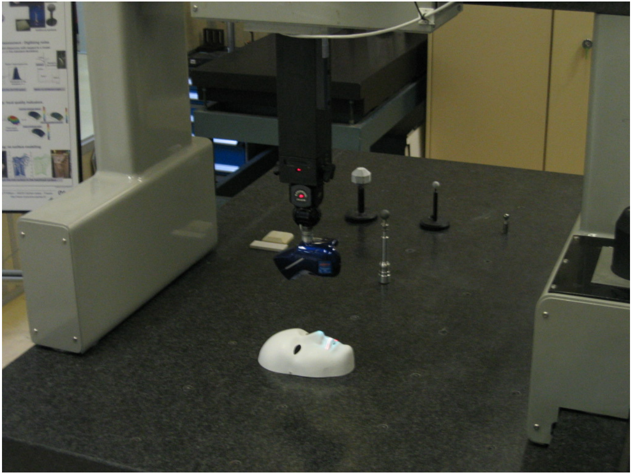
A Scale Space Approach to the Processing of Point Clouds
Master MVA - ENS Cachan

Julie Digne
CMLA - ENS Cachan

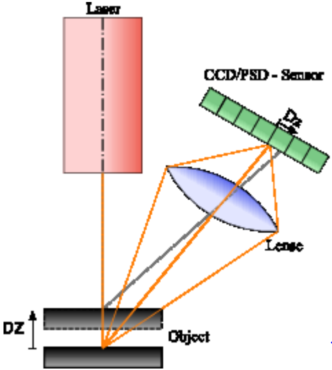
INRIA Sophia Antipolis

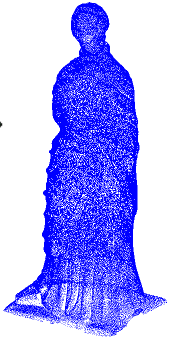
2011/10/14

Introduction: Acquisition of point clouds



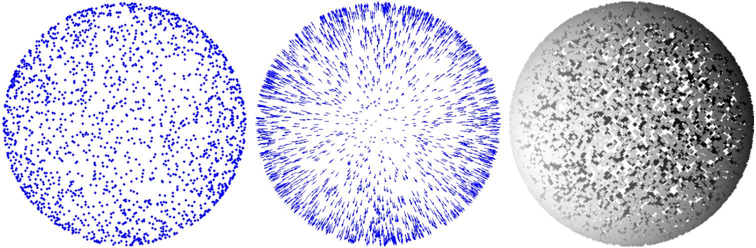
Input data



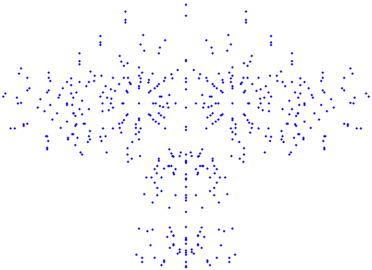


Triangulation laser scanner: triangle formed by the camera optic center, the laser emitter and the impact point. The result given by the scanner is a list of unoriented $3D$ points

3d surfaces typical challenges: Orienting the point set



3d surfaces typical challenges: Building a mesh from a set of points



3d surfaces typical challenges: Registering and merging scans

Outline

Outline

Definition of a surface

- Two-dimensional manifold embedded in a higher dimensional space.
- We will assume the surface to be $\mathcal{C}^2$. For each surface point **p**, there exists a neighborhood V around **p** where the surface is locally a graph $z = g(x, y)$ on the tangent plane.

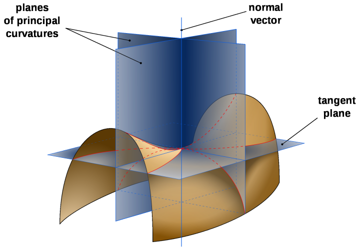


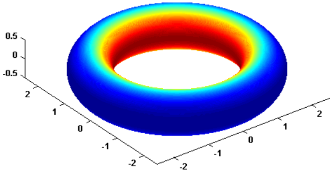
Figure: Curvatures of a surface. (Image by Eric Gaba, from Wikimedia Commons)

- The principal curvatures k_1 and k_2 of a $\mathcal{C}^2$ surface are the eigenvalues of $D^2 g$ such that $k_1 \geq k_2$. $D^2 g$ is the second derivative of g:

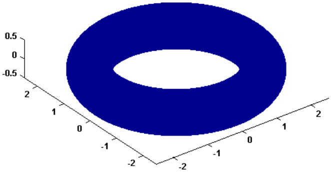
$$D^2 g = \begin{pmatrix} 2a & c \\ c & 2b \end{pmatrix}.$$

- The principal directions $(\vec{t}_1, \vec{t}_2)$ are defined as the eigenvectors associated to the eigenvalues of $D^2 g$
- The *Mean Curvature H* is defined as $H = \frac{1}{2}(k_1 + k_2)$.

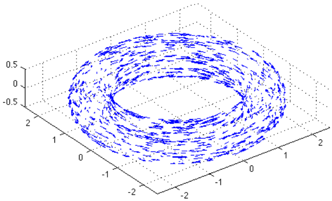
Torus



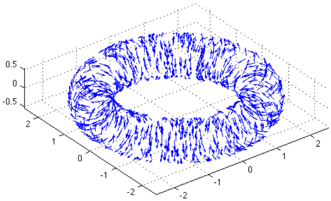
(a) k_1



(b) k_2



(c) t_1



(d) t_2

Intepretation of the principal curvatures

- C_1 and C_2 the planar curves formed by the intersection of the planes $(\mathbf{p}, \vec{t}_1, n)$ and $(\mathbf{p}, \vec{t}_2, n)$ with the surface.
- k_1 and k_2 are the curvature of the curves C_1 and C_2: the inverse of the radius of the osculating circle (up to the sign).

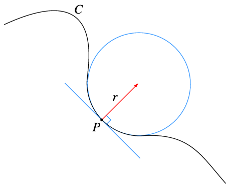


Figure: **Osculating circle of a planar curve. Image from Wikimedia Commons.**

Mean Curvature Motion

The equivalent of the heat equation for 2D images (isotropic diffusion) is the mean curvature motion for surface (MCM). Let S be a surface and $\mathbf{p}$ a point of S with normal $\mathbf{n}$ and mean curvature $H(\mathbf{p})$, then the mean curvature motion writes:

$$\frac{\partial \mathbf{p}}{\partial t} = -H(\mathbf{p})\mathbf{n}(\mathbf{p}) \tag{1}$$

Principal Component Analysis

- finds given a set of variables the directions capturing the highest data variation.

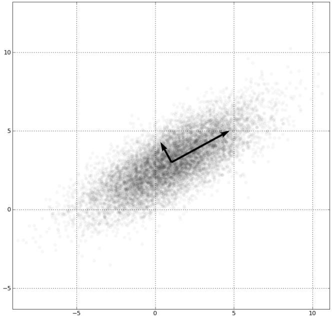


Figure: Principal directions found by PCA on a set of 2D points. Image by Ben FrantzDale from Wikimedia Commons.

Outline

- Mathematical background
- Scale Space Definition
- Point Set Orientation
- Mesh Reconstruction
- Scale Space Merging

Notations and definitions

- The surface $\mathcal{M}$ supporting the data point set is assumed to be at least C^2. The samples on the surface $\mathcal{M}$ are denoted by $\mathcal{M}_S$.
- $\mathbf{p}(x, y, z)$ be a point of the surface $\mathcal{M}$ with principal curvatures $k_1 > k_2$ (non-umbilical point)
- The quadruplet $(\mathbf{p}, \vec{t}_1, \vec{t}_2, \vec{n})$ is called the local intrinsic coordinate system. In this system we can express the surface as a C^2 graph $z = f(x, y)$. By Taylor expansion,

$$z = f(x, y) = -\frac{1}{2}(k_1 x^2 + k_2 y^2) + o(x^2 + y^2). \qquad (2)$$

Spherical neighborhoods vs cylindrical neighborhoods

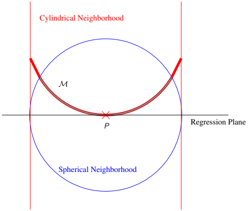


Figure: Cylindrical and spherical neighborhood

Lemma

Integrating on $\mathcal{M}$ any function $f(x,y)$ such that $f(x,y) = O(r^n)$ on a cylindrical neighborhood $\mathcal{C}_r(\mathbf{p})$ instead of a spherical neighborhood $\mathcal{B}_r(\mathbf{p})$ introduces an $o(r^{n+4})$ error. More precisely:

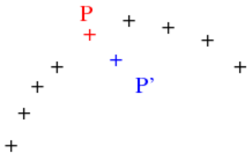
$$\int_{\mathcal{B}_r} f(x,y)dM = \int_{x^2+y^2<r^2} f(x,y)dxdy + O(r^{4+n}). \quad (3)$$

Projection on the local barycenter

Let $\mathbf{p}$ be a point of a data set $\mathcal{M}$ and denote by $\mathcal{N}_r(\mathbf{p})$ the set of all points $\mathbf{q}$ in $\mathcal{M}$ such that $\|\mathbf{p}-\mathbf{q}\| < r$.

Theorem

In the local intrinsic coordinate system, the barycenter O of a neighborhood $B_r(\mathbf{p})$ where $\mathbf{p}$ is the origin of the neighborhood has coordinates $x_O = o(r^2)$, $y_O = o(r^2)$ and $z_O = -\frac{Hr^2}{4} + o(r^2)$, where $H = \frac{k_1+k_2}{2}$ is the mean curvature at $\mathbf{p}$.



Link between normal to the regression plane and PCA least eigenvector

Lemma

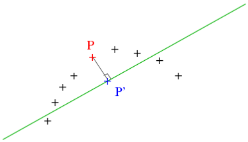
The normal $\vec{v}$ to the PCA regression plane at $\mathbf{p} \in \mathcal{M}$ is equal to the surface normal at point $\mathbf{p}$, up to a negligible factor: $\vec{v} = \vec{n}(\mathbf{p}) + O(r)$.

Projection on the local regression plane

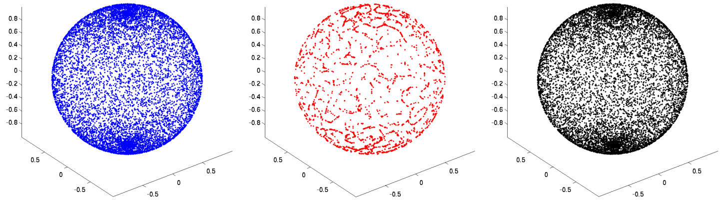
Let $\mathbf{p}$ be a point of a data set $\mathcal{M}$ and denote by $\mathcal{N}_r(\mathbf{p})$ the set of all points $\mathbf{q}$ in $\mathcal{M}$ such that $\|\mathbf{p} - \mathbf{q}\| < r$.

Theorem (Plane projection filter)

In the local intrinsic coordinate system of a continuous and smooth two-dimensional manifold $\mathcal{M}$, for $\mathbf{p} \in \mathcal{M}$ the projection $\mathbf{p}'$ of $\mathbf{p}$ on the local regression plane has coordinates $x_{\mathbf{p}'} = o(r^2)$, $y_{\mathbf{p}'} = o(r^2)$ and $z_{\mathbf{p}'} = \frac{Hr^2}{4} + o(r^2)$, where $H = \frac{k_1 + k_2}{2}$ is the surface mean curvature at $\mathbf{p}$ and k_1, k_2 the surface principal curvatures at $\mathbf{p}$.



The discrete case: plane vs barycenter projection filter



(a) Original samples on a sphere

(b) 4 iterations of barycenter filter

(c) 4 iterations of the projection filter

Figure: Comparison of the clustering effect for the barycenter filter and the projection filter on a randomly sampled sphere.

Consequence

Theorem

Let T_r be the operator defined on the surface $\mathcal{M}$ transforming each point **p** *into its projection on the local regression plane. Then*

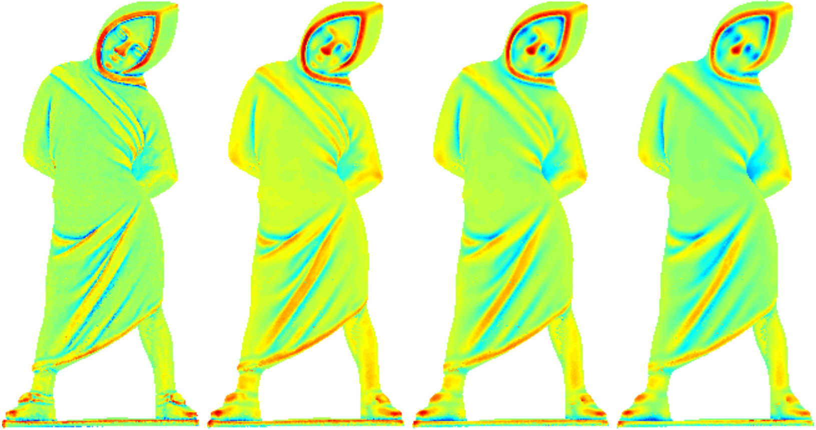
$$T_r(\mathbf{p}) - \mathbf{p} = \frac{Hr^2}{4}\mathbf{n}(\mathbf{p}) + o(r^2). \tag{4}$$

- $H \approx \frac{4\langle \mathbf{p}' - \mathbf{p}, \mathbf{n}(\mathbf{p})\rangle}{r^2}$
- It is a very stable estimate since it relies on order 1 approximation

Practical Scale Space Algorithm

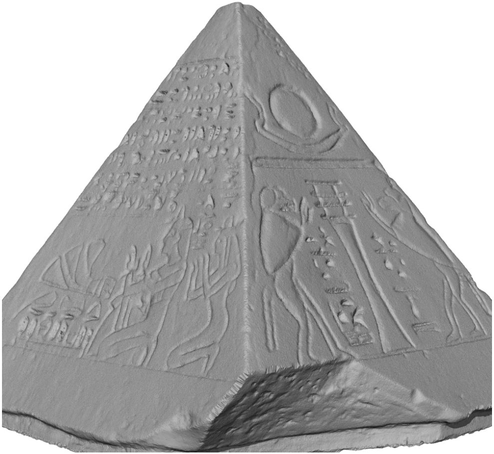
Projection on the *PCA regression plane*

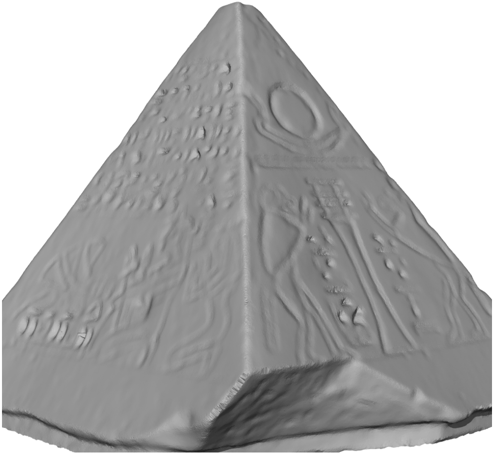
1. Get the set of neighbors $\mathcal{N}_r(\mathbf{p})$
2. Compute the barycenter $O = \sum_{Q \in \mathcal{N}_{\boldsymbol{r}}(\mathbf{p})} Q$
3. Compute the centered covariance matrix $\Sigma = \sum_{Q \in \mathcal{N}_{\boldsymbol{r}}(\mathbf{p})} (Q - O)^T (Q - O)$
4. Get the eigenvector v_0 corresponding to the least eigenvalue of Σ.
5. $\mathbf{p}_{new} = \mathbf{p} + \langle \mathbf{p} - O, v_0 \rangle v_0$

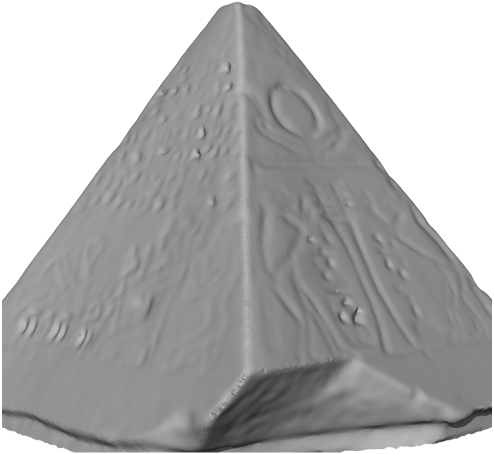


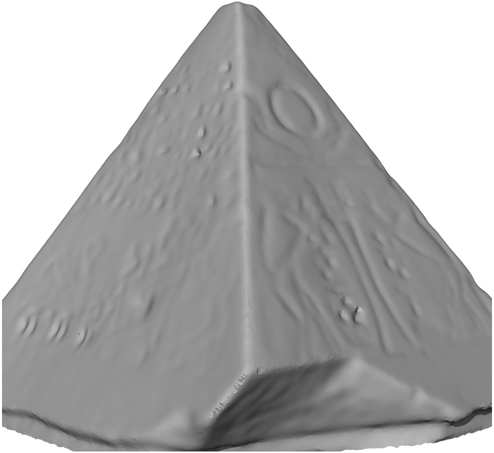
(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

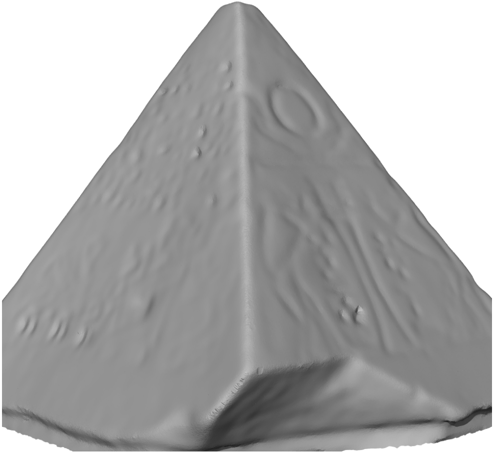
Figure: Curvature evolution by iterative projection (T_r)

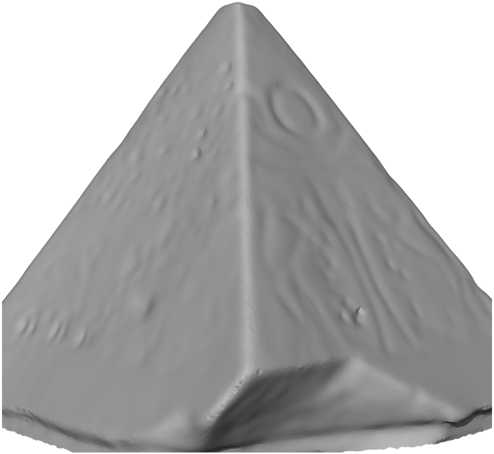


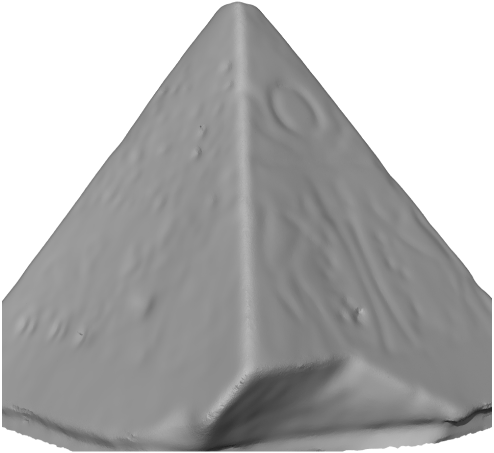


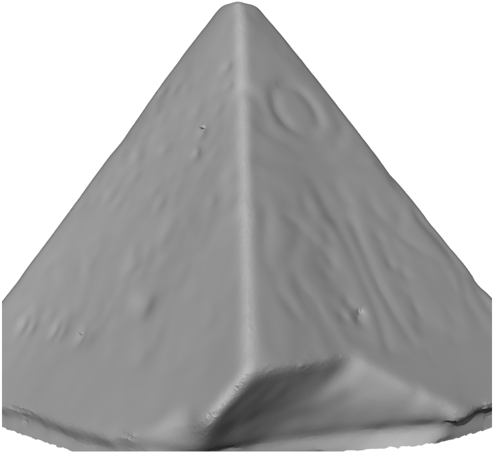


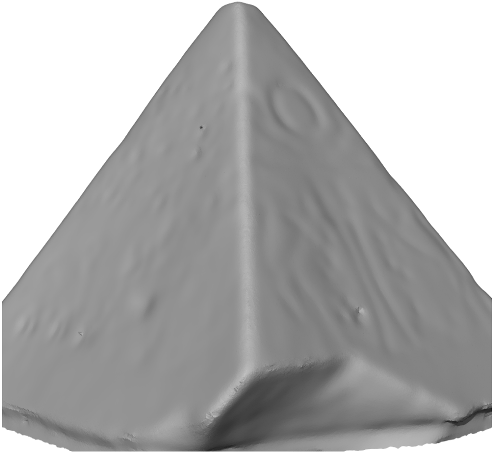












3D scale space

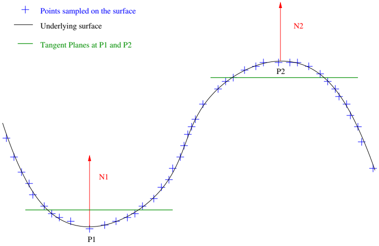
- Scale space for images: simplify the image to get the global information
- 3D equivalent to the 2D heat equation scale space
- Idea: perform low scale robust processing and propagate the information back on the original data

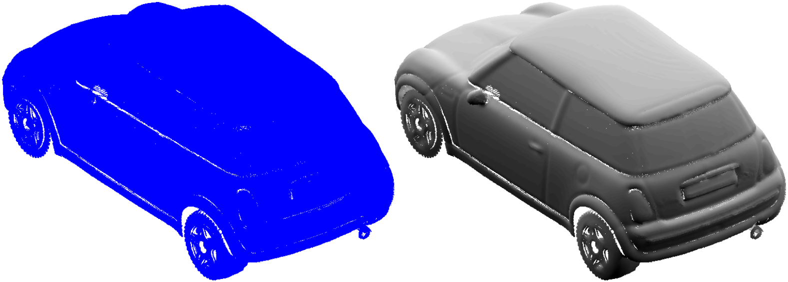
Outline

Application to the 3D point set orientation problem

- The eigenvector corresponding to the least eigenvalue of the local covariance matrix is a good approximation of the normal direction
- There is still an ambiguity on the orientation
- We must find a coherent orientation (outward pointing normal for example)
- Idea: perform the orientation propagation at coarse scale

- Compute the normal directions for all points
- Apply N scale space iterations
- Choose a point in a flat area, pick one of the two possible orientations
- Propagate orientation in the neighborhoods





Outline

Application: reconstructing shapes with textures and details

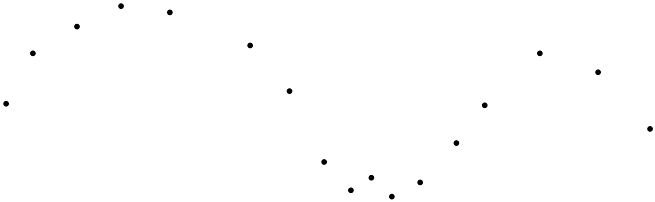
- The aim is to preserve textures and small details
- Any Level Set method is forbidden to avoid shape smoothing ([Hoppe et al., 92], [Khazdan, 05], [Khazdan et al., 06], [Alliez et al., 07])
- Triangulation by α-shapes the standard Ball Pivoting Algorithm also removes details ([Edelsbrunner, Mücke, 94], [Bernardini et al., 99])

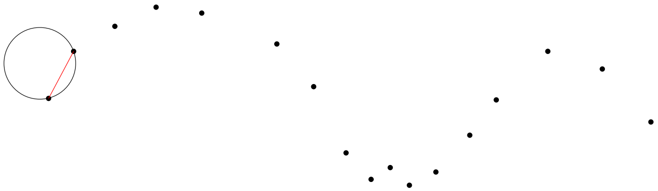
Algorithm

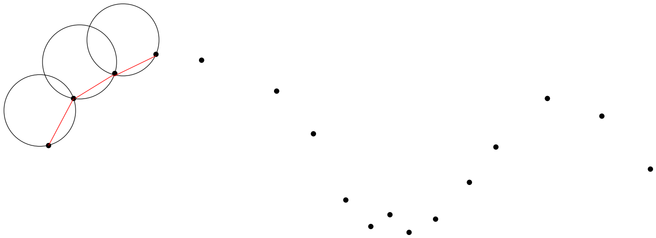
1. Apply N scale space iterations and keep a track at each step of the point displacements;
2. Mesh the resulting samples. The obtained mesh is singularity free;
3. Project the mesh back to the original points;
4. The result is an interpolating mesh which preserves textures and details.

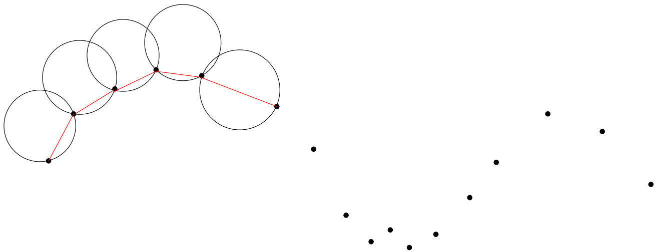
Meshing technique: Ball Pivoting Algorithm [Bernardini et al. 99]

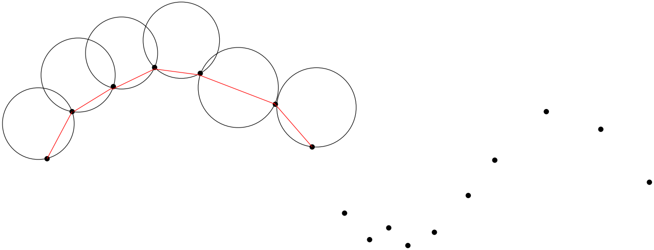
- Three points should be linked by a triangle if and only if one can fit a ball with radius r through the points and that this ball contains no other point.
- Starting with such a triplet of points, a ball is *pivoted* around each edge of the triangle until it meets another point, if the ball is empty then a new triangle is formed.

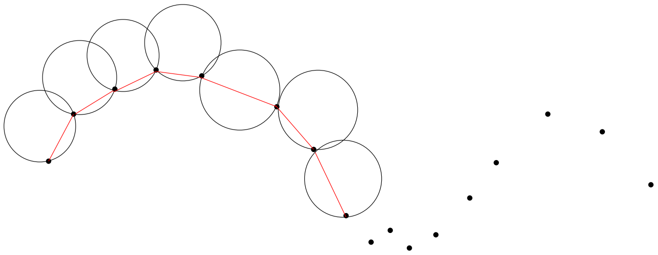


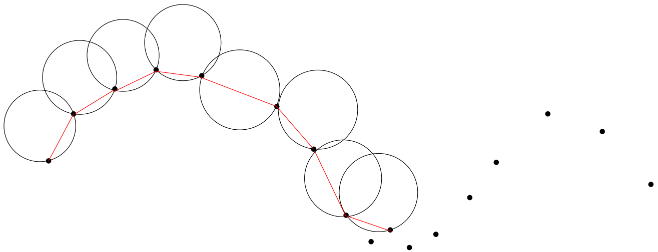


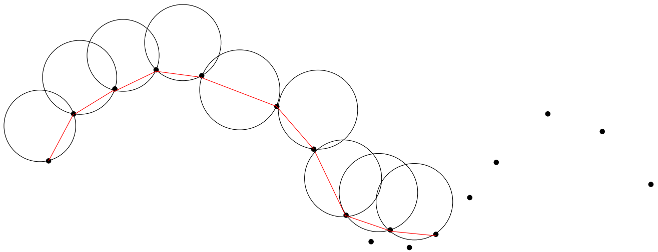


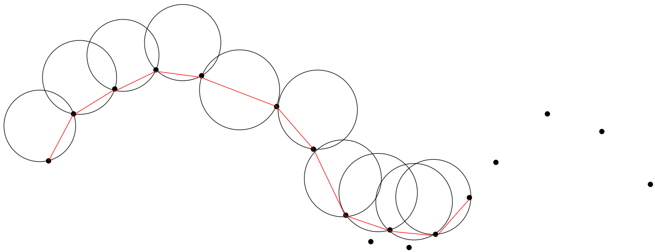


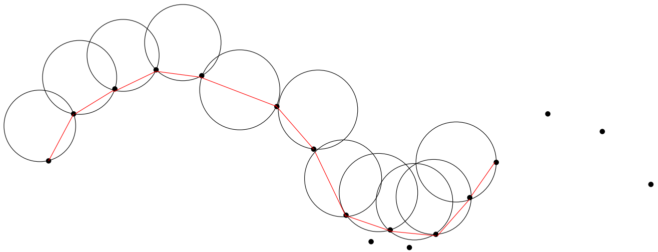


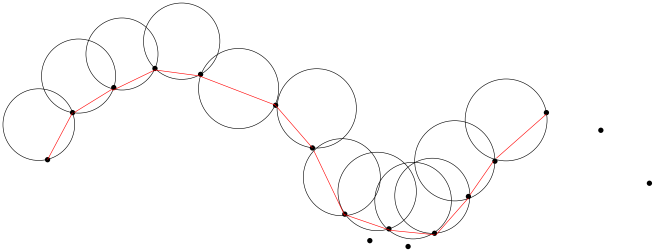


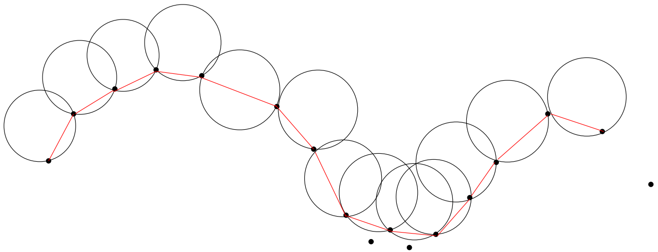


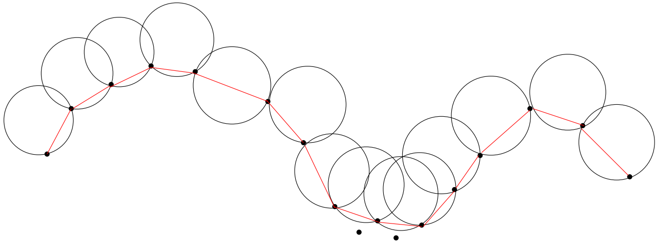




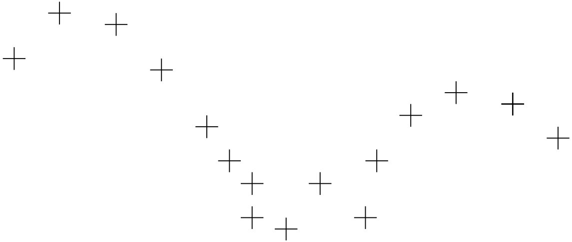






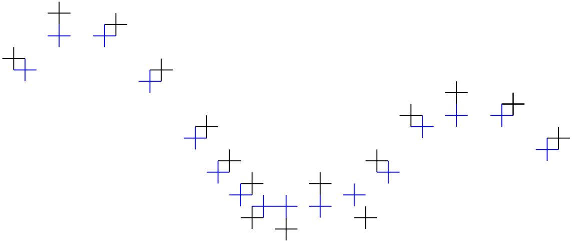


synthetic 1D example



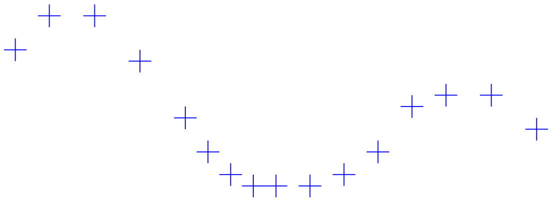
Initial points

synthetic 1D example



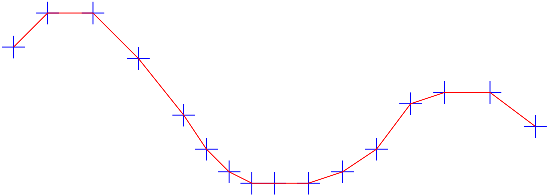
Initial points and their projections

synthetic 1D example



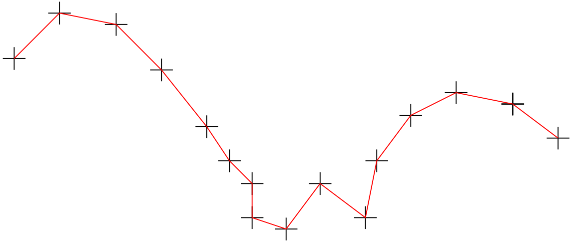
Projected points

synthetic 1D example



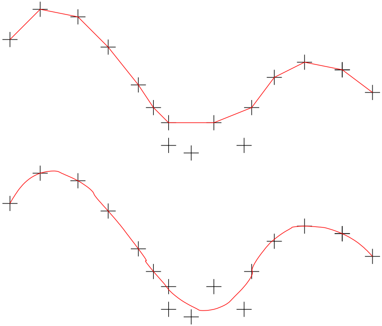
Resulting mesh of the projected points

synthetic 1D example



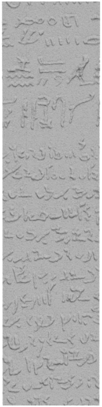
Back projected mesh of the initial points

synthetic 1D example



Top: same initial points with direct BPA triangulation

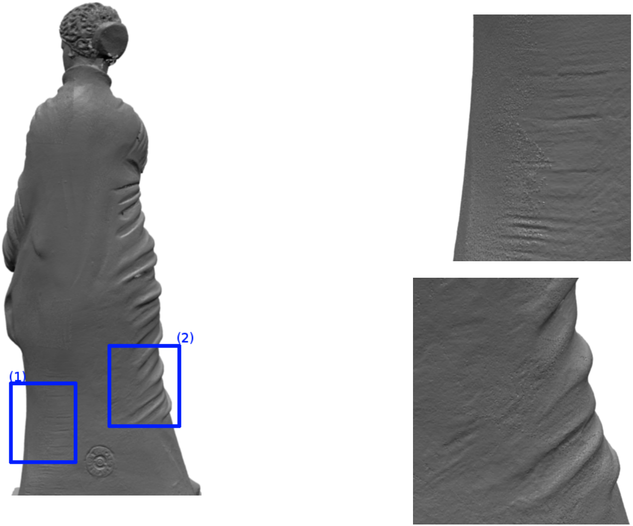
Bottom: same initial points with Poisson Reconstruction



Original Object: 20 cm high Tanagra

Coarse resolution Mesh (after projection iterations)

Mesh obtained at a high resolution (back-projected)


(1)
(2)

Comparison with other methods

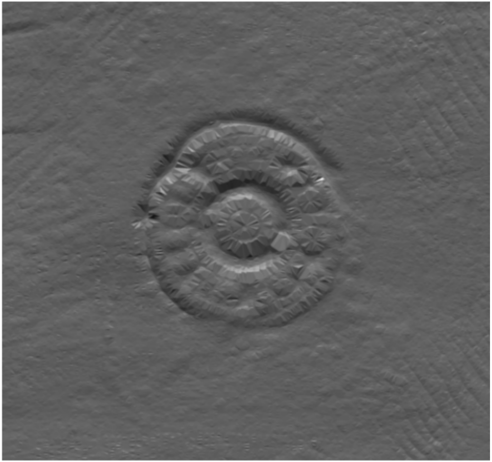


Figure: Direct meshing (Ball pivoting)

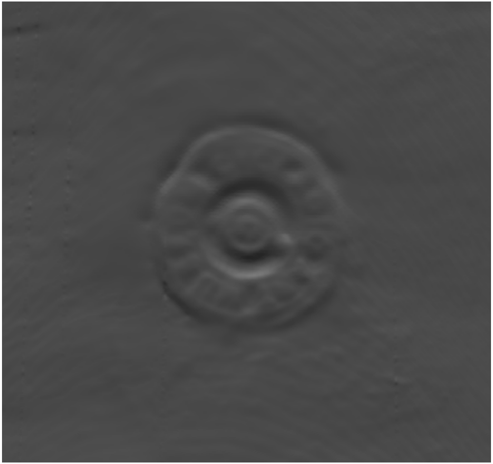


Figure: Poisson Reconstruction [Khazdan et al. 06]

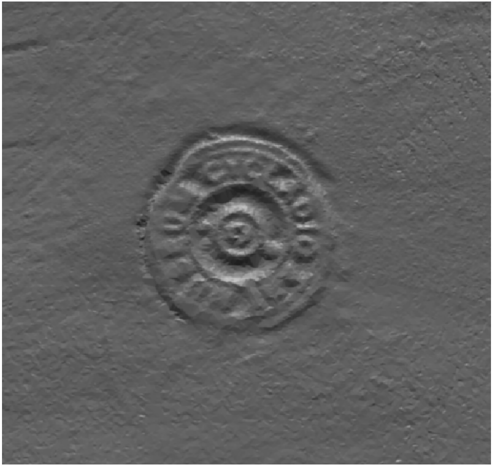


Figure: Scale Space Meshing

More comparisons...

Figure: Original Fragment

More comparisons...

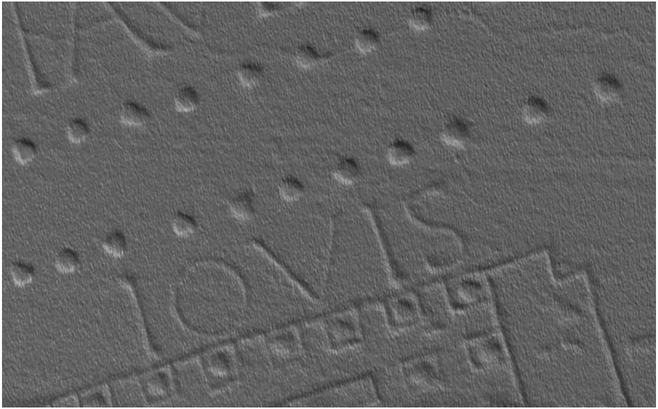


Figure: Back-projected mesh

More comparisons...

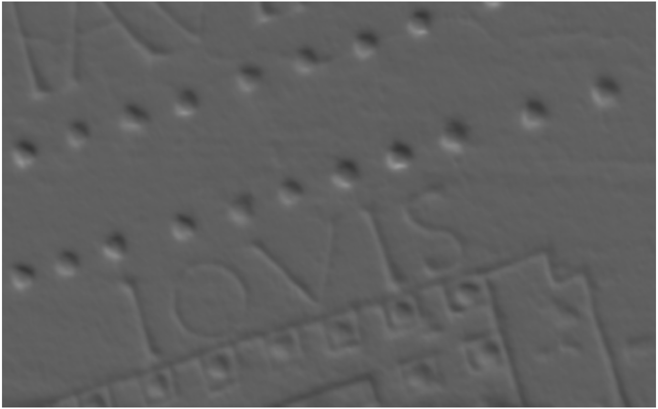


Figure: Poisson Mesh Reconstruction

Outline

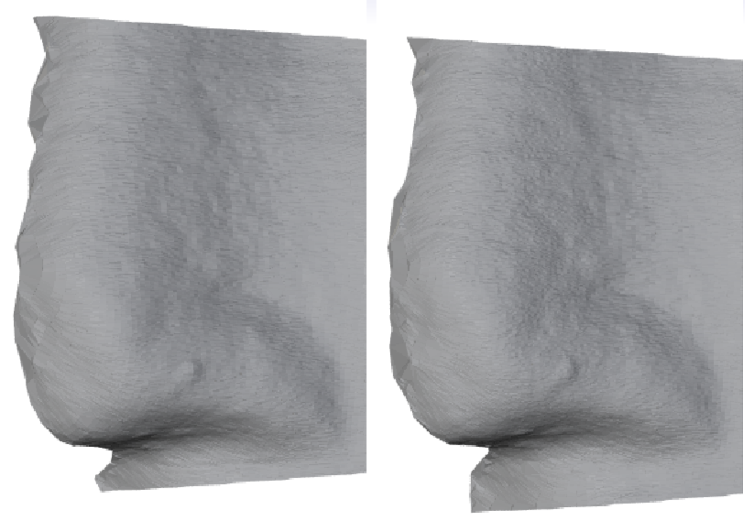
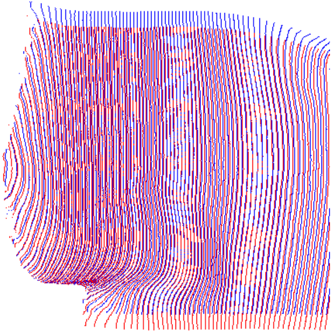


Figure: Two scans covering the same area



(a) Overlapping scans

(b) Joint Mesh of both scans

Scan Superposition Artefacts

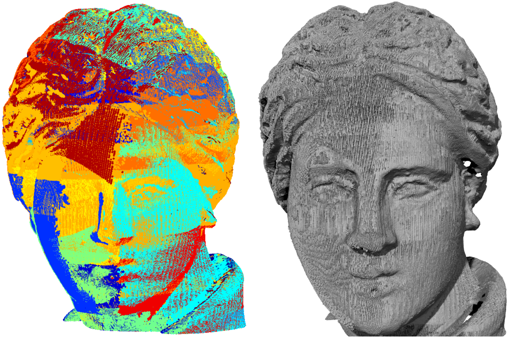


Figure: Example of artefacts created by the superposition of multiple scans (35 scans)

Method Overview

- Input: a set of scans $S_1, S_2, \cdots, S_N$
- Each scan can be decomposed into a smooth base and a high frequency part: $S_i = B_i + h_i$
- Find a common basis for all scans and add local high frequencies
- The idea is close to blending images with gaussian pyramids ([Burt and Adelson, 83]) or morphing methods ([Pauly et al., 06])

Algorithm

- N iterations of T_r are applied to each scan separately, the displacement vectors δp are stored and points are moved back to their original positions.
- N iterations of T_r are applied to all the input scans together yielding a set of globally smoothed positions p'
- The displacement vectors are then added back to the smoothed points: $p_f \leftarrow p' + \delta p$
- Two parameters method: radius r and number of iterations N

Results

Figure: left: input, middle: smooth base found, right: adding high frequencies

Some more results (video)

Comparison of the merging result with a groundtruth

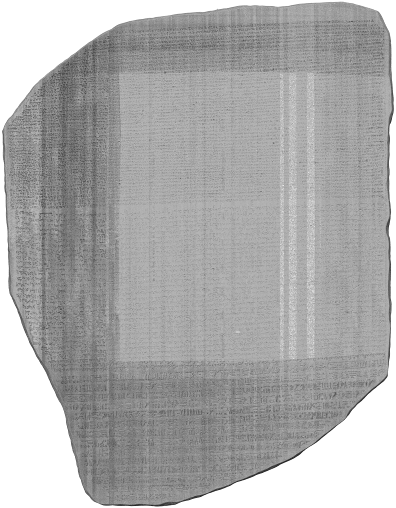
Figure: Left: ground truth, middle: and the merging of all scans that overlap in the same region, right: joint mesh with merging

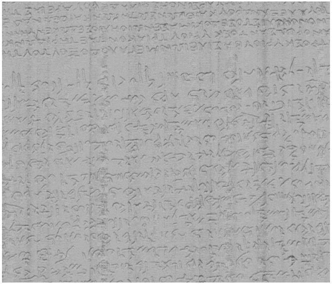
How accurate can we be?

- Test object: a reproduction of the rosetta stone.
- Engravings are around $50 - 100\mu$ deep
- Can these engravings be acquired and processed?
- Laser acquisition yielded $36,201,537$ points and 32 scans.

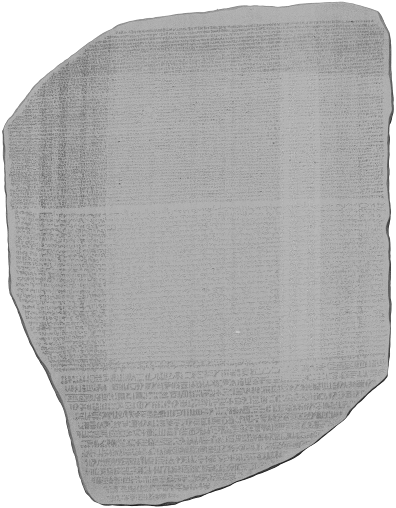
Picture of the object

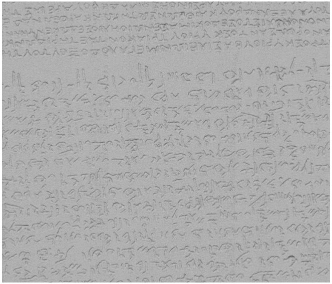
Joint mesh of the scans



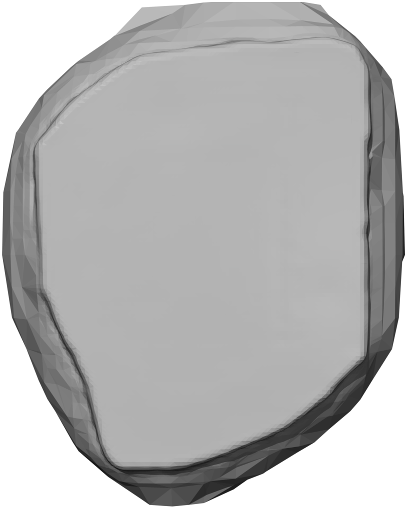


Merged scans mesh





Why not use Poisson Reconstruction [Kazhdan et al. 06]?



Conclusion

- A pipeline for processing highly accurate point clouds based on the scale space was introduced
- In a unified way the scale space solves various problems: point set orientation, meshing, merging and mesh segmentation
- Data available on www.ipol.im : *Farman Institute 3D Point Sets.*

Some algorithmic problems and how to solve them

- All algorithms for 3D point sets are based on the definition of a *neighborhood*
- Two kinds of neighborhoods: k-nearest neighbors or ball neighborhood.
- Data is unstrucured: how do we get the neighbors? Naive algorithm is $O(N^2)$

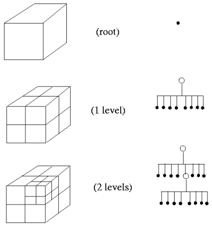


Figure: An octree

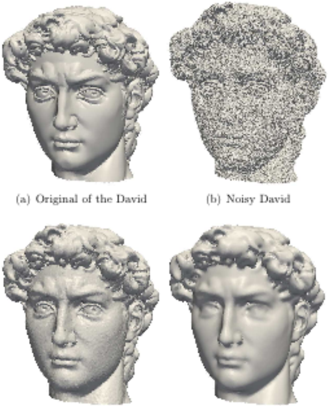
An algorithm to sort the points

Given a set of points and a depth d:

- An initial root node is built so that it contains all points;
- When a point $\mathbf{p}$ is added to a cell C:
  - If the cell has depth d it is a leaf and the point is simply added to the list of points of C;
  - Otherwise, look for the cell child C_i that would contain the point;
  - If C_i does not exist create it and add the point to it.

Bilateral filtering of point sets

- Goal: denoising a shape without loosing the sharp features
- Generalization of the gray image bilateral filter
- Each point is updated in the normal direction by δp



(a) Original of the David (b) Noisy David

(c) Bilateral denoising (d) MCM

$$\delta p = \frac{1}{C} \sum_{p' \in \mathcal{N}(p)} \exp - \frac{\|p - p'\|^2}{\sigma_d^2} \exp - \frac{\langle n, p' - p \rangle^2}{\sigma_n^2} \langle n, p' - p \rangle$$

Resampling of point sets: Parameterization-free projection for geometry reconstruction

- Goal: resampling a shape while keeping sampling holes and sharp features
- A local projection operator (LOP) is defined, by projecting a set of points on the surface, a new surface sampling is obtained.
- The projected points $\mathbf{q}$ are the fixed points of an equation $\mathbf{q} = G(\mathbf{q})$ where G is a functional made of two balancing terms: one that drives $\mathbf{q}$ to the points $\mathbf{p}$ of the original set and one that strives to keep the distribution of $\mathbf{q}$ homogeneous.