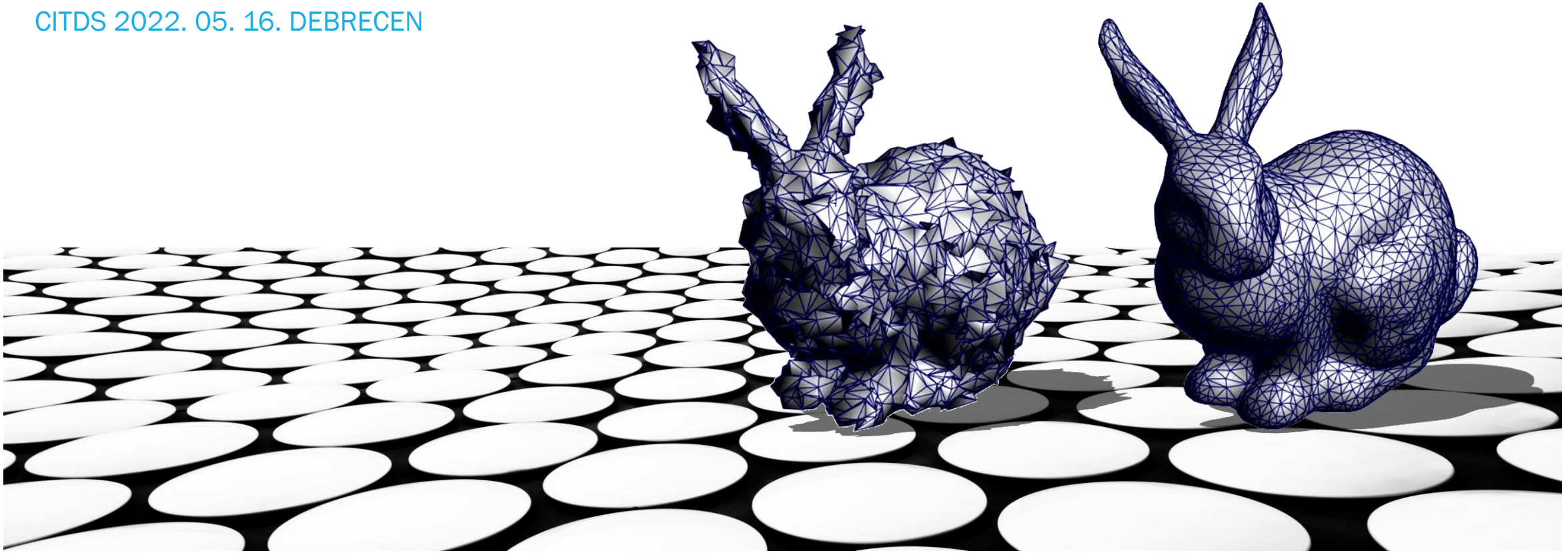
GENERALIZED SAVITZKY—GOLAY FILTER FOR SMOOTHING TRIANGULAR MESHES

GÁBOR FÁBIÁN // ELTE-FI // DEPARTMENT OF NUMERICAL ANALYSIS // BUDAPEST

CITDS 2022. 05. 16. DEBRECEN



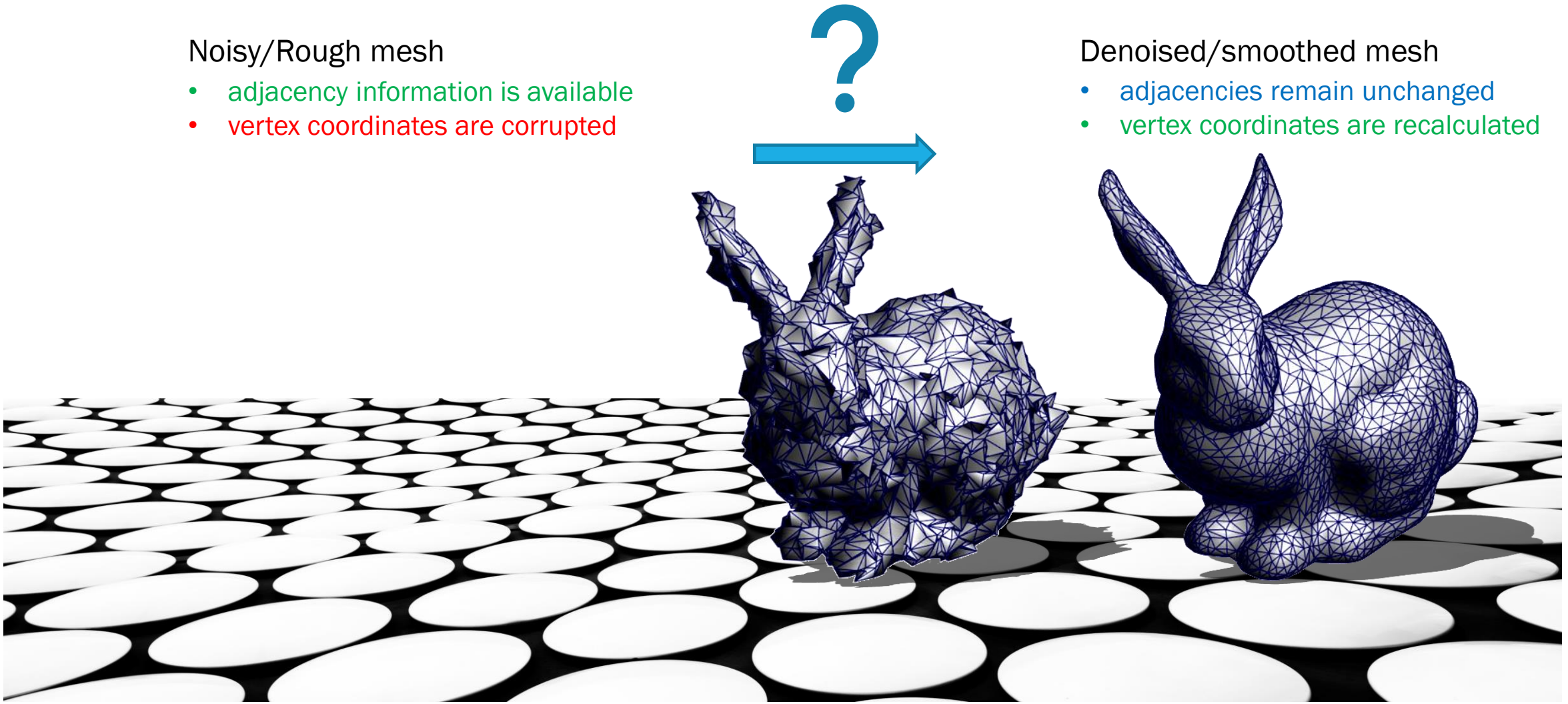
MESH SMOOTHING

Noisy/Rough mesh

- adjacency information is available
- vertex coordinates are corrupted

Denoised/smoothed mesh

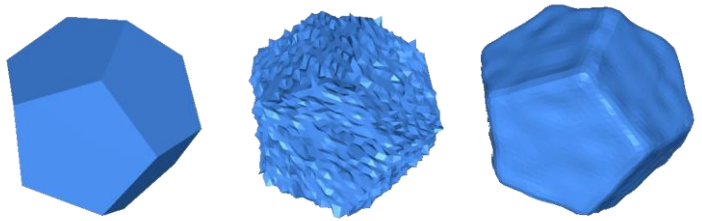
- adjacencies remain unchanged
- vertex coordinates are recalculated



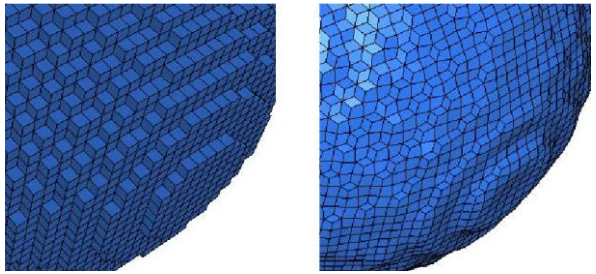
MESH SMOOTHING PROBLEMS

Common use-cases of mesh smoothing algorithms

- Mesh denoising
  - Surface reconstruction of point clouds (3D Scanning: e.g. LiDAR)
  - Biomedical imaging (e.g. CT)

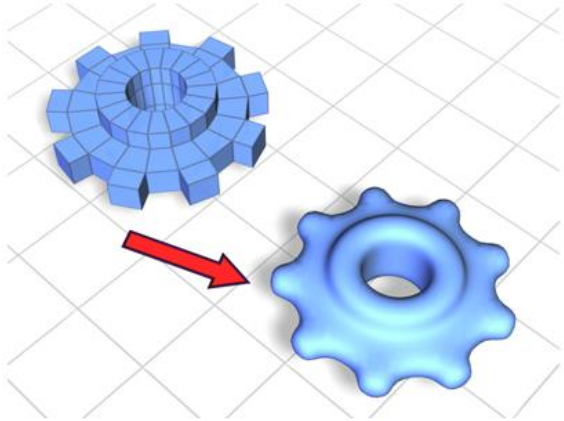


https://www.mdpi.com/1424-8220/20/11/3206/htm



https://www.semanticscholar.org/paper/On-Smoothing-Surfaces-in-Voxel-Based-Finite-Element-Arbenz-Flaig/91615584b1b4c67178a2d6fb09713f0a33da6ef0/figure/0

- Mesh smoothing
  - CAD: Smoother surface without subdivision
  - Analysis of vertex neighborhoods
    - 1st derivative ~ surface normal
    - 2nd derivative ~ surface curvature



https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/3DSMax-Modifiers/files/GUID-EA8FF838-B197-4565-9A85-71CE93DA4F68-htm.html

MESH SMOOTHING ALGORITHMS

Known algorithms for mesh smoothing

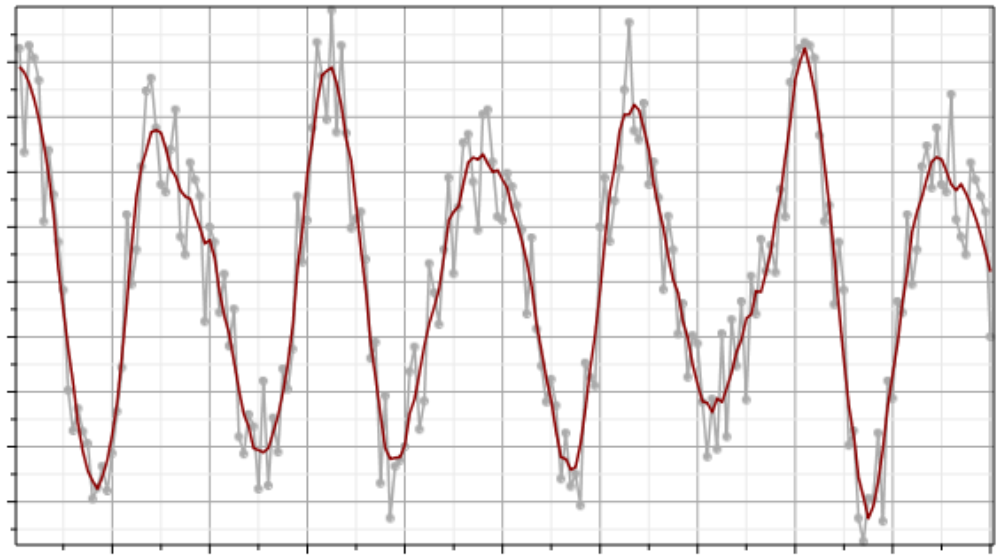
- Popular algorithms
  - Laplacian filter (discrete Laplace operator)
  - Taubin's smoothing filter (erosion + dilation)
  - Laplacian/bilaplacian/curvature flows (geometric flows)

- Properties
  - Algorithms are controlled by (hardly explainable) parameters
  - Iterative methods
    - Too few iterations: the mesh remains noisy
    - Too many iterations: the mesh is getting oversmoothed

https://www.sciencedirect.com/science/article/abs/pii/S0010448501000951

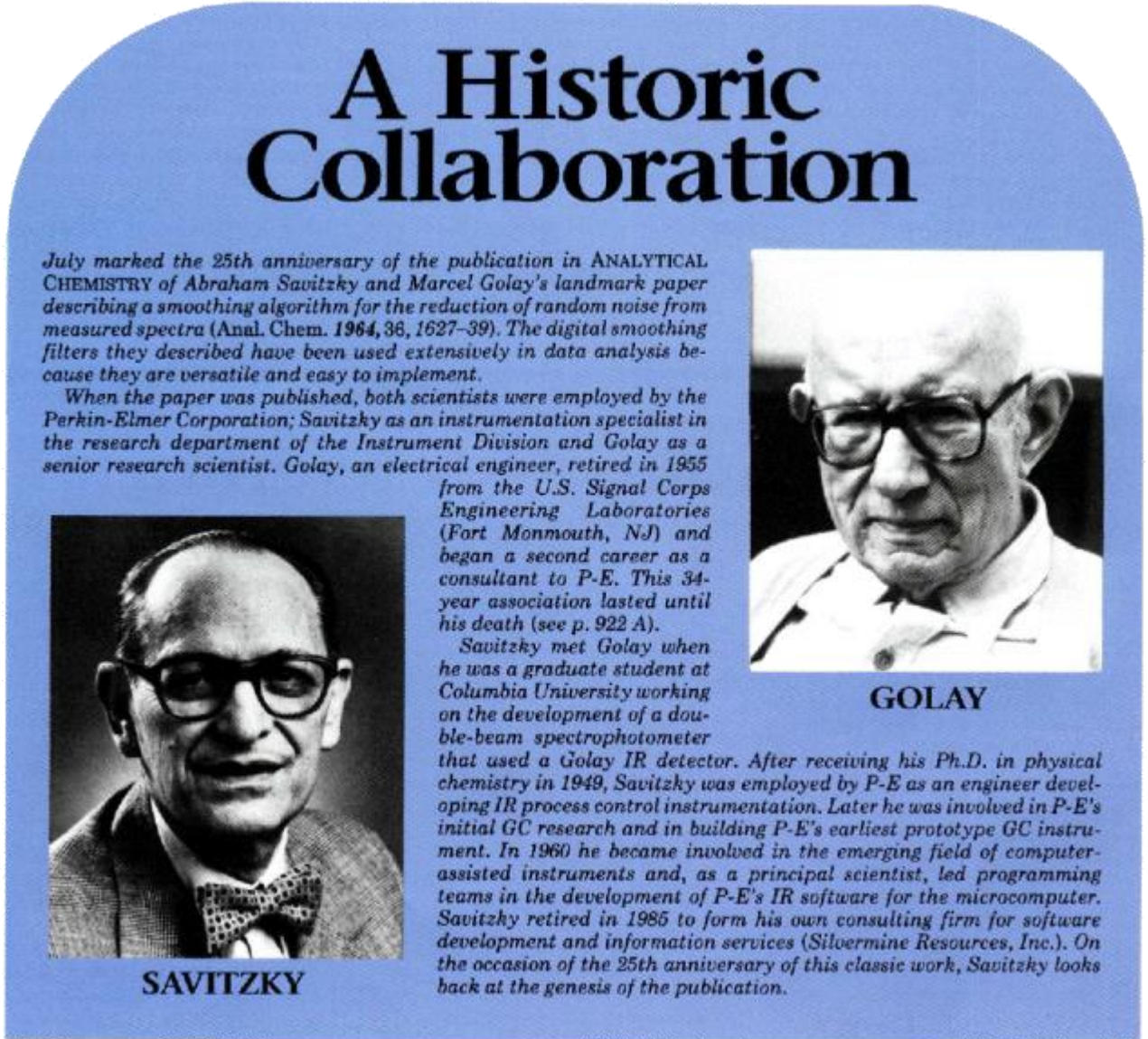
SAVITZKY—GOLAY FILTER

- Abraham Savitzky & Marcel J. E. Golay:
  „Smoothing and differentiation of data by simplified least squares procedures” in Analytical Chemistry 36:8, 1964.
- Denoising 1D functions (signals).
- Estimating derivative of a function.
- Main idea: fitting a polynomial locally to the data points.



https://www.maplesoft.com/Applications/Detail.aspx?id=154593

https://pubs.acs.org/doi/abs/10.1021/ac00190a744



A Historic Collaboration

July marked the 25th anniversary of the publication in ANALYTICAL CHEMISTRY *of Abraham Savitzky and Marcel Golay's landmark paper describing a smoothing algorithm for the reduction of random noise from measured spectra* (Anal. Chem. *1964,* 36, *1627–39). The digital smoothing filters they described have been used extensively in data analysis because they are versatile and easy to implement.*

When the paper was published, both scientists were employed by the Perkin-Elmer Corporation; Savitzky as an instrumentation specialist in the research department of the Instrument Division and Golay as a senior research scientist. Golay, an electrical engineer, retired in 1955 from the U.S. Signal Corps Engineering Laboratories (Fort Monmouth, NJ) and began a second career as a consultant to P-E. This 34-year association lasted until his death (see p. 922 A).

Savitzky met Golay when he was a graduate student at Columbia University working on the development of a double-beam spectrophotometer that used a Golay IR detector. After receiving his Ph.D. in physical chemistry in 1949, Savitzky was employed by P-E as an engineer developing IR process control instrumentation. Later he was involved in P-E's initial GC research and in building P-E's earliest prototype GC instrument. In 1960 he became involved in the emerging field of computer-assisted instruments and, as a principal scientist, led programming teams in the development of P-E's IR software for the microcomputer. Savitzky retired in 1985 to form his own consulting firm for software development and information services (Silvermine Resources, Inc.). On the occasion of the 25th anniversary of this classic work, Savitzky looks back at the genesis of the publication.

GOLAY

SAVITZKY

SAVITZKY—GOLAY FILTER DETAILED

- Discrete data points

$$v_{i-k}, \ldots, v_i, \ldots, v_{i+k} \qquad (k \le i \le n-k)$$

- Low degree polynomial

$$p(x) := \sum_{j=0}^{d} c_j x^j$$

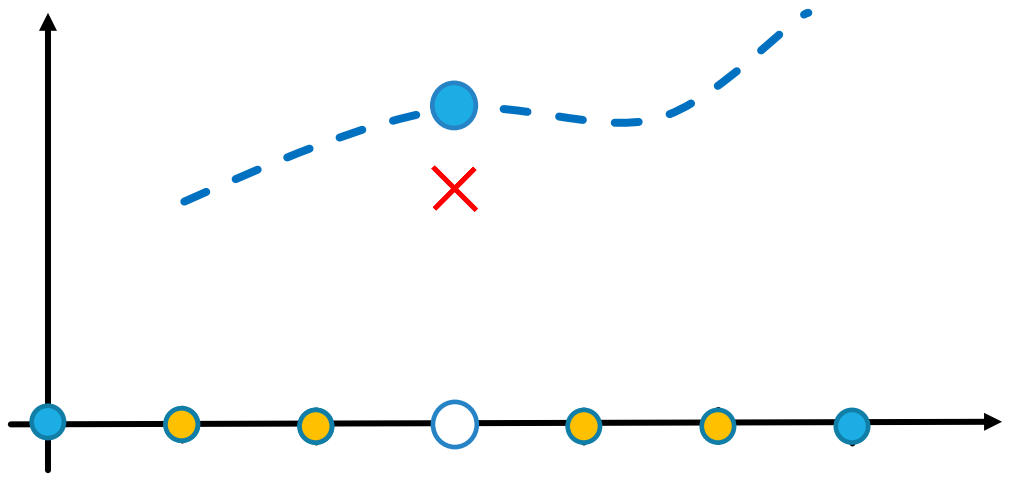
- Minimize the quadratic error

$$\sum_{\ell=-k}^{k} \Big(p(\ell) - v_{i+\ell} \Big)^2 = \sum_{\ell=-k}^{k} \left(\sum_{j=0}^{d} c_j \ell^j - v_{i+\ell} \right)^2$$

- Recalculate the value

$$\widetilde{v}_i := p(0) = c_0$$

Smoothed value



$$(i-k,\ v_{i-k}),\ \ldots\ ,(i+k,\ v_{i+k}) \in \mathbb{R}^2$$

$$B := \begin{bmatrix} 1 & -k & \ldots & (-k)^d \\ 1 & -k+1 & \ldots & (-k+1)^d \\ \vdots & \vdots & & \vdots \\ 1 & k & \ldots & k^d \end{bmatrix} \quad v := \begin{bmatrix} v_{i-k} \\ \vdots \\ v_{i+k} \end{bmatrix} \quad c := \begin{bmatrix} c_0 \\ \vdots \\ c_d \end{bmatrix}$$

$$\|Bc - v\|_2^2 \longrightarrow \min \iff B^T B c = B^T v$$

Gaussian normal equations

SAVITZKY—GOLAY FILTER ON IRREGULAR GRID

- Discrete data points

$$v_{i-k}, \ldots, v_i, \ldots, v_{i+k} \qquad (k \le i \le n-k)$$

- Low degree polynomial

$$p(x) := \sum_{j=0}^{d} c_j (x - x_i)^j$$

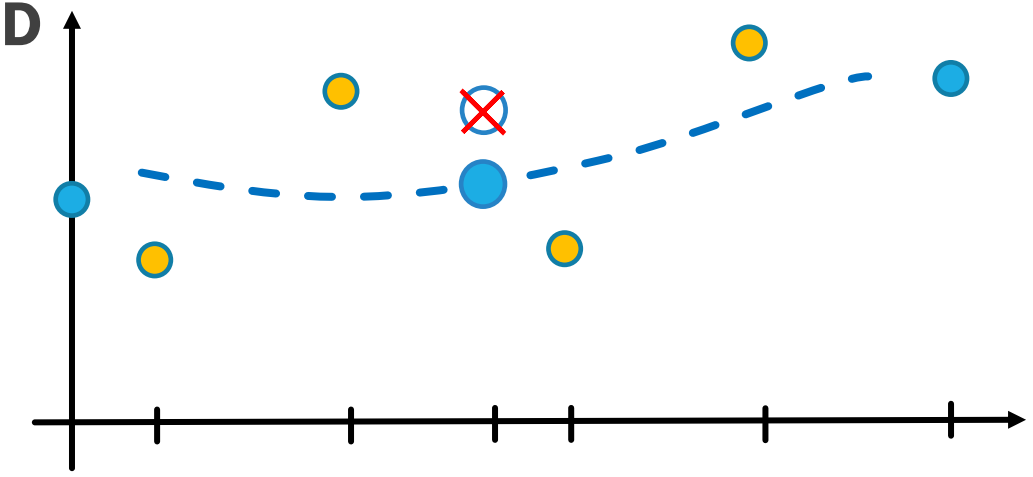
- Minimize the quadratic error

$$\sum_{\ell=-k}^{k} \Big(p(x_{i+\ell}) - v_{i+\ell} \Big)^2 = \sum_{\ell=-k}^{k} \left(\sum_{j=0}^{d} c_j (x_{i+\ell} - x_i)^j - v_{i+\ell} \right)^2$$

- Recalculate the value (smoothing)

$$\widetilde{v}_i := p(x_i) = c_0$$

Smoothed value



$$(x_{i-k},\ v_{i-k}),\ \ldots\ ,(x_{i+k},\ v_{i+k}) \in \mathbb{R}^2$$

$$B := \begin{bmatrix} 1 & x_{i-k} - x_i & \ldots & (x_{i-k} - x_i)^d \\ 1 & x_{i-k+1} - x_i & \ldots & (x_{i-k+1} - x_i)^d \\ \vdots & \vdots & & \vdots \\ 1 & x_{i+k} - x_i & \ldots & (x_{i+k} - x_i)^d \end{bmatrix} \quad v := \begin{bmatrix} v_{i-k} \\ \vdots \\ v_{i+k} \end{bmatrix} \quad c := \begin{bmatrix} c_0 \\ \vdots \\ c_d \end{bmatrix}$$

$$\|Bc - v\|_2^2 \longrightarrow \min \iff B^T B c = B^T v$$

Gaussian normal equations

TOWARDS GENERALIZATION…

- Original Savitzky—Golay filter works on 1D functions.
  The main steps of the smoothing algorithm are the following.
  - Consider a **neighborhood** of a data point.
  - Fit a **polynomial** to the neighboring data points.
  - The smoothed data point is the **constant term** of the fitted polynomial.

- We are about to generalize the concept to surfaces, that are eventually functions from 2D to 3D.
  - What do we mean by **neighborhood** of a point of a surface?
  - To fit a polynomial a **common domain** is needed.
  - The surface should to be (locally) parameterized.
  - The inverse of an **embedding** is a parameterization.
  - Neighborhoods should to be locally embedded into 2D.

TRIANGULAR MESHES

- Surfaces often represented by triangular meshes in computer graphics.

```
v  0.018427 0.156004 -0.000000
v  0.018427 0.145774 -0.156074
v -0.051700 0.216506 -0.146842
v -0.042075 0.216506  0.000000
v  0.018427 0.156004  0.000000
v  0.018427 0.125000 -0.262331
v  0.007980 0.125000 -0.314855
[...]
f 1 3 4
f 1 4 5
f 1 2 3
f 4 3 10
f 4 10 11
f 11 10 20
f 11 20 21
f 20 35 34
```

$$V \subset \mathbb{R}^3$$

Vertices

$$I := \{1, 2, \ldots, |V|\}$$

$$T \subset I \times I \times I$$

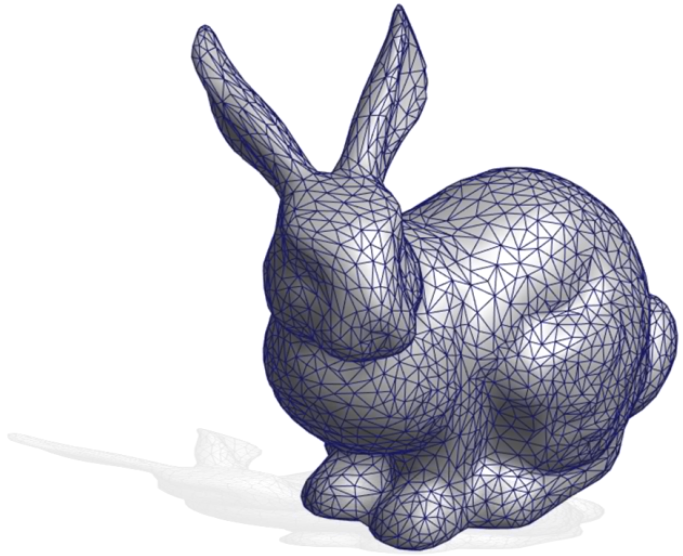
Triangles

$$(V, T)$$

Mesh

$$\mathcal{S}$$

Surface



Triangulation

Topological space

Compact 2-manifold

Each vertex has a neighborhood homeomorphic to the unit disk.

K-NEIGHBORHOOD OF A VERTEX

$$V_1(v) := \{v' \in V \mid \exists t \in T \ : \ (v, v') \text{ is an edge of } t\}$$

$$T_1(v) := \{t \in T \mid v \text{ is a vertex of } t\}$$

$$(V_1(v), T_1(v)) \qquad \mathcal{N}_1(v) \subset \mathbb{R}^3$$

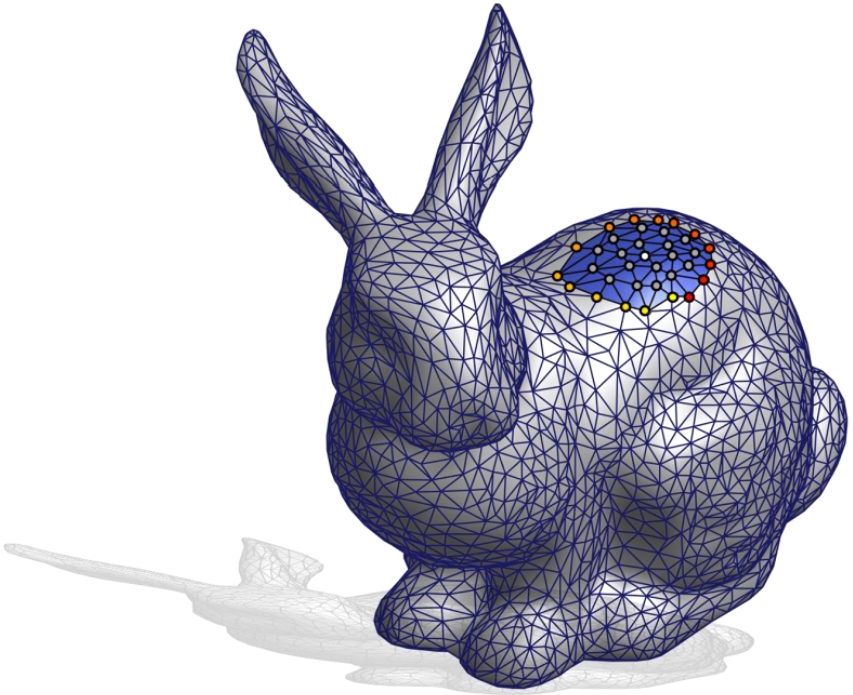
Mesh → Surface

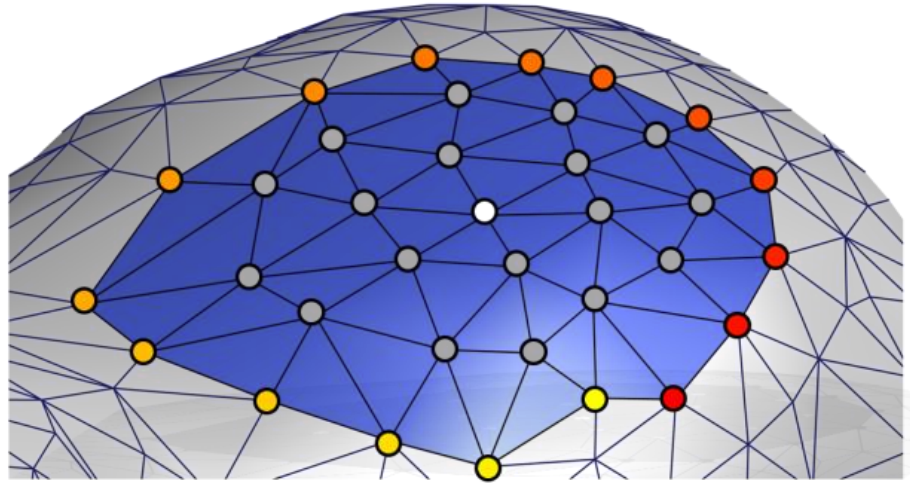
$$V_k(v) := \bigcup_{v' \in V_{k-1}(v)} V_1(v')$$

$$T_k(v) := \bigcup_{v' \in V_{k-1}(v)} T_1(v')$$

$$(V_k(v), T_k(v)) \qquad \mathcal{N}_k(v) := \bigcup_{v' \in V_{k-1}(v)} \mathcal{N}_1(v')$$

Mesh → Surface





LOCAL EMBEDDINGS OF 1-NEIGHBORHOODS

- We are looking for a map that sends the points of a 1-neighborhood to the unit disk.

$$f : \mathcal{N}_1(v) \to \mathbb{D} \qquad \overline{\mathbb{D}} := \{x \in \mathbb{R}^2 \quad : \quad \|x\|_2 \leq 1\}$$

- The constuction is quite easy.

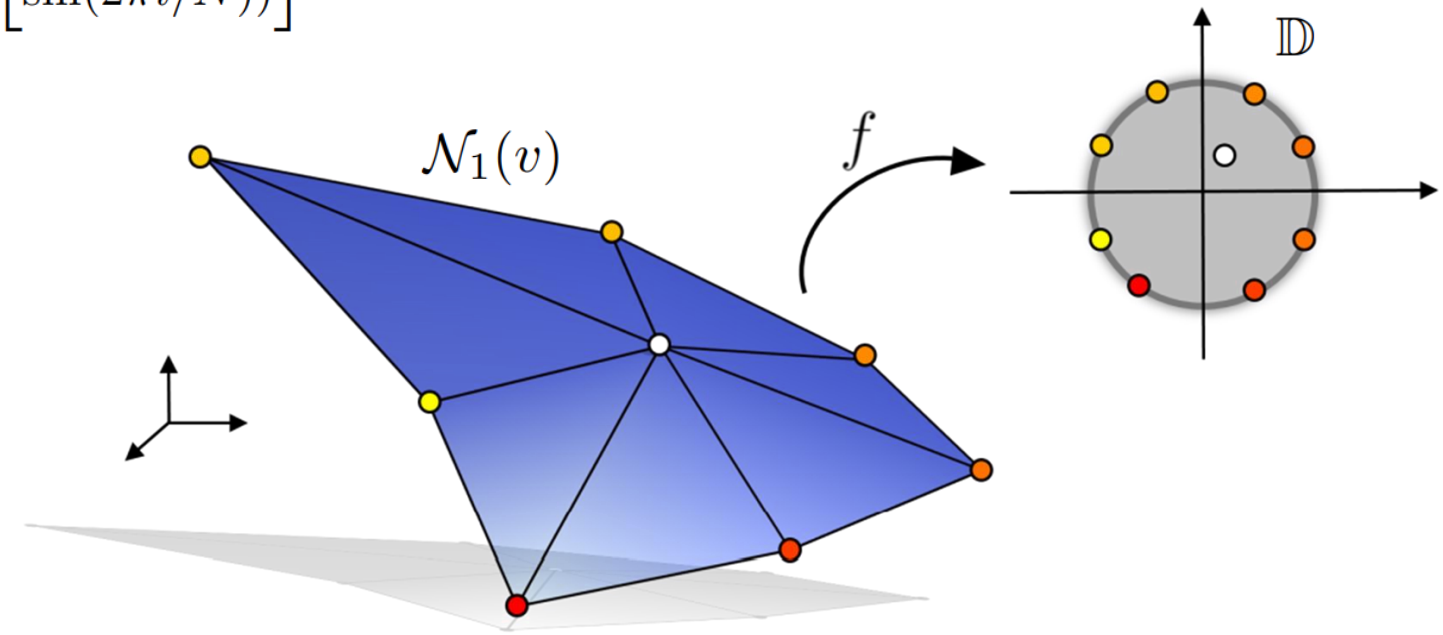
$$f(v_1) := \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad f(v_{i+2}) := \begin{bmatrix} \cos(2\pi i/N) \\ \sin(2\pi i/N)) \end{bmatrix} \quad (i = 0, \ldots, N-1)$$

$$u_i := f(v_i)$$

$$U_1(v) := \{u_1, \ldots, u_{N+1}\}$$

$$(U_1(v), T_1(v))$$

Triangulation of the unit disk



LOCAL EMBEDDINGS OF K-NEIGHBORHOODS

- We are looking for a map that sends the point of a k-neighborhood to the unit disk. It is not so easy as before.
- Vertex coordinates assumed to be the barycentric combination of the coordinates of its neighbors' coordinates.

$$f(v_i) = \sum_{v_j \in V_1(v_i)} \alpha_{ij} f(v_j) \qquad (i = 1, \ldots, n)$$

$$d_i = |V_1(v_i)|$$

Degree of a vertex

inner vertices

outer vertices

$$V_k(v) := \{ \underbrace{v_1}_{\text{central vertex}}, v_2, \ldots, v_n, v_{n+1}, \ldots, v_N \}$$

$$f(v_i) = \sum_{v_j \in V_1(v_i)} \frac{1}{d_i} f(v_j) = \frac{1}{d_i} \sum_{v_j \in V_1(v_i)} f(v_j)$$

Tutte's embedding

$$f(v_{n+1+i}) := \begin{bmatrix} \cos(2\pi i/N) \\ \sin(2\pi i/N)) \end{bmatrix} \qquad (i = 0, \ldots, N-1)$$

Send outer vertices to the unit circle

TUTTE'S EMBEDDING THEOREM

- Adjacency matrix

$$(A)_{ij} = \begin{cases} 1 & i \neq j \text{ and } v_j \in V_1(v_i) \\ 0 & \text{otherwise,} \end{cases}$$

$$A = \left[\begin{array}{c|c} A_{\text{in}} & A_{\text{out}} \\ \hline A_{\text{out}} & \cdots \end{array}\right]$$

- Degree matrix

$$(D)_{ij} = \begin{cases} 0 & i \neq j \\ |V_1(v_i)| & i = j \end{cases}$$

$$D = \left[\begin{array}{c|c} D_{\text{in}} & 0 \\ \hline 0 & \cdots \end{array}\right]$$

- Linear system for inner vertices

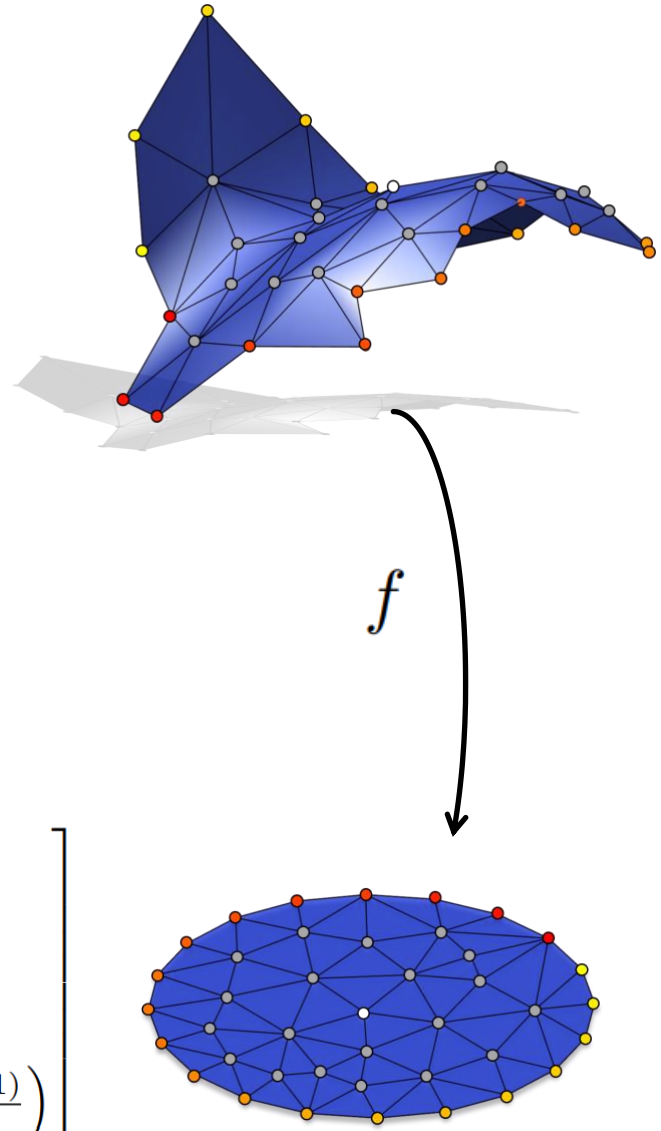
$$(I - D_{\text{in}}^{-1} A_{\text{in}})\, u_{\text{in}} = D_{\text{in}}^{-1} A_{\text{out}}\, u_{\text{out}}$$

$$U_k(v) := \{u_1, \ldots, u_{n+N}\}$$

$$(U_k(v), T_k(v))$$

Triangulation of the unit disk

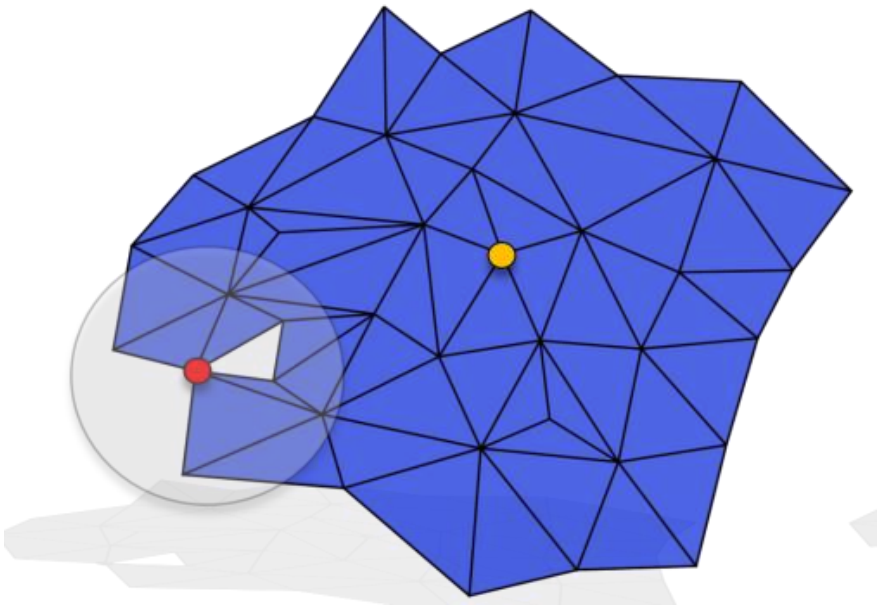
$$u_{\text{out}} := \begin{bmatrix} 1 & 0 \\ \cos\left(\frac{2\pi}{N}\right) & \sin\left(\frac{2\pi}{N}\right) \\ \vdots & \vdots \\ \cos\left(\frac{2\pi(N-1)}{N}\right) & \sin\left(\frac{2\pi(N-1)}{N}\right) \end{bmatrix}$$



DEGENERACIES IN K-NEIGHBORHOODS

But wait… is it always true, that the k-neighborhood of a vertex is homeomorphic to the disk?

- Tutte's embedding theorem works only for meshes that are homeomorphic to the unit disk.
- The 1-neighborhoods are always disk-like surfaces.
- Unfortunately, the k-neighborhoods are not!



- We have to deal with the degeneracies!

DISK-LIKE K-NEIGHBORHOODS

- In some cases the neighborhood of a vertex is not homeomorphic to the unit disk.
- Boundary vertices should have a degree of 2.
- The degree of degenerate vertices is greater than 2!
- Removing degenerate vertices is required.

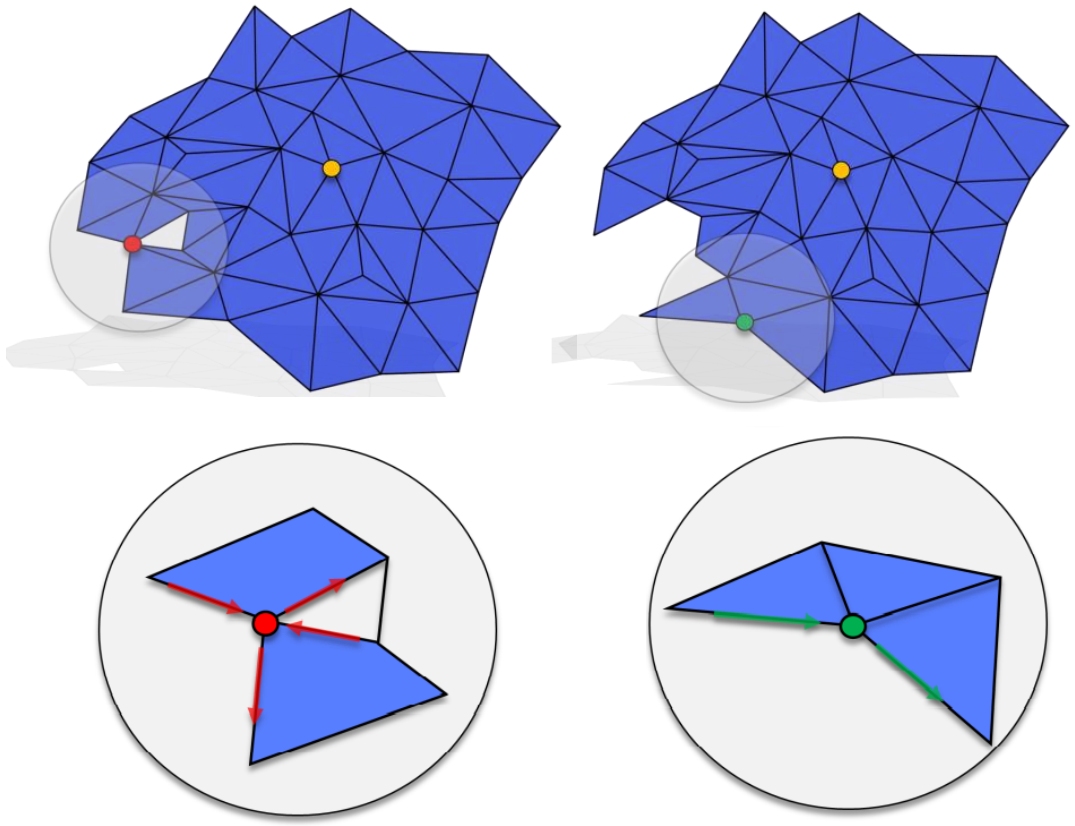
$$X_k(v) := \{w \in V_k(v) \mid w = v_i,\ i \geq n+1,\ d_i > 2\}$$

$$V_k^\circ(v) := V_k(v) \setminus X_k(v)$$

$$T_k^\circ(v) := T_k(v) \setminus \bigcup_{w \in X_k(v)} T_1(w)$$

$$(V_k^\circ(v), T_k^\circ(v)) \qquad \mathcal{N}_k^\circ(v)$$

Mesh → Disk-like surface

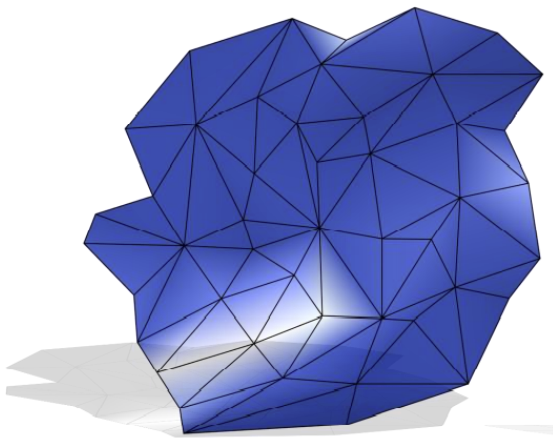


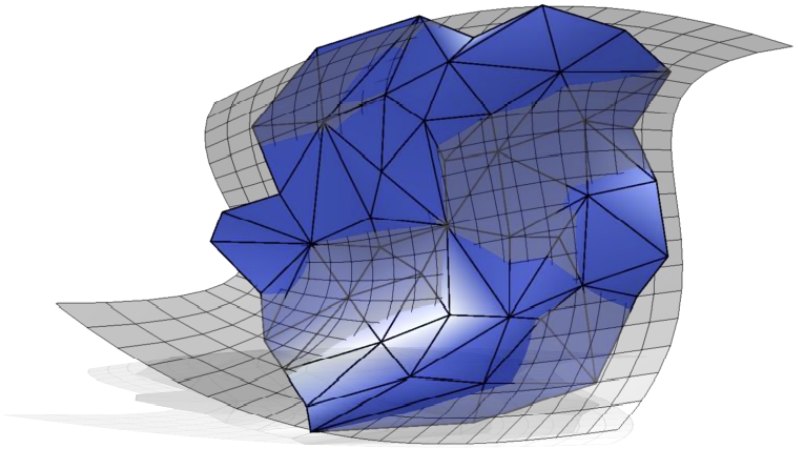
FITTING A POLYNOMIAL

$$\overline{u}_i := (\overline{x_i}, \overline{y_i}) = u_i - u_1 = (x_i - x_1, y_i - y_1)$$

$$p(x, y) = \sum_{j,k=0}^{d} c_{jk}(x - x_1)^j (y - y_1)^k$$

$$p(x_i, y_i) = \sum_{j,k=0}^{d} c_{jk}\overline{x_i}^j \overline{y_i}^k$$





$$B := \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \overline{x}_2 & \overline{y}_2 & \dots & \overline{x}_2^d \overline{y}_2^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \overline{x}_{n+N} & \overline{y}_{n+N} & \dots & \overline{x}_{n+N}^d \overline{y}_{n+N}^d, \end{bmatrix}$$

$$C := \begin{bmatrix} c_{00} \\ c_{10} \\ c_{01} \\ \vdots \\ c_{dd} \end{bmatrix} \quad D := \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n+N} \end{bmatrix}$$

Gaussian normal equations

$$B^T BC = B^T D$$

Smoothed vertex coordinate

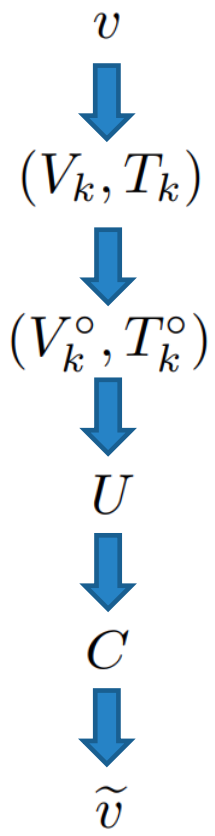
$$\widetilde{v} := \widetilde{v}_1 := p(x_1, y_1) = c_{00}$$

GENERALIZED SAVITZKY—GOLAY FILTER ALGORITHM

- Input parameters:
  - size of the neighborhood,
  - degree of the polynomial to be fitted.

- For all **vertex** of the mesh.
  - Calculate **k-neighborhood**.
  - Remove degeneracies obtaining a **disk-like neighborhood**.
  - Calculate **embedding** (inverse of the parameterization).
  - Fit a **polynomial** to the parameterized (sub)surface.
  - Calculate the **smoothed vertex coordinate**.

- First we calculate the smoothed coordinates of each vertex, only afterwards update their positions.

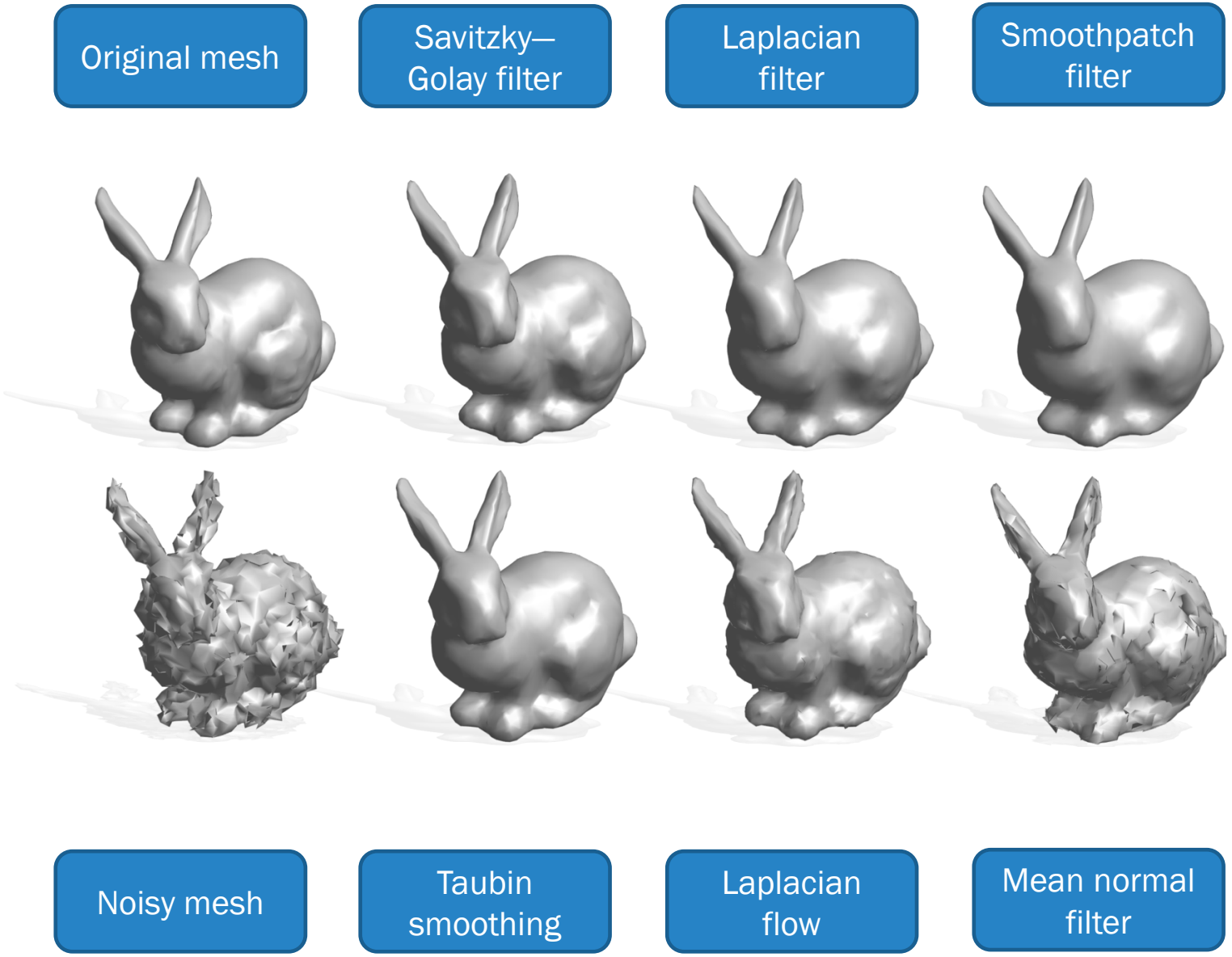


RESULTS

Mesh: Stanford Bunny

Vertices: 2503

Triangles: 5002



RESULTS

Mesh: Ducky

Vertices: 6390

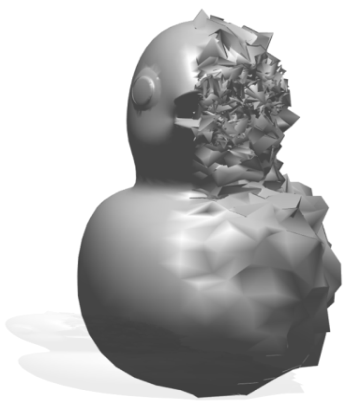
Triangles: 12776

Original mesh

Savitzky–Golay filter

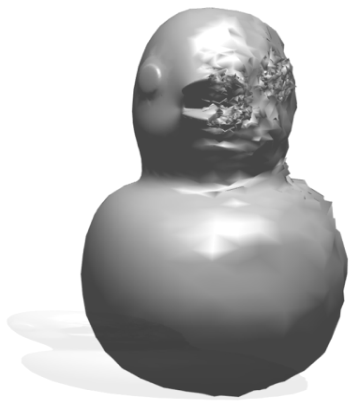
Laplacian filter

Smoothpatch filter

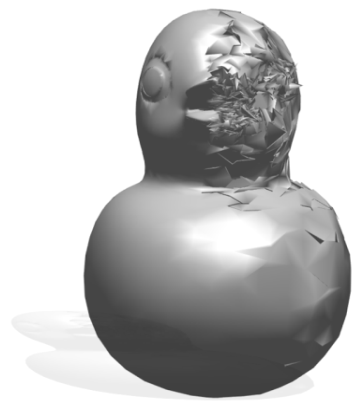


Noisy mesh

Taubin smoothing



Laplacian flow



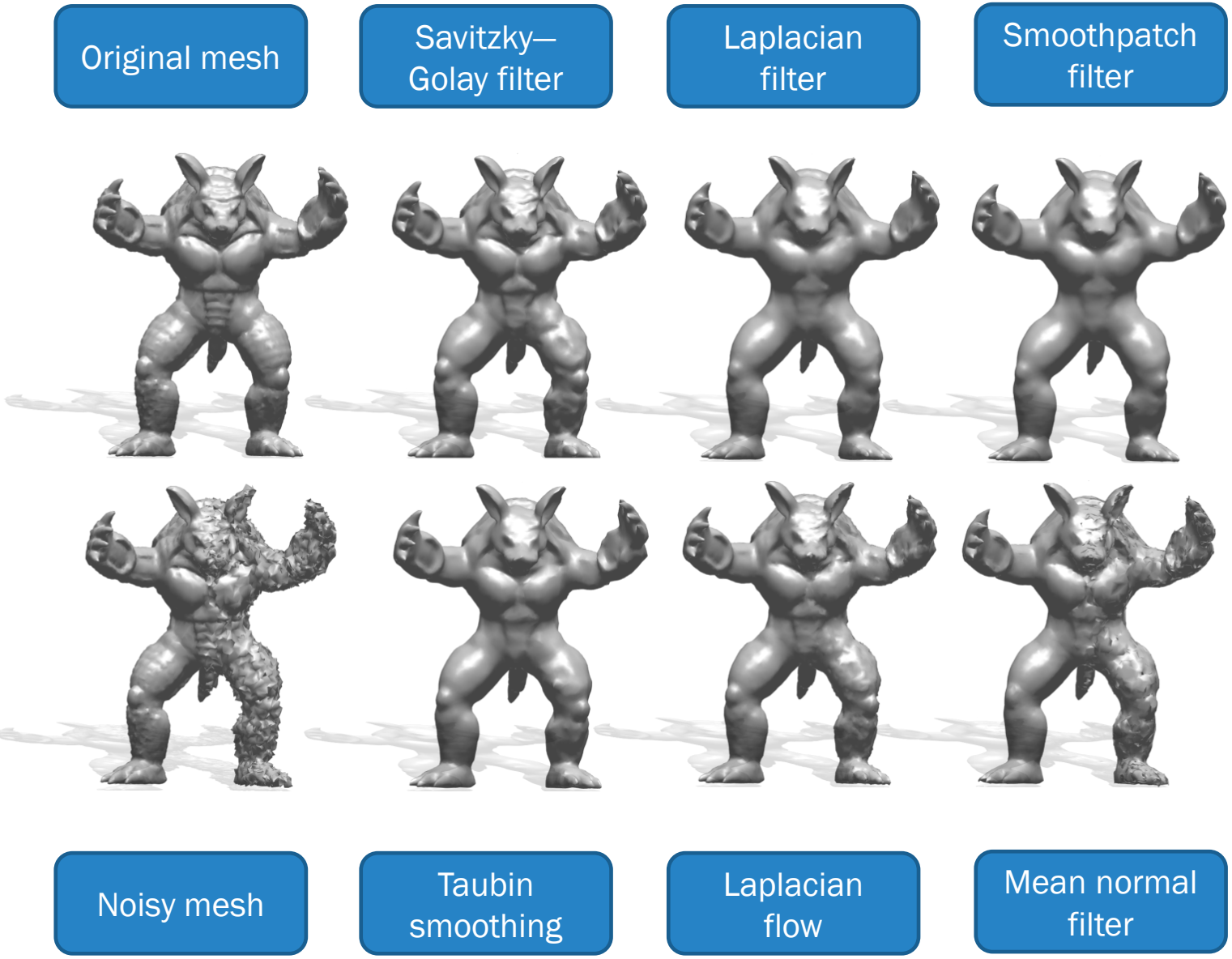
Mean normal filter

RESULTS

Mesh: Armadillo

Vertices: 15002

Triangles: 30000



SUMMARY

- Proposed algorithm is a **proof of concept**.

- Qualitative comparison results are promising.
  - Our filter produced similar (or better) results as the standard mesh smoothing filter.
  - In most cases we used a single smoothing step, while other algorithms often use dozens of smoothing steps.
  - Setting the parameters of our method is relatively easy, generally no fine-tuning required.

- Future plans
  - Effective implementation in a high level programming language.
  - Planning special data structure for efficient generation of neighborhoods.
  - Quantitative comparison based on some standard error metrics.

THANK YOU FOR YOUR ATTENTION

- Savitzky—Golay filter
  - A. Savitzky, M. J. E. Golay, Smoothing and differentiation of data by simplified least squares procedures., Analytical Chemistry 36 (8) (1964)
  - R. Schafer, What is a savitzky-golay filter? [lecture notes], IEEE Signal Processing Magazine - IEEE SIGNAL PROCESS MAG 28 (2011) 111–117.
- Mesh smoothing
  - A. Belyaev, Y. Ohtake, A comparison of mesh smoothing methods, Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics, Tel Aviv University, 83-87 (2003).
  - R. Bade, J. Haase, B. Preim, Comparison of fundamental mesh smoothing algorithms for medical surface models., Vol. 1, 2006, pp. 289–304.
- MATLAB implementations of mesh smoothing algorithms
  - Laplacian smoothing: A. Jacobson, gptoolbox (2022) https://github.com/alecjacobson/gptoolbox
  - Smoothpatch filter: D.-J. Kroon, Smooth triangulated mesh (2022). https://www.mathworks.com/matlabcentral/fileexchange/26710-smooth-triangulated-mesh
  - Taubin's smoothing filter: Manu, Taubin smooth (2022). https://nl.mathworks.com/matlabcentral/fileexchange/59863-taubin-smooth
  - Laplacian flow & Mean normal filtering: J. O. Kaven, Two functions for smoothing/denoising of triangular meshes (2022). https://www.mathworks.com/matlabcentral/fileexchange/24943-two-functions-for-smoothing-denoising-of-triangular-meshes