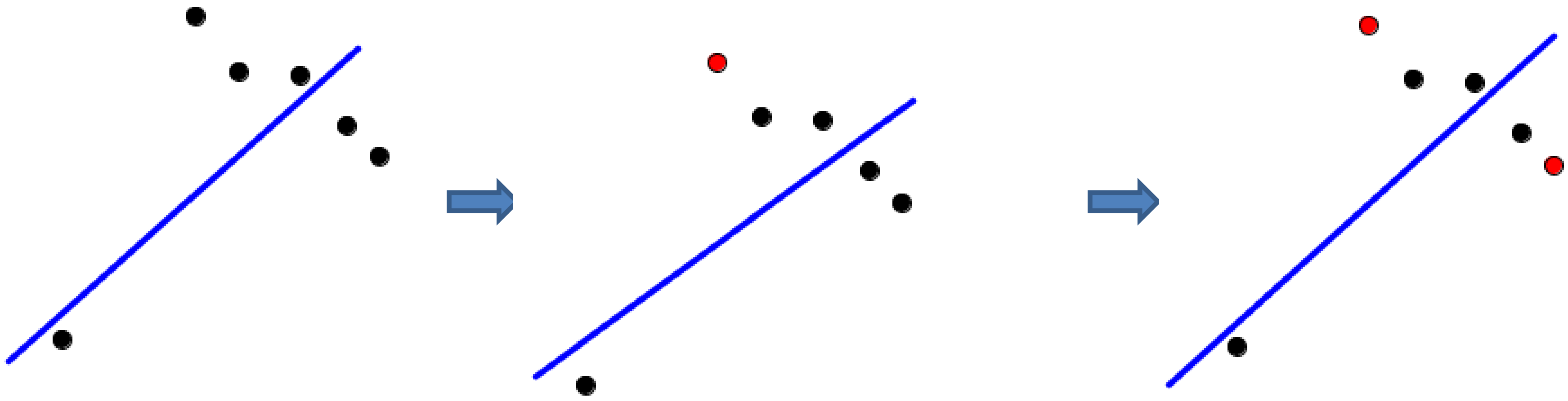
Image from:

http://i.cs.hku.hk/GraphicsGroup/projects/segmentation/segmentation.php

Statistical Methods

Backward Method

- Fit a model for the entire set.
- Incrementally remove outliers and update the model.

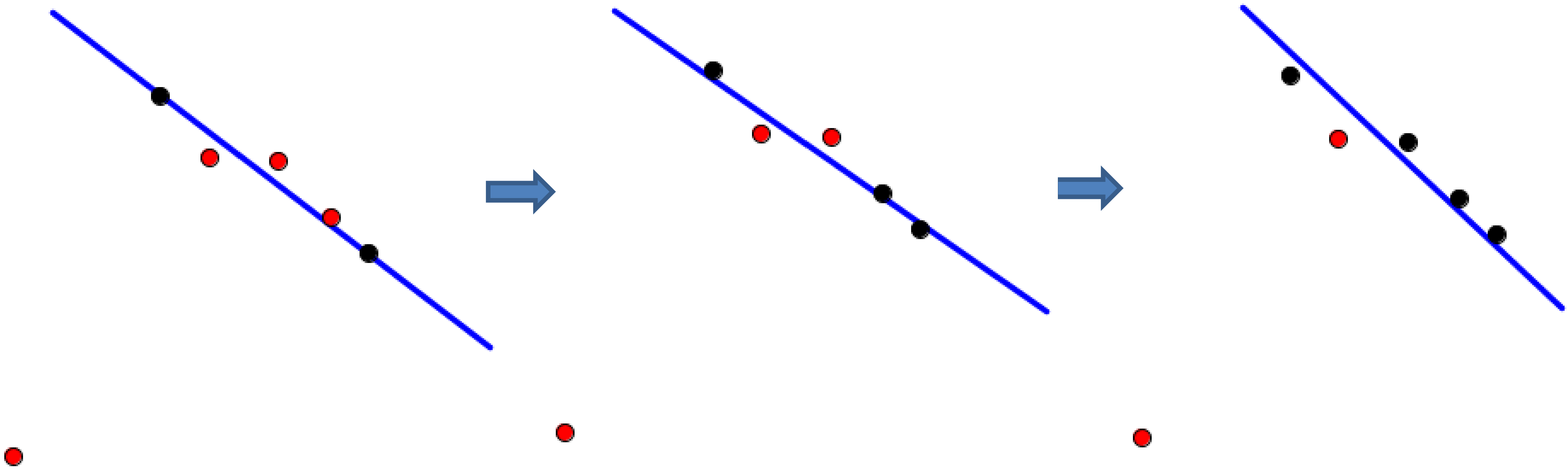


A single outlier may corrupt the estimator.

Statistical Methods

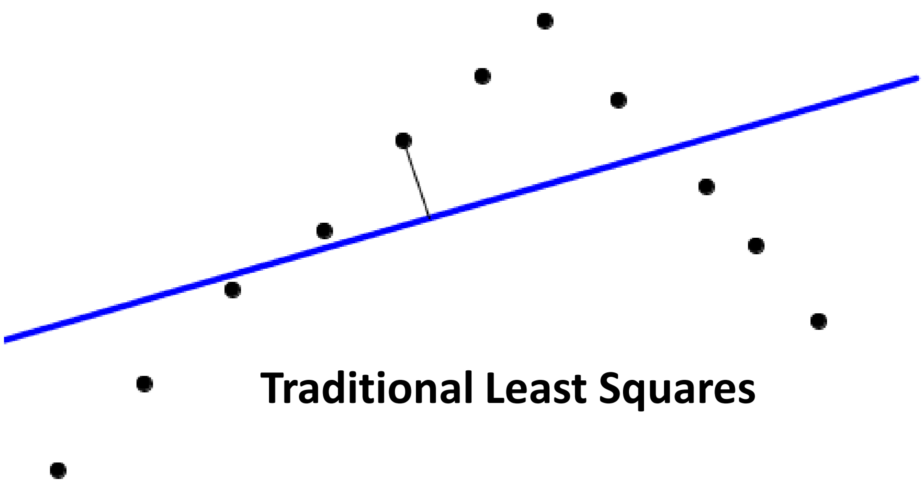
Forward Method

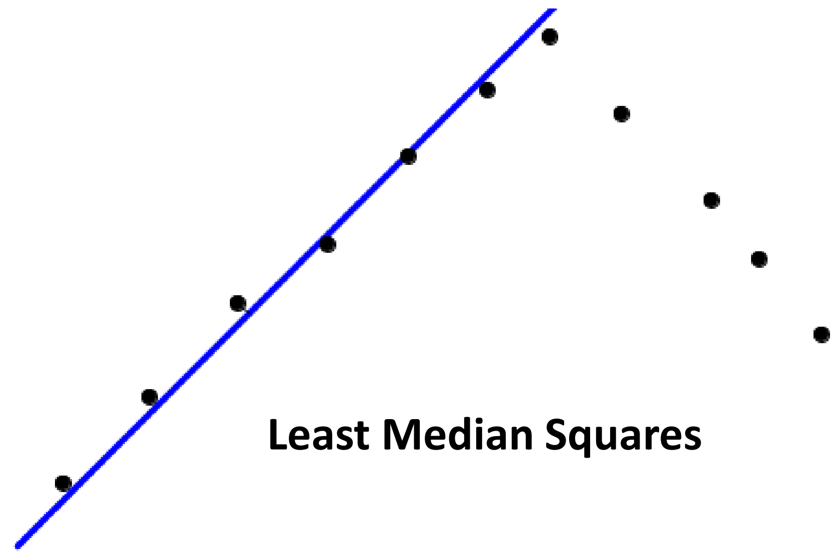
- Fit a model for a selected reliable initial set.
- Incrementally udpate the inlier set and the model.



Least Median of Squares (LMS)

$$\underset{\beta}{\operatorname{argmin}} \; \underset{i}{\operatorname{median}} \left| f_\beta(\mathbf{x}_i) - y_i \right|$$





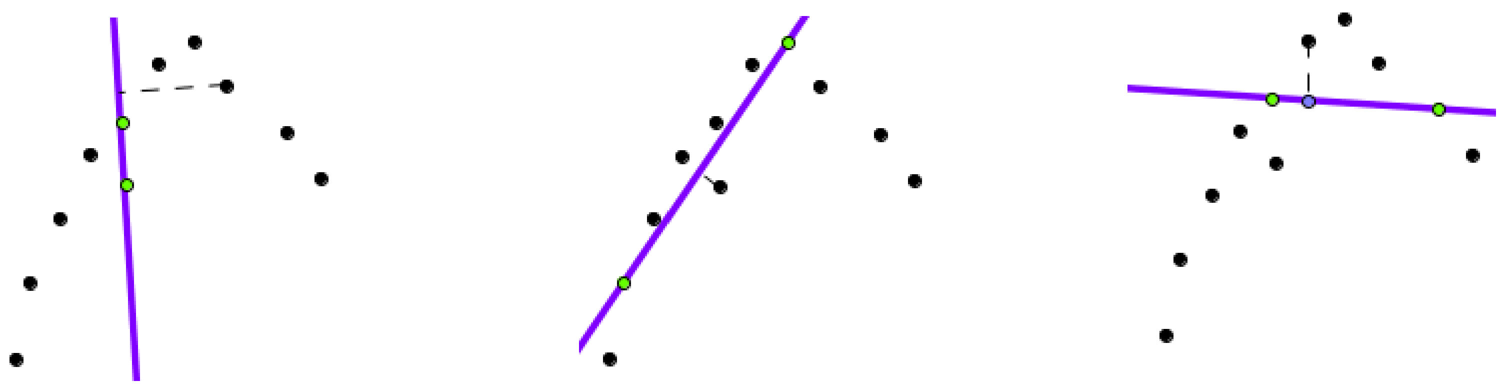
Least Median Squares fit models with up to 50% outliers.

Least Median of Squares (LMS)

Computing an optimal estimator, $\underset{\beta}{\operatorname{argmin}}\ \underset{i}{\operatorname{median}} \left| f_\beta(\mathbf{x}_i) - y_i \right|$, is hard.

Instead, an iterative approach is used:

1. Define a number of experiments T.
2. In each experiment select k samples at random to fit a model, and compute the median error using the remaining $N - k$ samples.
3. Select the experiment with the lowest median error.

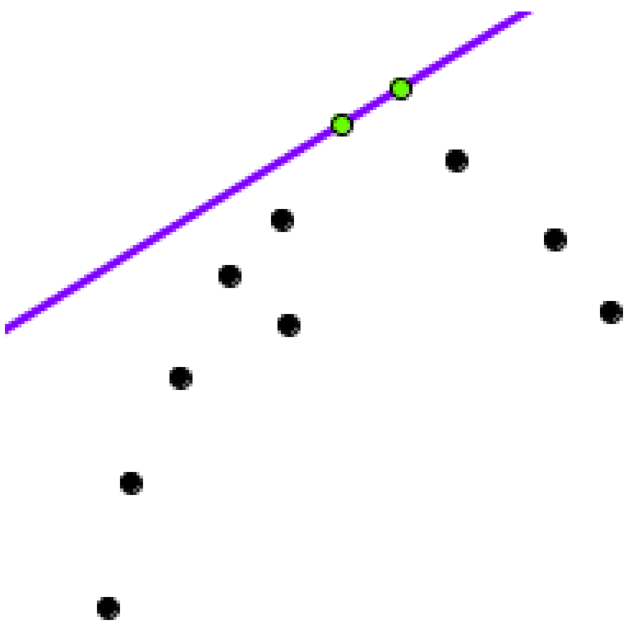


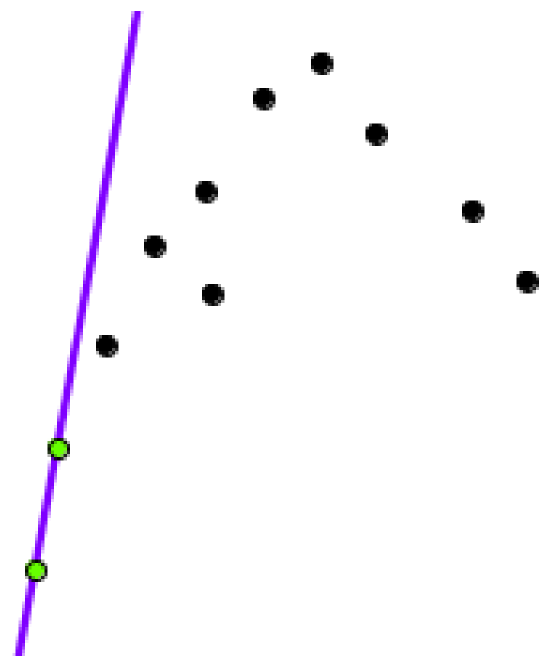
Best Result!

Least Median of Squares (LMS)

The number of samples used o fit the model, k, must be greater than equal than the number of parameters of the model, p. (For instance $p = 2$ to fit a line, and $p = 3$ to fit a plane).

If k is **SMALL** you may not consider enough samples to build a good initial model

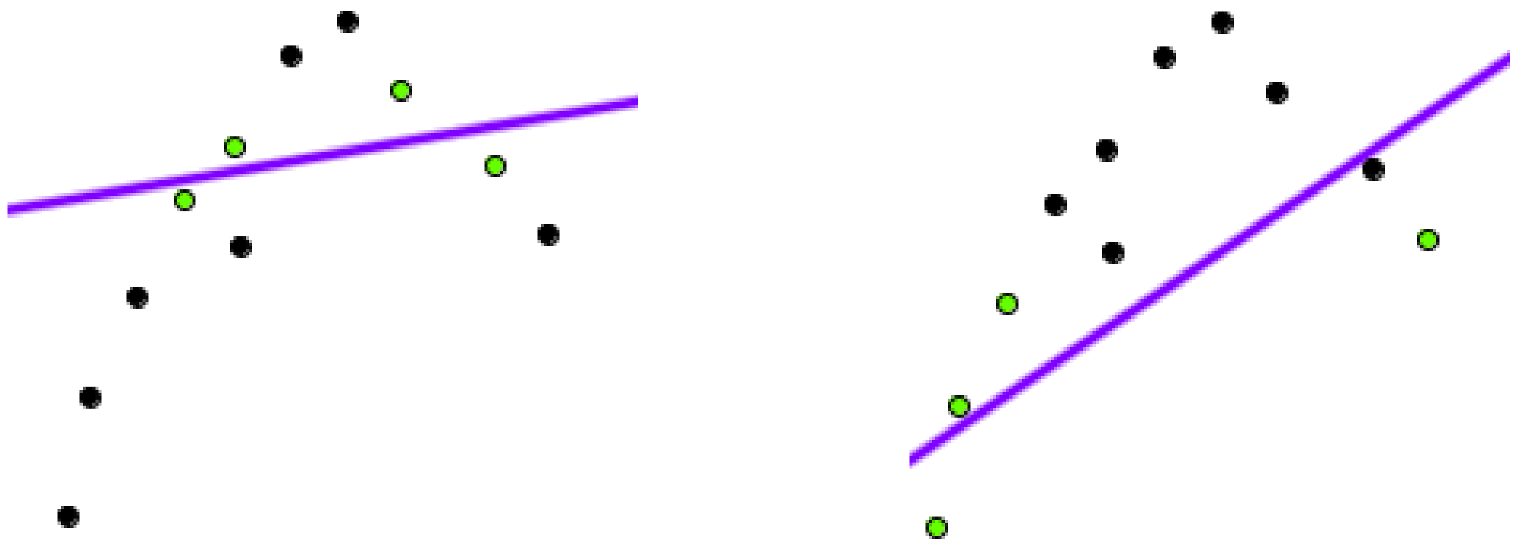




Least Median of Squares (LMS)

The number of samples used o fit the model, k, must be greater than equal than the number of parameters of the model, p. (For instance $p = 2$ to fit a line, and $p = 3$ to fit a plane).

If k is **BIG** you increase the probability to introduce outliers



If g is the proportion of inliers the probability of having a good fit after T experiments is $1 - (1 - g^k)^T$

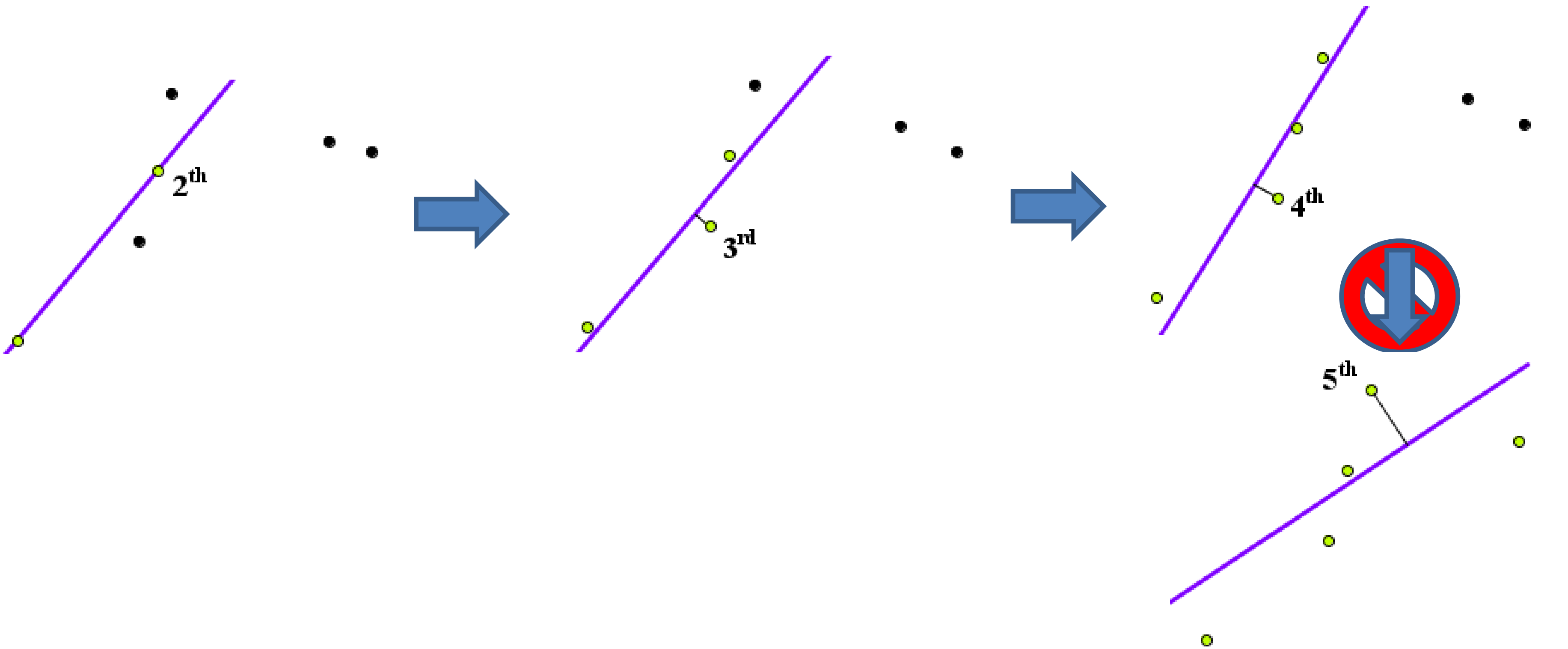
Forward Search

1) Find an initial model and set of inliers (E_p, Q_p) using LMS with $k = p$.

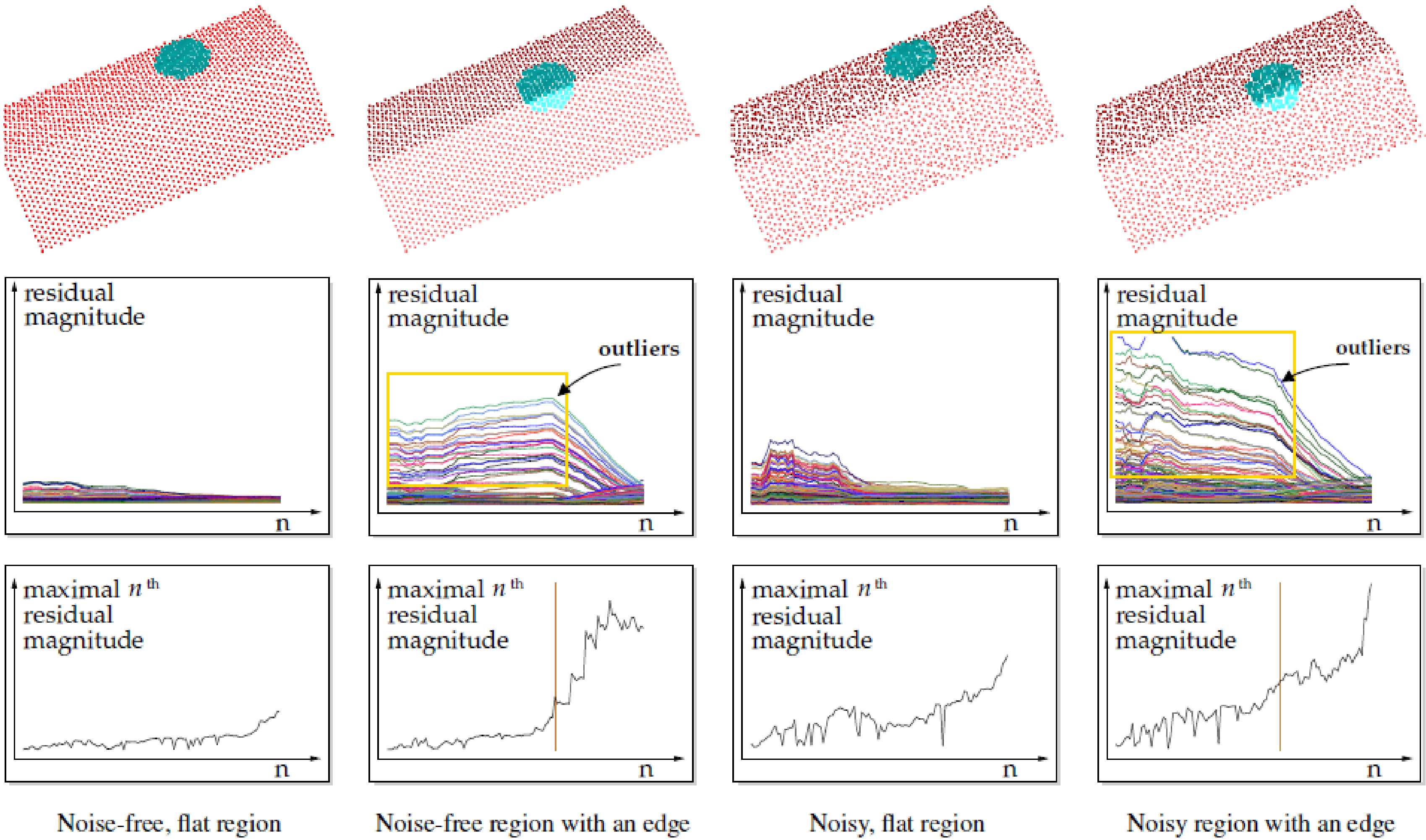
2) For $i > p$:

- Define Q_i as the set of i samples with smallest residual in the current model E_{i-1}. Fit a new model E_i from the samples in Q_i.
- If the $i-th$ order statistic is above a threshold break the loop.

3) Return (E_{i-1}, Q_{i-1}).



Forward Search



Surface Fitting with Forward Search

Algorithm 1: Smooth Component Fitting

1) Using LMS, fit a bivariate **polinomial of degree 2** E_p over a small number of points. This defines the initial set of inliers Q_p and the reference plane π.

2) For $i > p$:

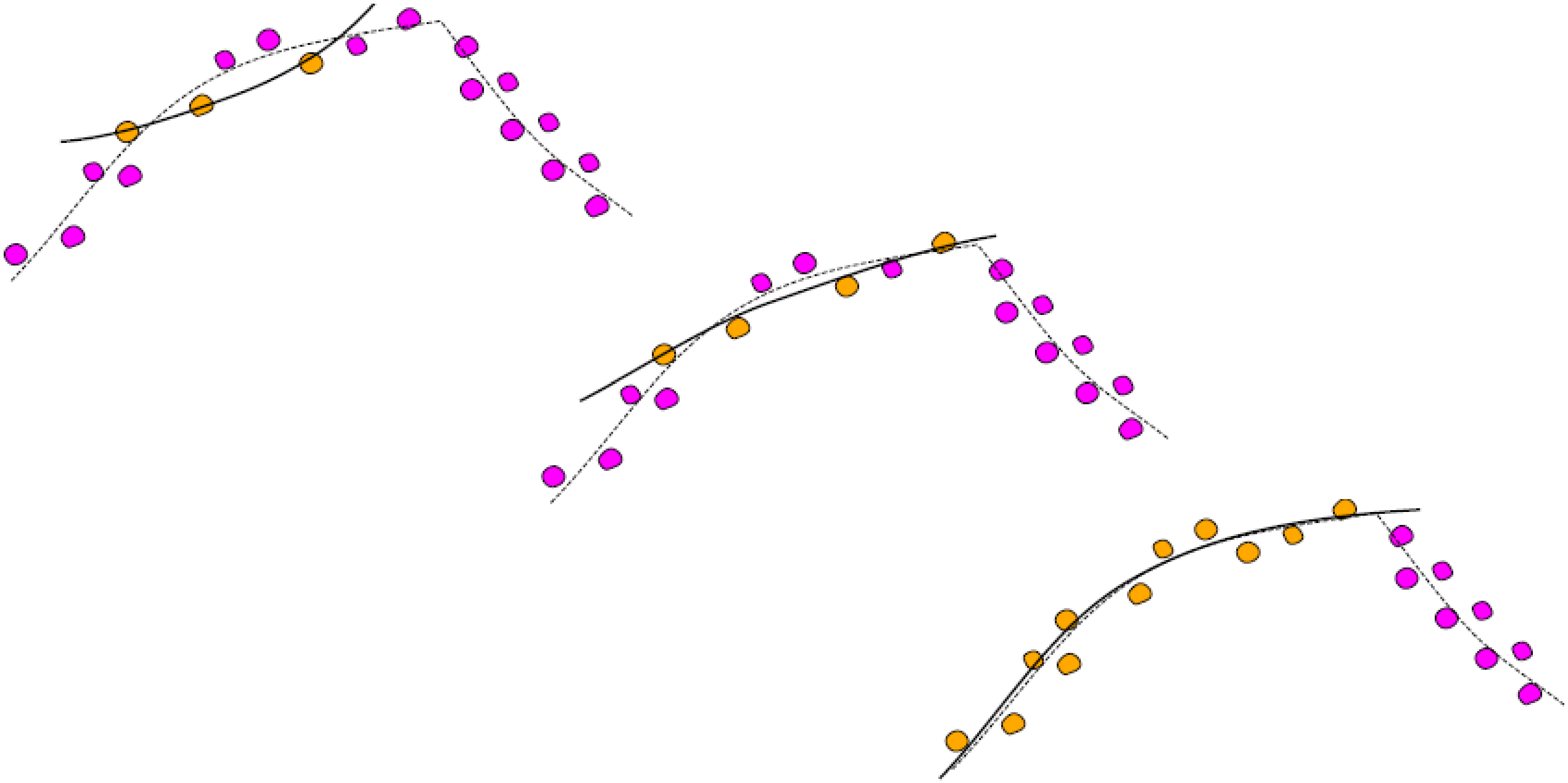
- Define a set of candidates using Q_{i-1} and its inmediate neighbours.
- Set the model E_i (i.e., compute a new second degree polynomial) using least squares with the samples on Q_{i-1} and the initial plane π.
- If the $i - th$ order statistic is above the threshold break the loop.
- Else, set Q_i to be the i candidates with smallest residuals.

3) Return (E_{i-1}, Q_{i-1}).

Observation: The plane is fixed!

Surface Fitting with Forward Search

Algorithm 1: Smooth Component Fitting

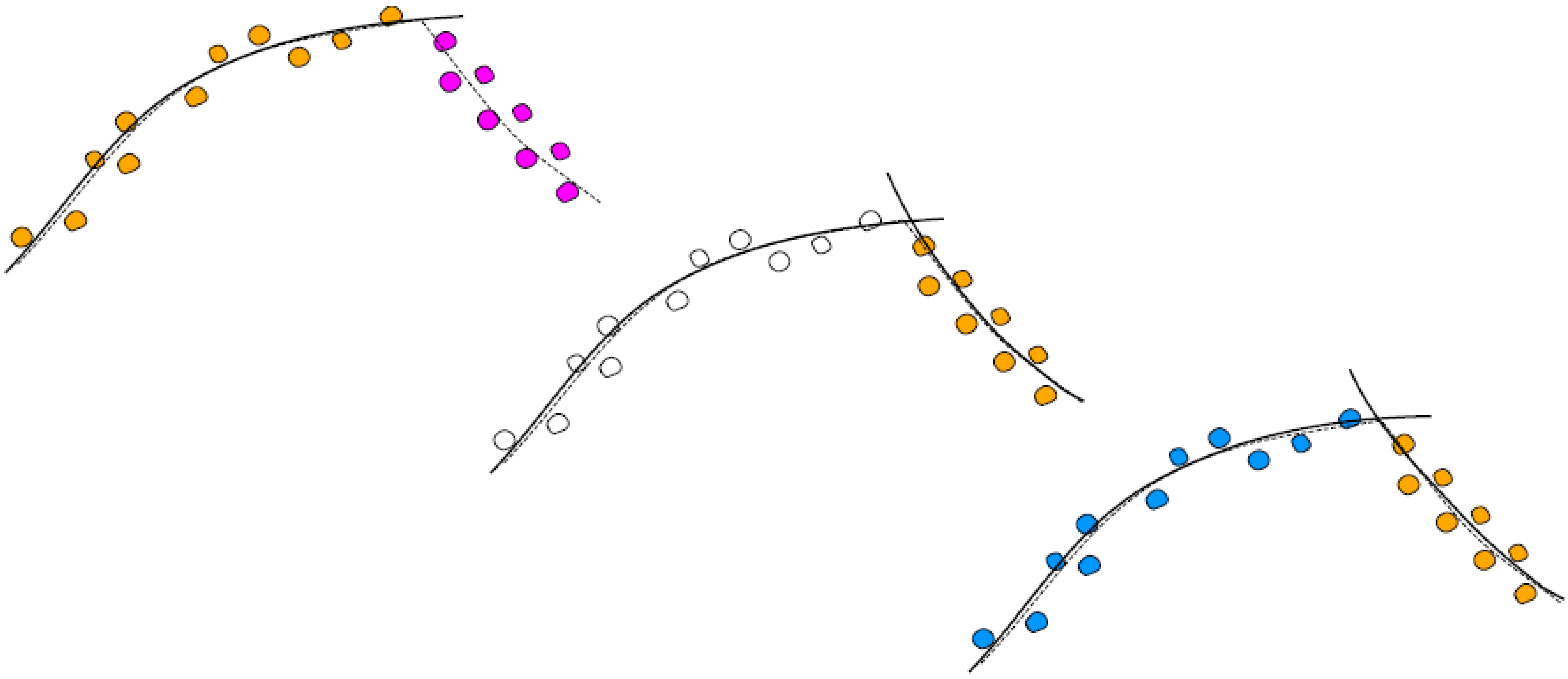


Algorithm 2: Iterative Fitting

1) Apply ***Smooth Component Fitting*** in the current sample set.

2) Remove the fitted points from the sample set.

3) Return to step 1.



Algorithm 3: Point Projection

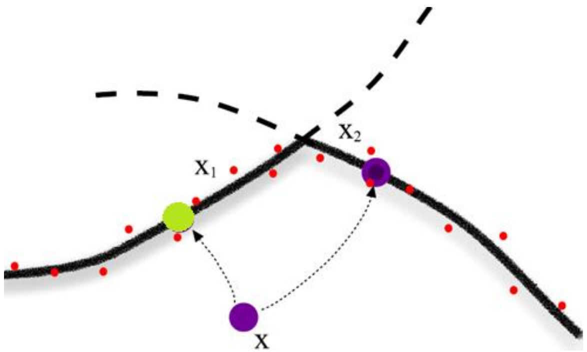
1) Apply ***Iterative Fitting*** in a neighborhood of point x.

2) Project x in each of the smooth components using MLS.

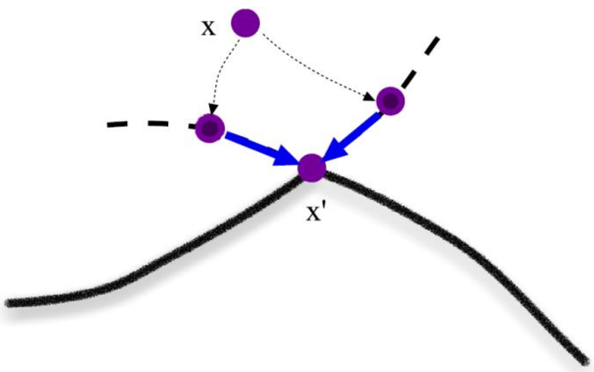
3) Verify on which surfaces the projection is valid

4) Choose the closest valid projection as the "point projection". Else, select the closest point in the surfaces intersection as the "point projection".

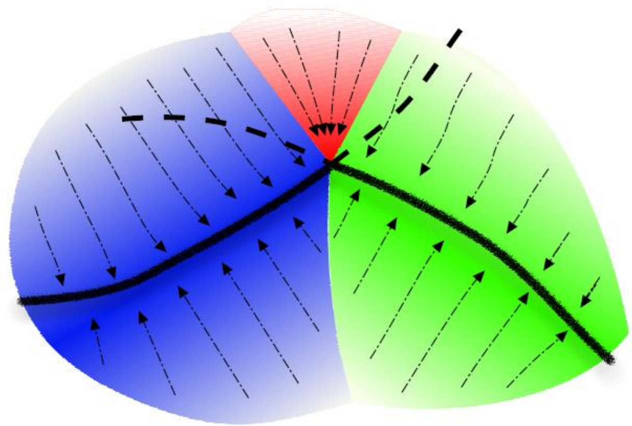
Case I



Case II



Projection Map



Projection Validation

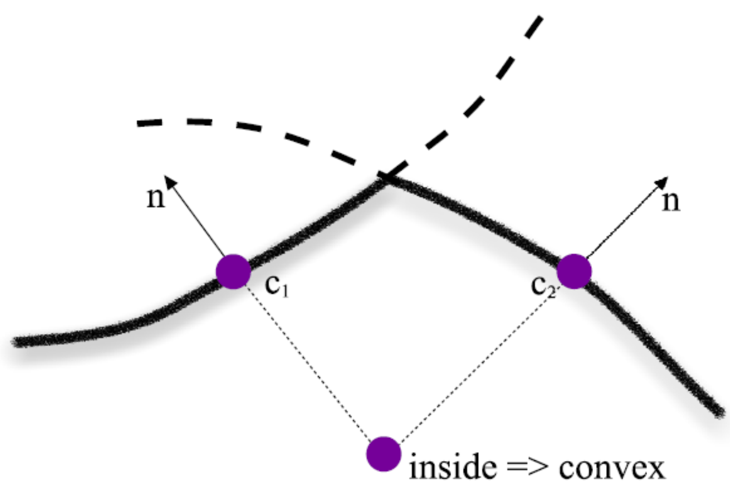
Idea : A surface should be contained in one side of the other.

The validation test described in the paper is not clear. It uses:

- Approximate (exterior?) normals to the surface.
- The ability of classify points as interior or exterior to the object.

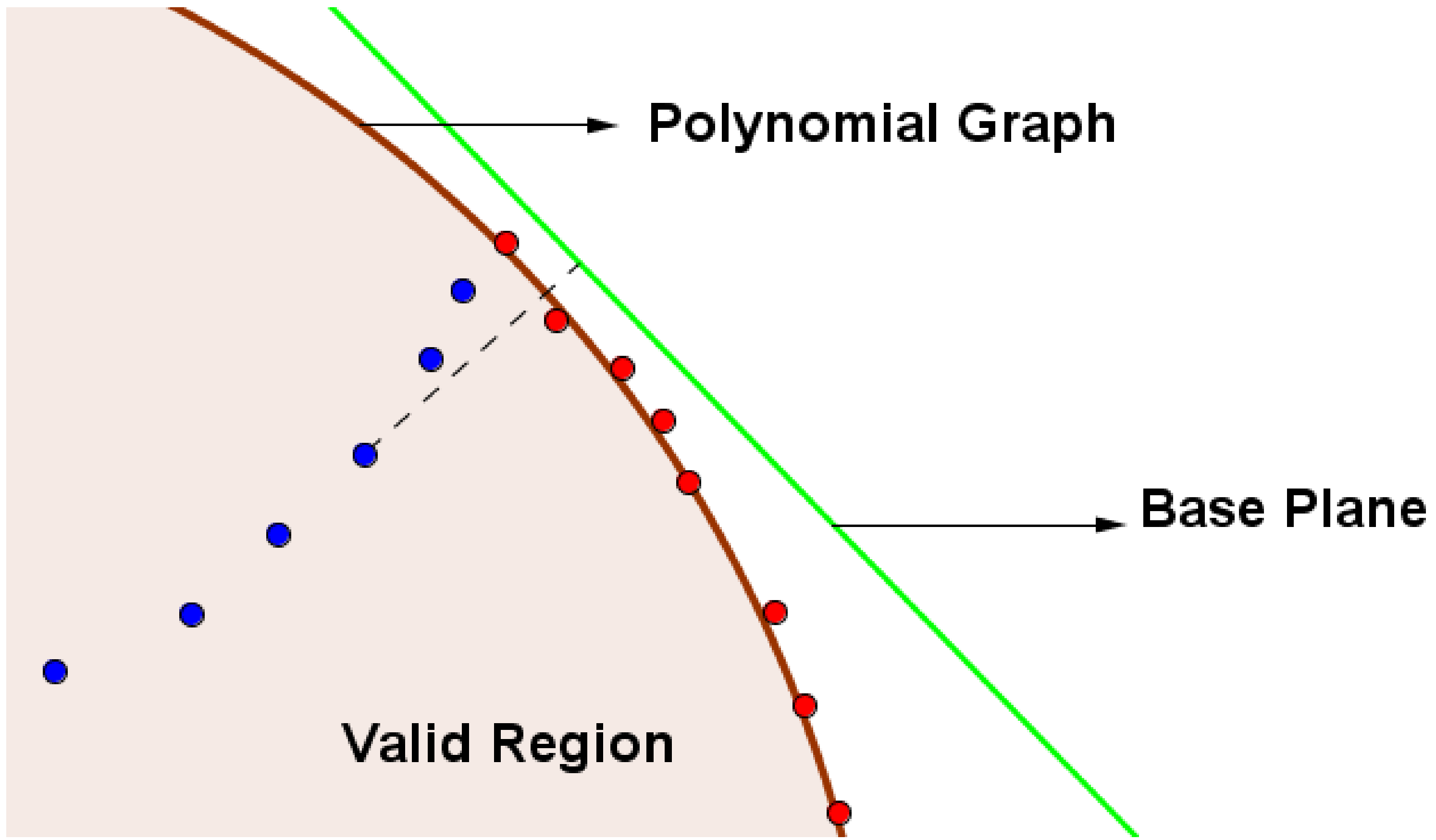
Procedure:

1) Find the closest point O to the normals.
2) Classify O interior or exterior to the surface S_2.
3) Check wheter the projection of x in the surface S_1, (i.e x_1) is in the same side of S_2 than O.
4) If they are in the same side and O is interior, or they are in opposite sides and O is exterior, then the projection is valid.



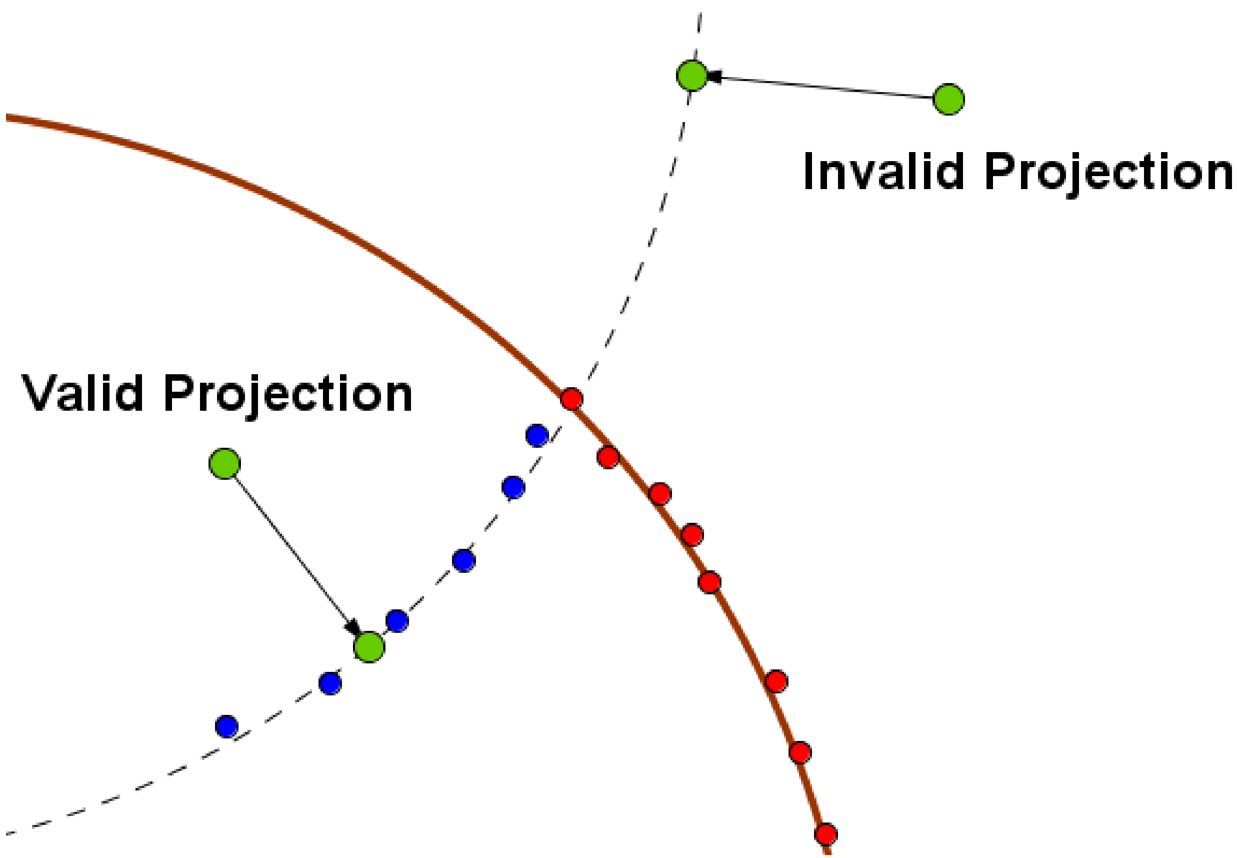
Projection Validation

Idea : A surface should be contained in one side of the other.



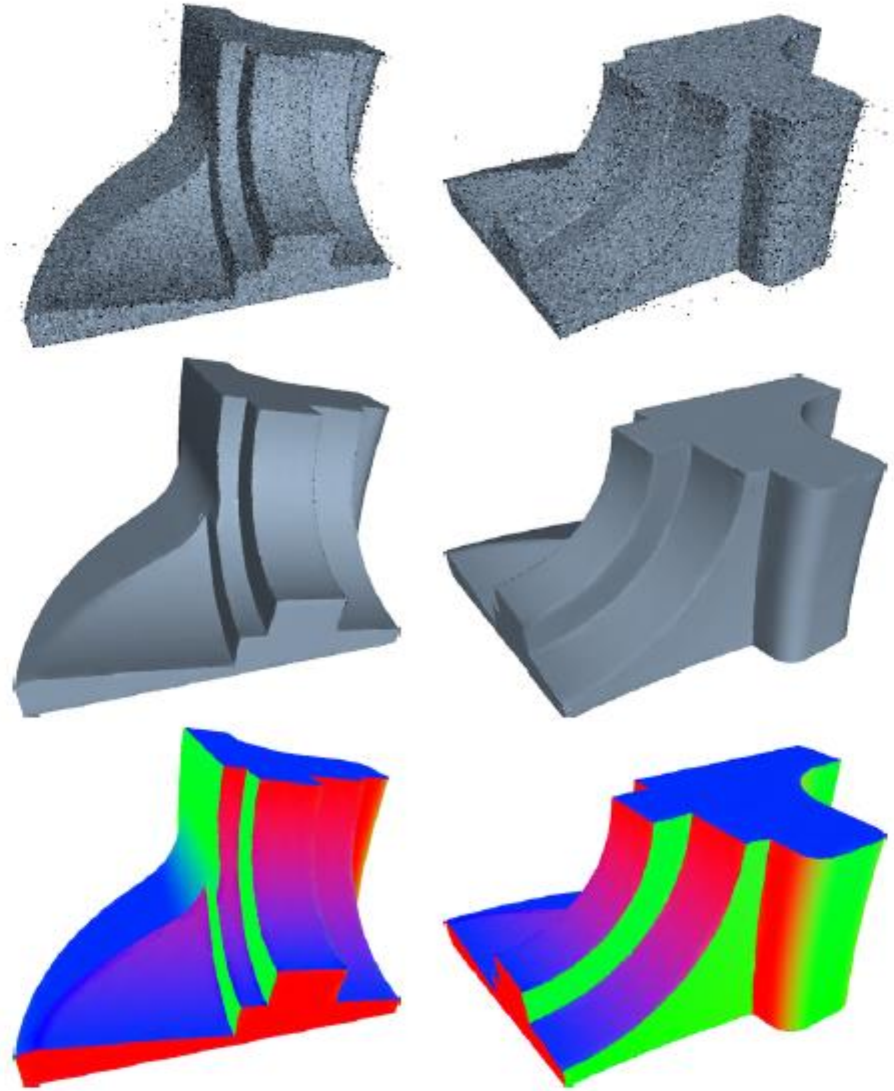
Projection Validation

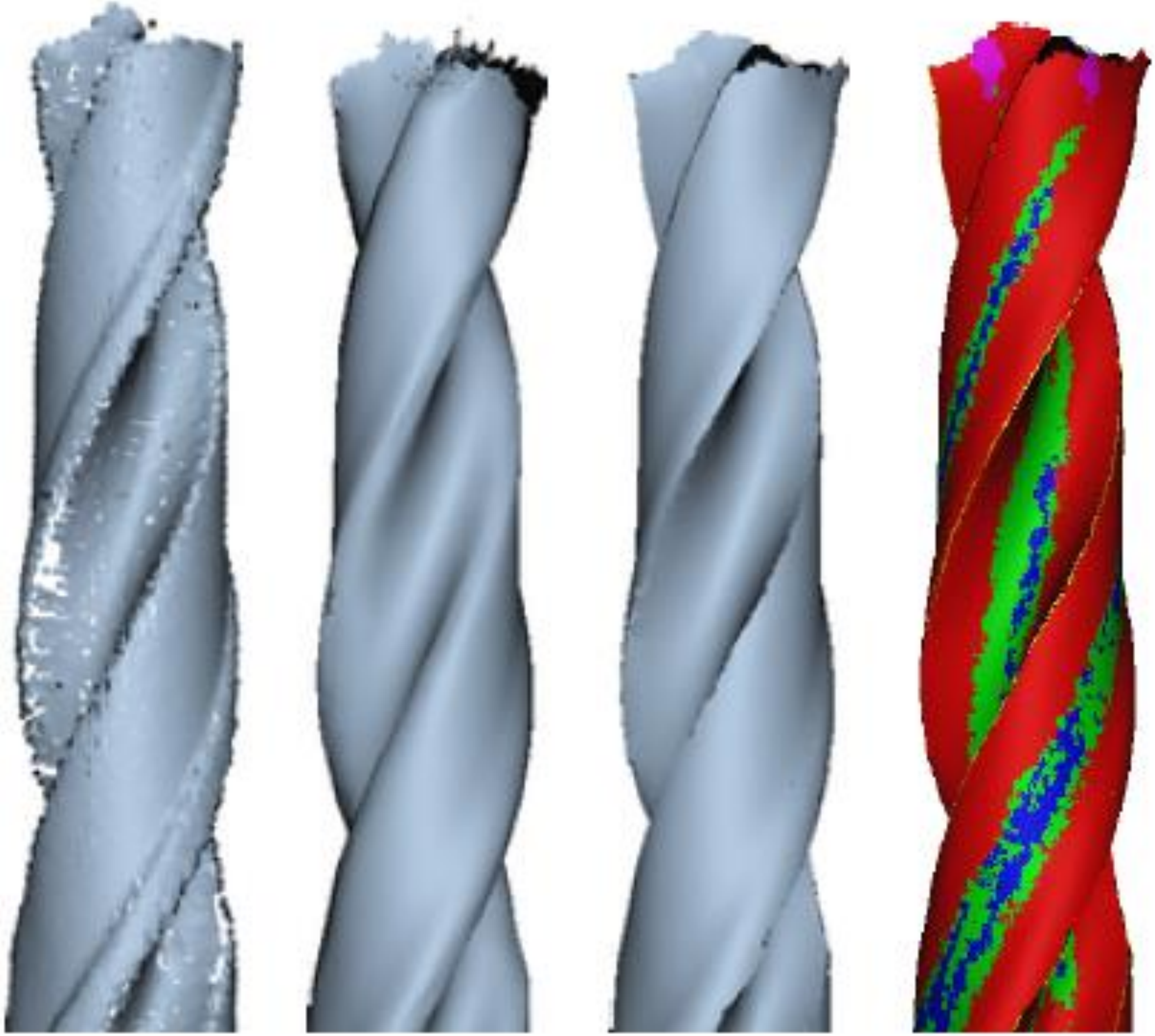
Idea : A surface should be contained in one side of the other.



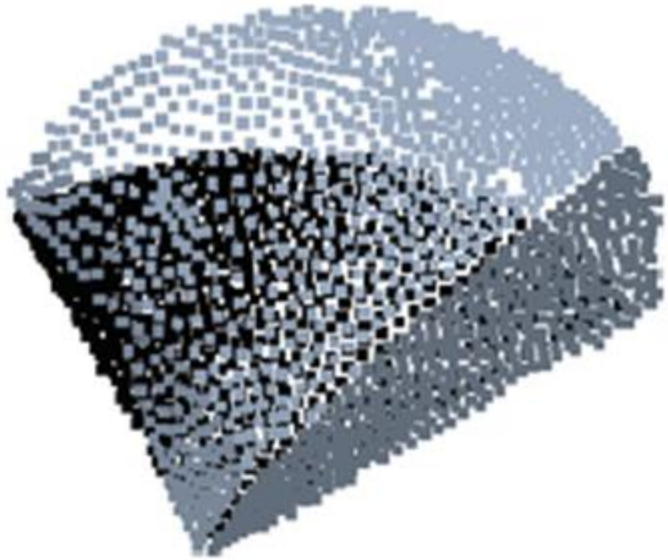
Results

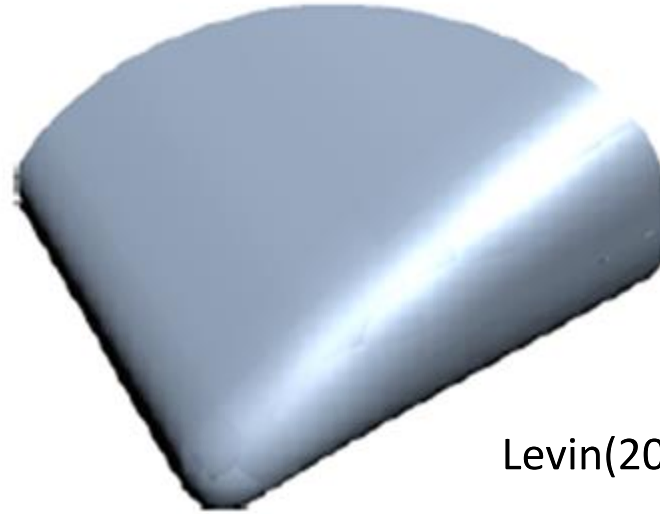
Noisy Sample Set



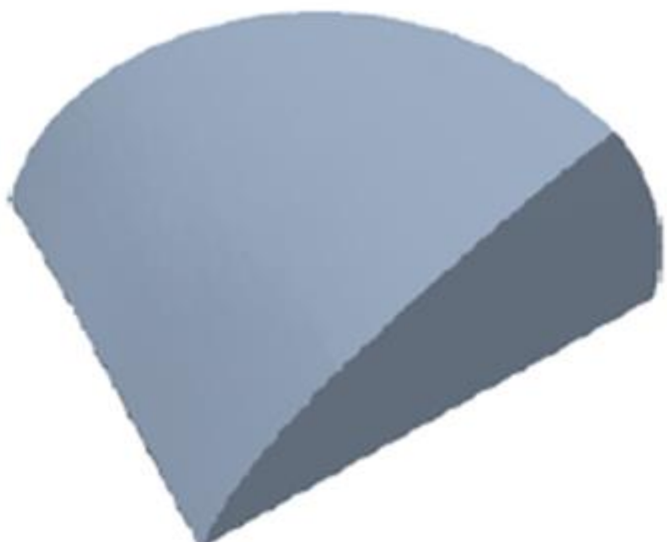


Levin(2003) Fleishman(2005)





Levin(2003)



Fleishman(2005)