Structured light 3D reconstruction

Reconstruction pipeline and industrial applications

Emanuele Rodolà
rodola@dsi.unive.it

11/05/2010

3D Reconstruction

3D reconstruction is the process of capturing the shape and appearance of real objects

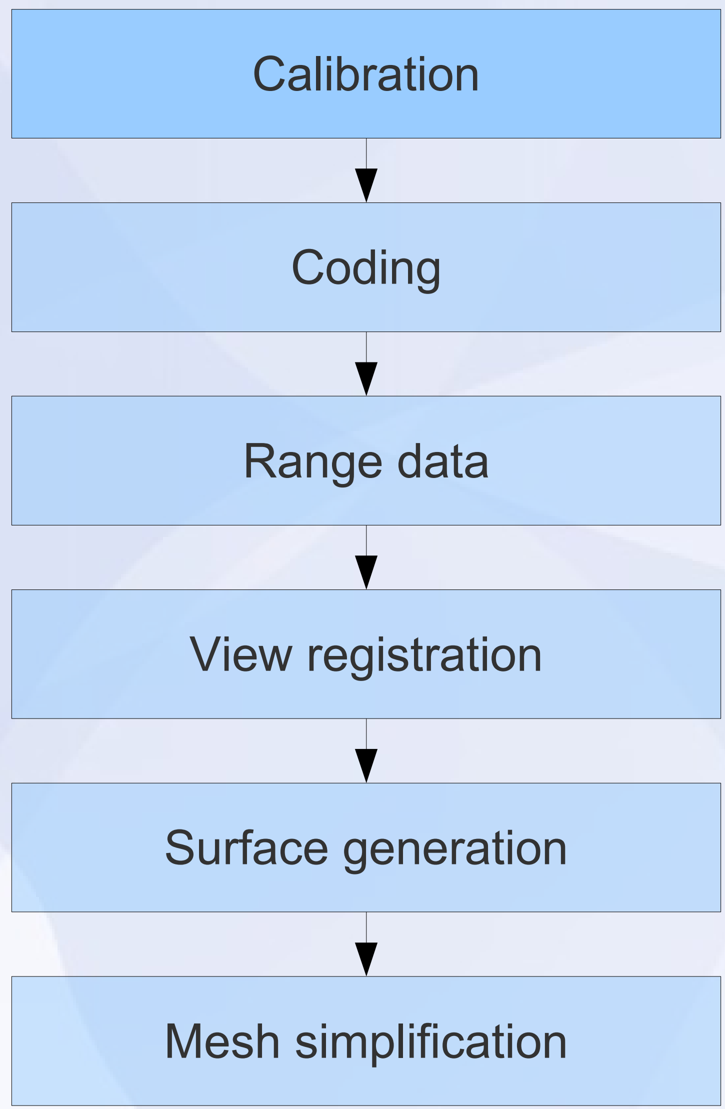
- Passive methods (*Structure from Motion*)
- Active methods (*time-of-flight*, *laser scanning*, *structured light*)

Application scenarios

- Reverse engineering
- Industrial monitoring
- Cultural heritage preservation
- Real-time modeling
- Interaction of Robots with the environment
- Underwater scenes reconstruction

Stanford bunny

Reconstruction pipeline

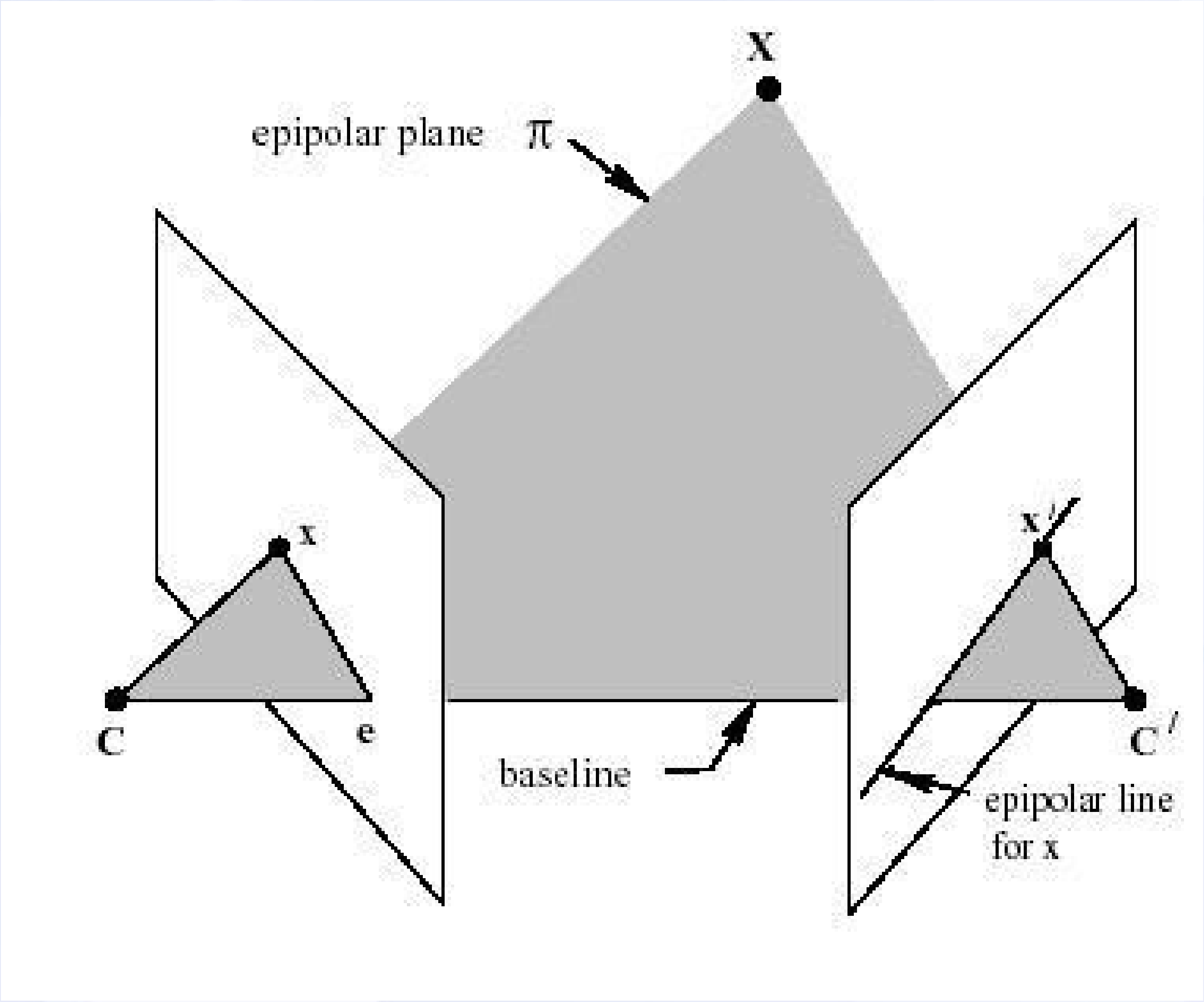


The correspondence problem

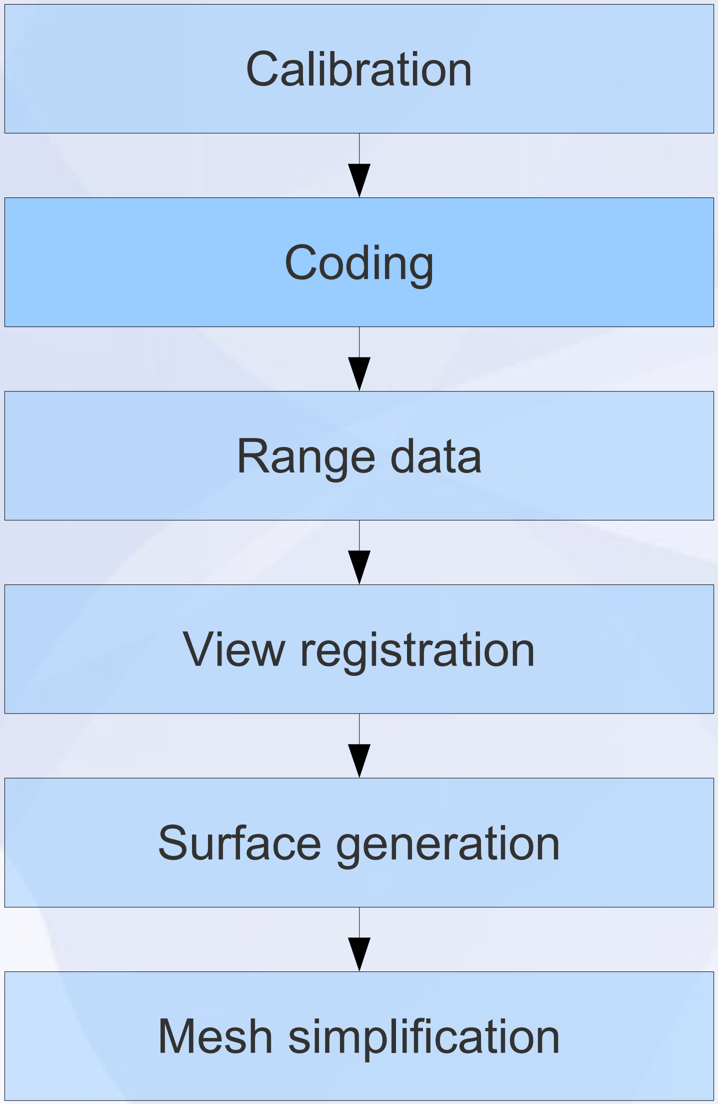
Given two or more images of the same scene, taken from different *points of view*, the "correspondence problem" is to find a set of points in one image which can be identified as the same points in another image.

The correspondence problem typically occurs when two images of the same scene are used (*stereo* correspondence problem)

The correspondence problem



Reconstruction pipeline



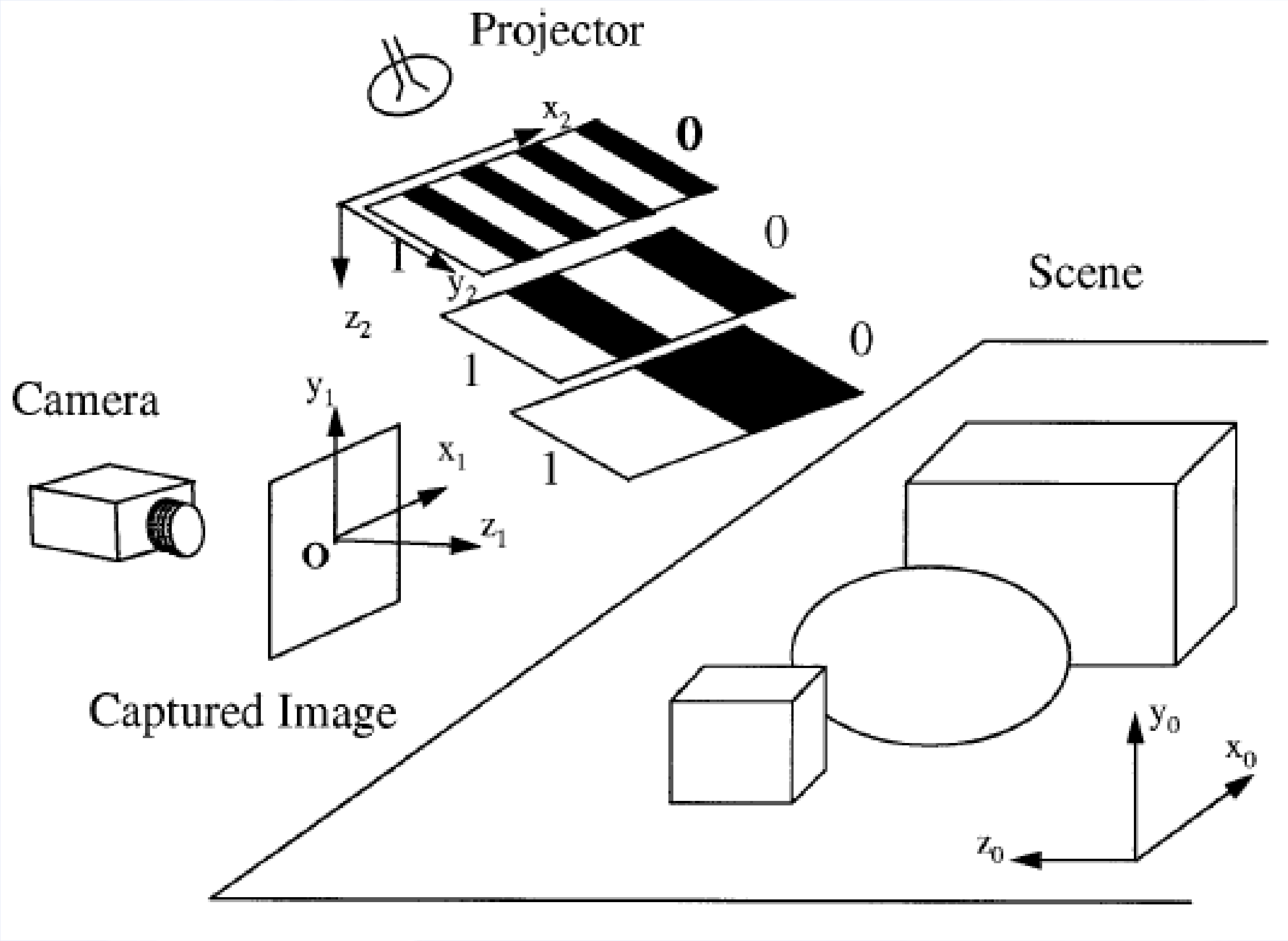
Structured light

A light source projects a known pattern of light on the scene.

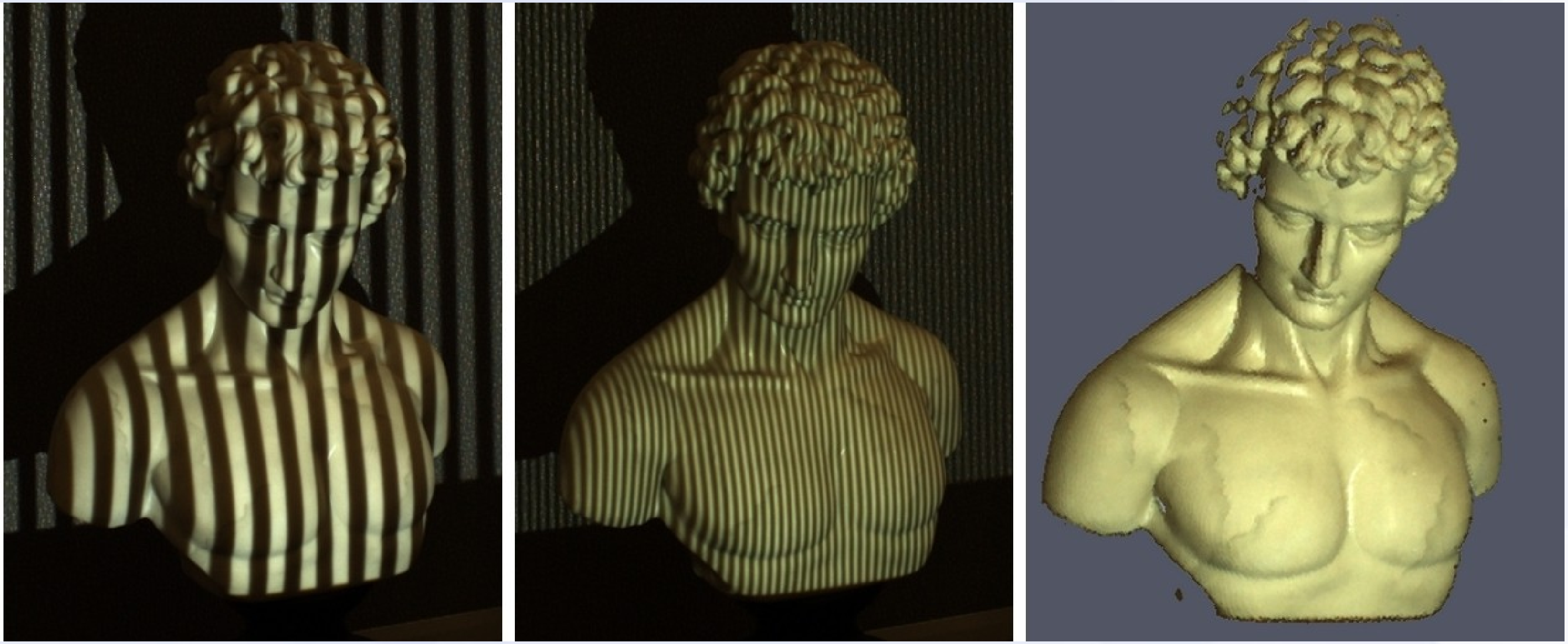
The projected pattern is then captured by one (or more) cameras.

Correspondences can be established by looking at the coded pixels.

Binary coding

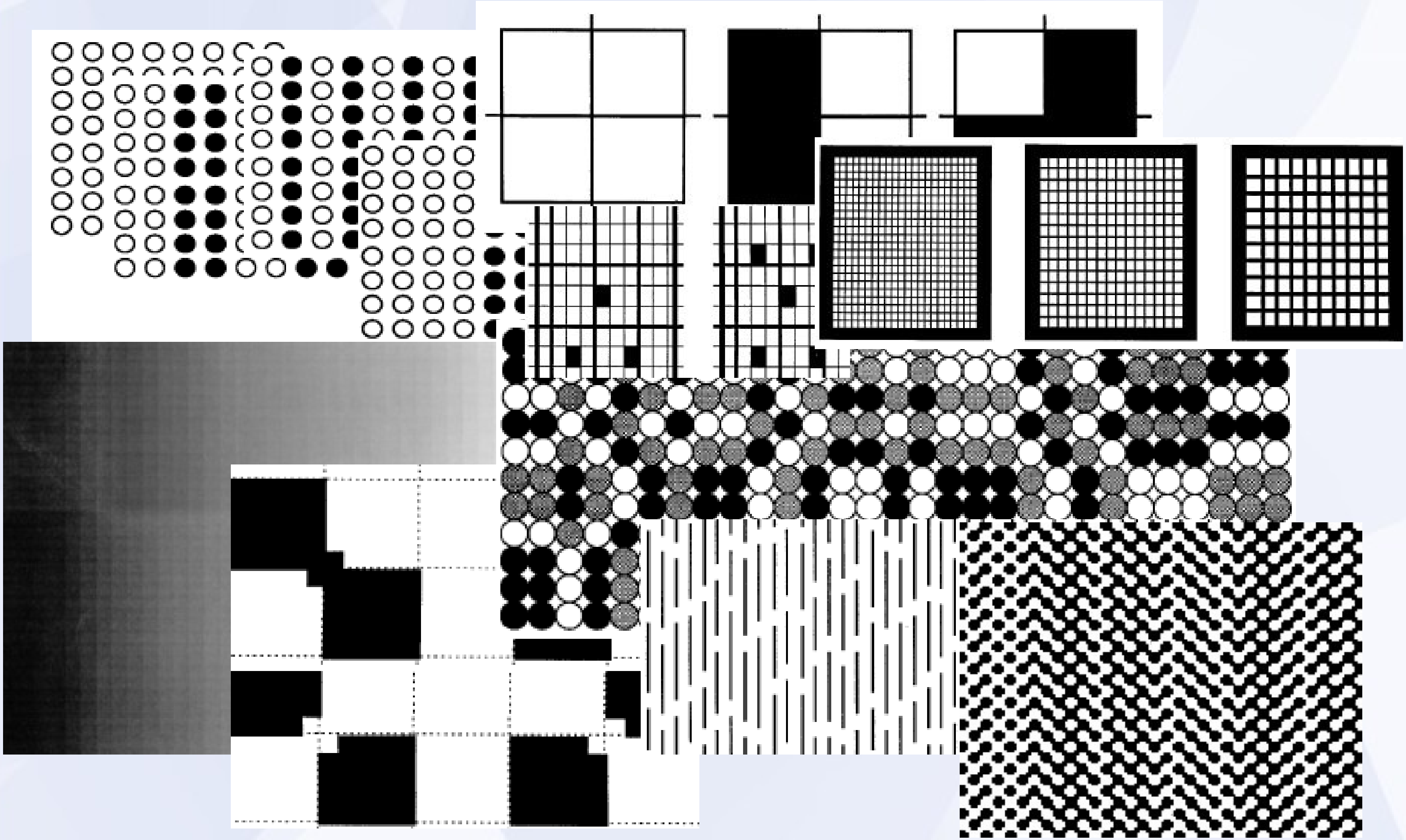


Binary coding



What is the scan resolution?
How will the scan look like?

Coding techniques



Phase-shift coding

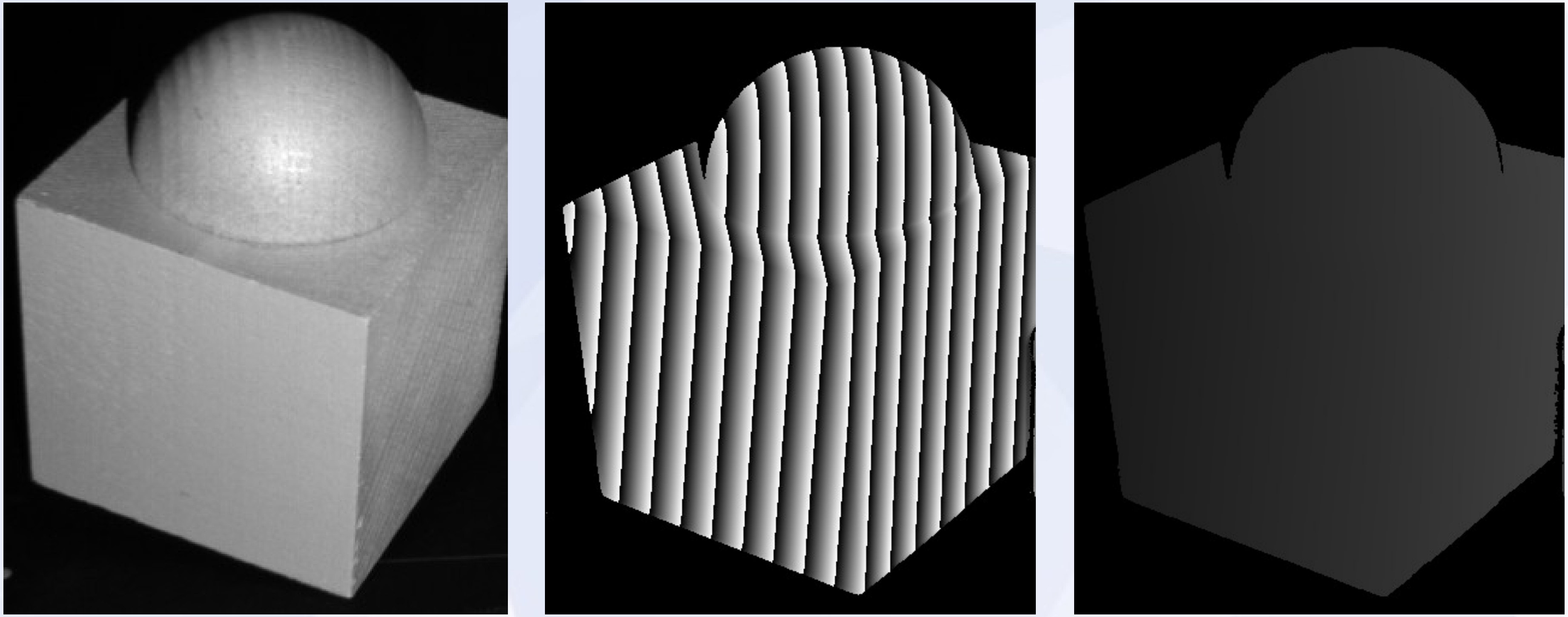
Project a periodic intensity pattern several times by shifting it in every projection.

For every given pixel, the phase of the first periodic pattern projected to the corresponding surface point must be found.

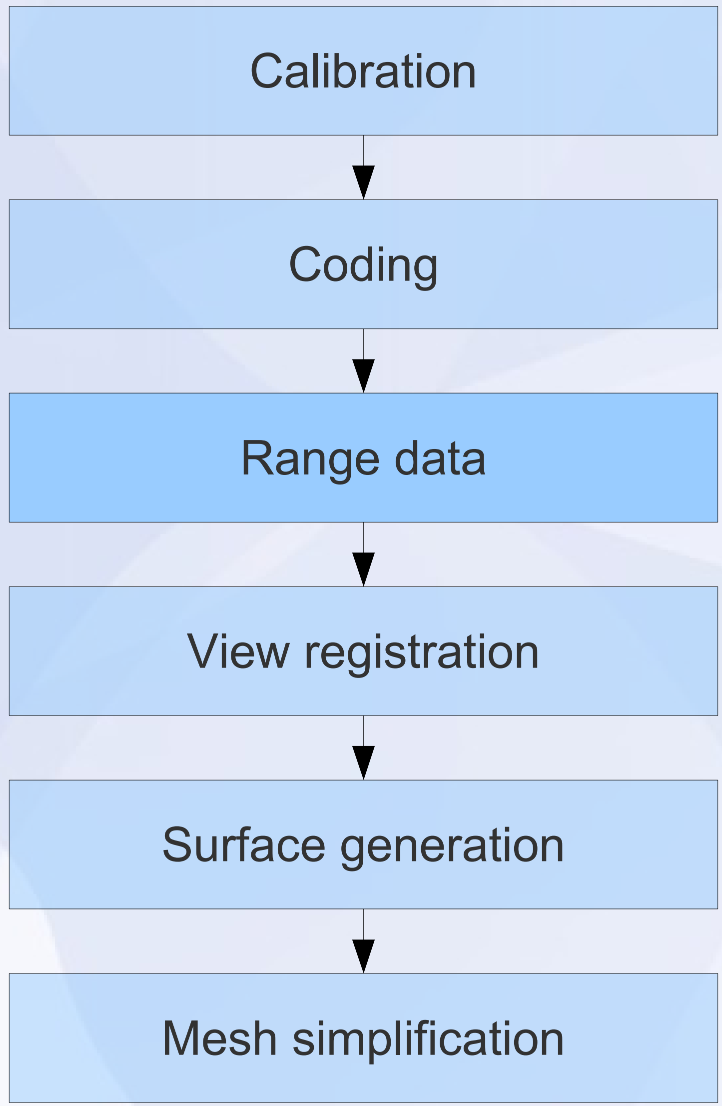
4-step: $$\phi(x, y) = \arctan\left(\frac{I_2 - I_4}{I_3 - I_1}\right)$$

Phase-shift coding

Phase-shift codes are (typically) ambiguous!



Reconstruction pipeline



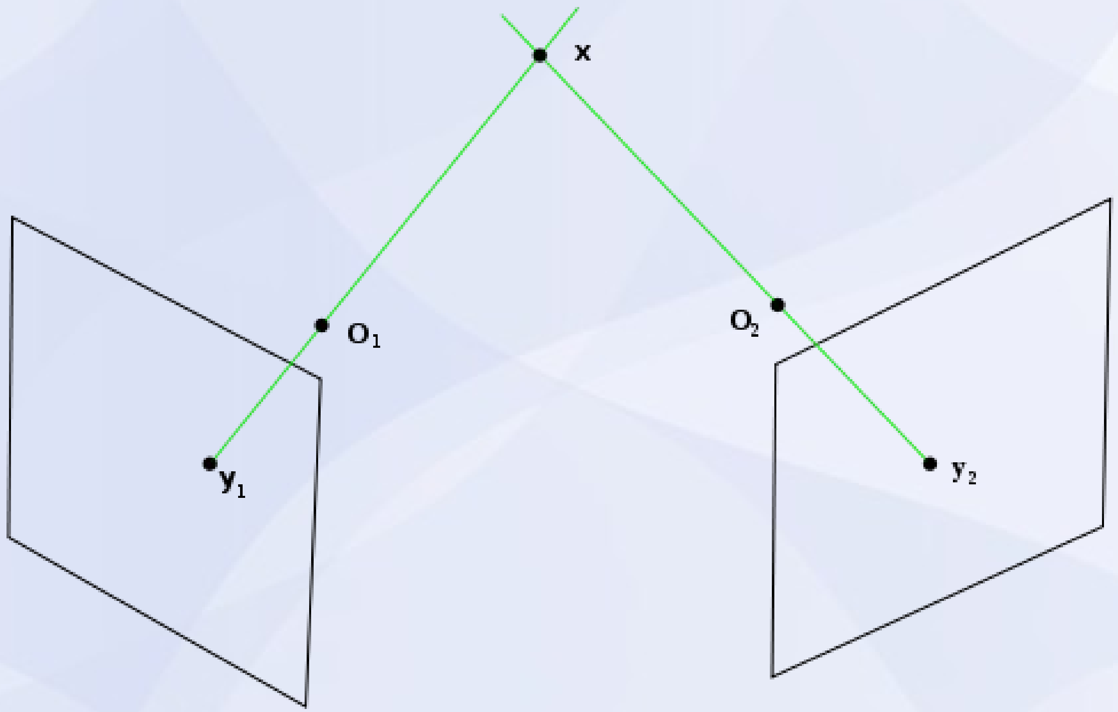
Triangulation

The process of determining a point in 3D space given its projections onto two, or more, images.

It is necessary to know the parameters of the camera projection function from 3D to 2D for the cameras involved.

$$P = K\,[\,R\ t\,] \qquad sx = PX$$

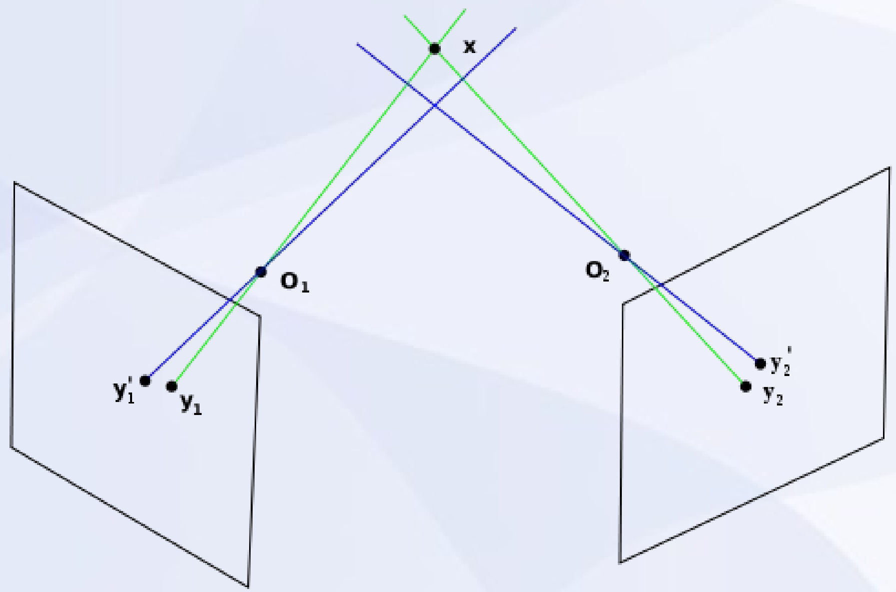
Triangulation (ideal)



Corresponding image points are the projection of a common 3D point

Triangulation (real)

- Coordinates of image points cannot be measured with arbitrary accuracy (lens distortion, detection error, ambiguities)
- We look for a 3D point which “optimally” fits the measured image points.



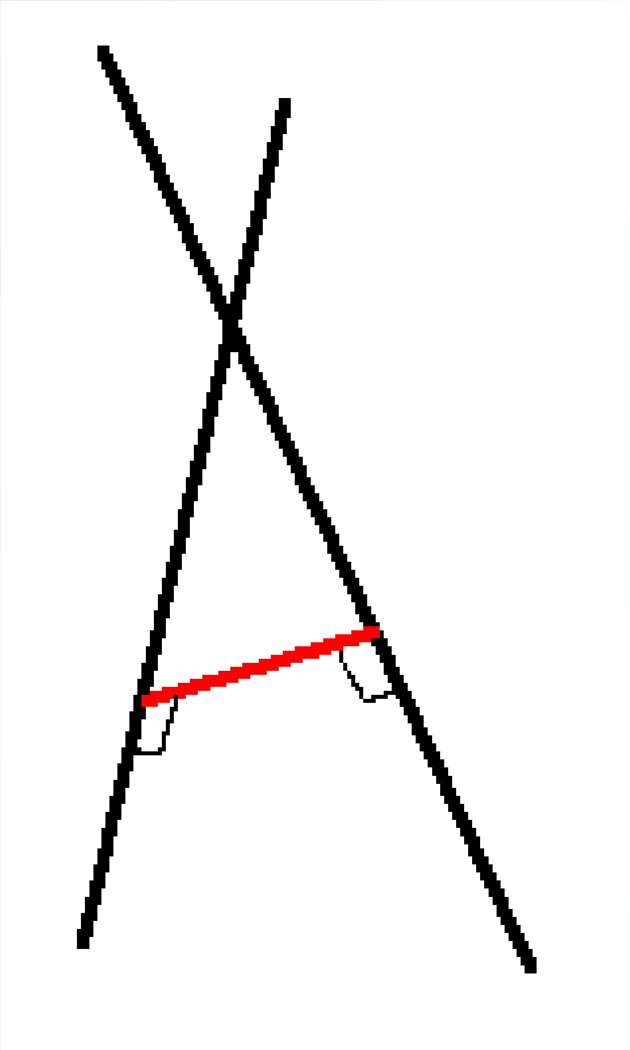
Mid-point algorithm

d(L,x) = Euclidean dist. between **L** and **x**

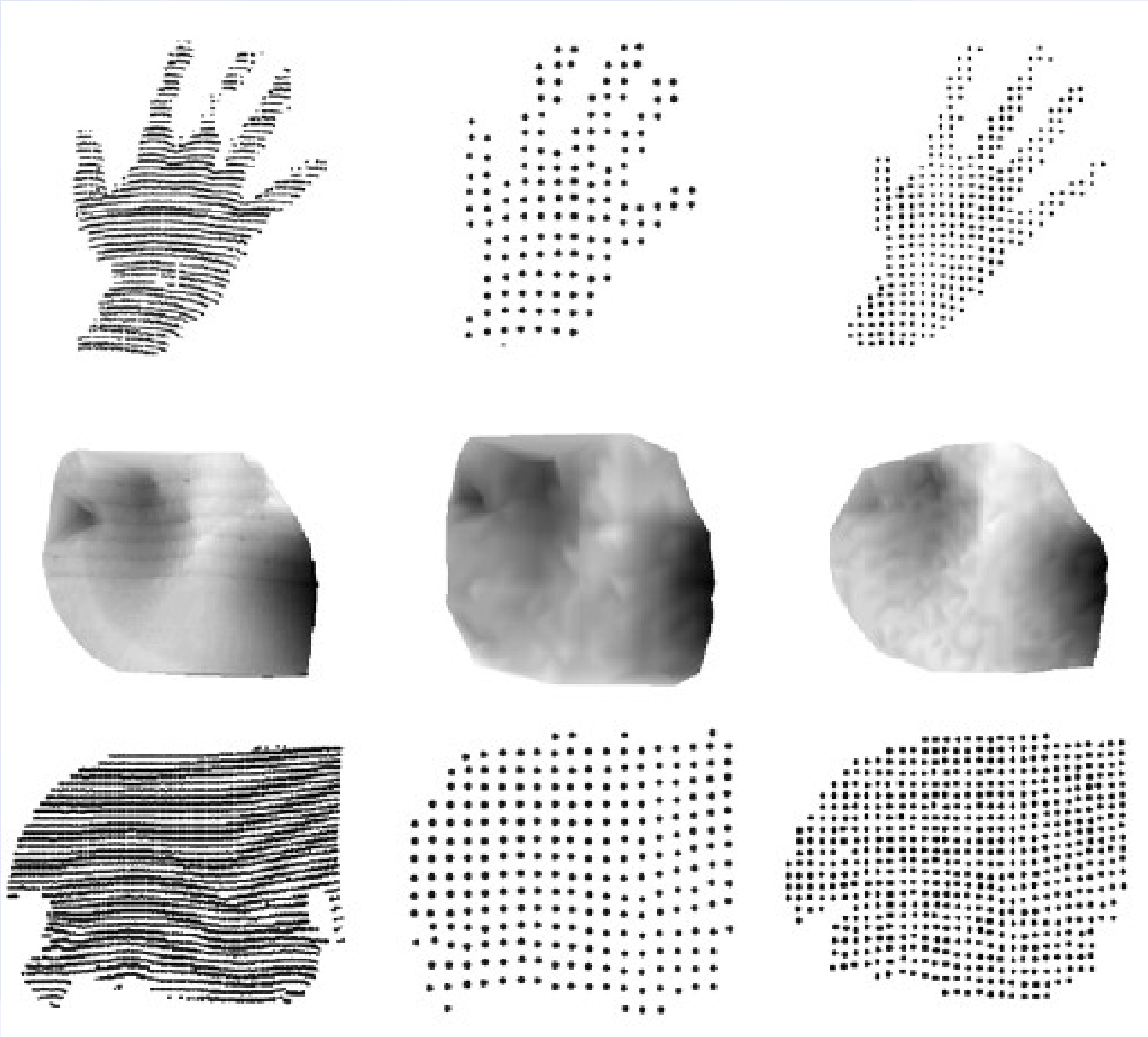
This method finds the point **x'** which minimizes:

$$\mathbf{d(L,x)^2 + d(L',x)^2}$$

It turns out that **x'** lies exactly at the middle of the shortest segment which joins the two projection lines.

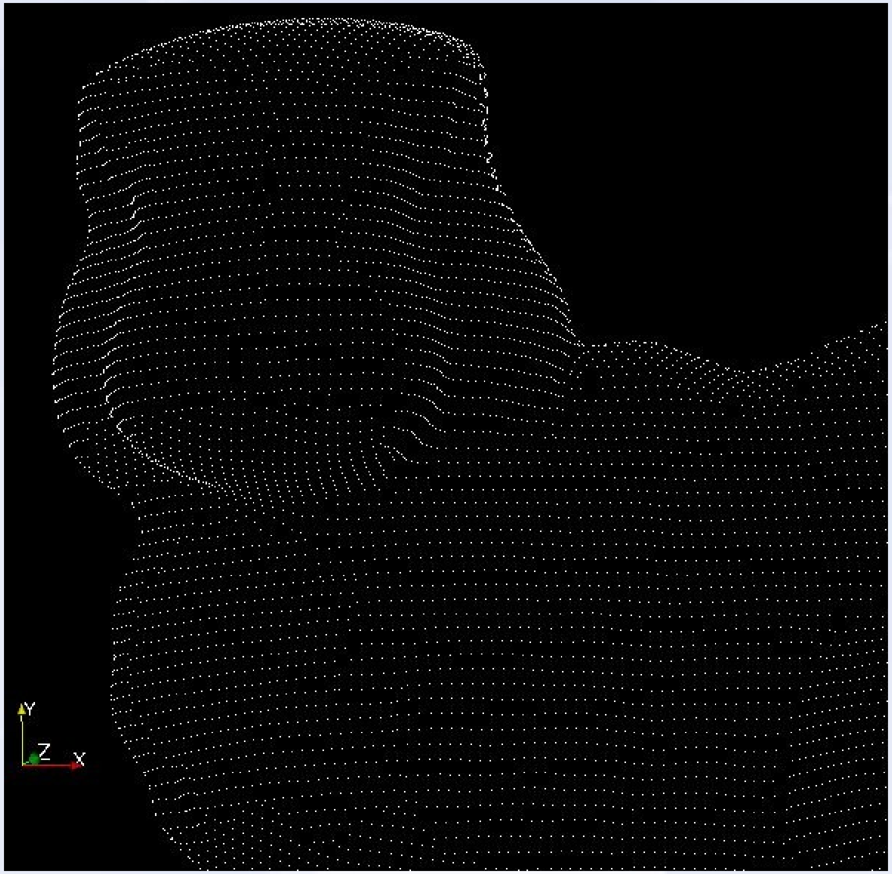


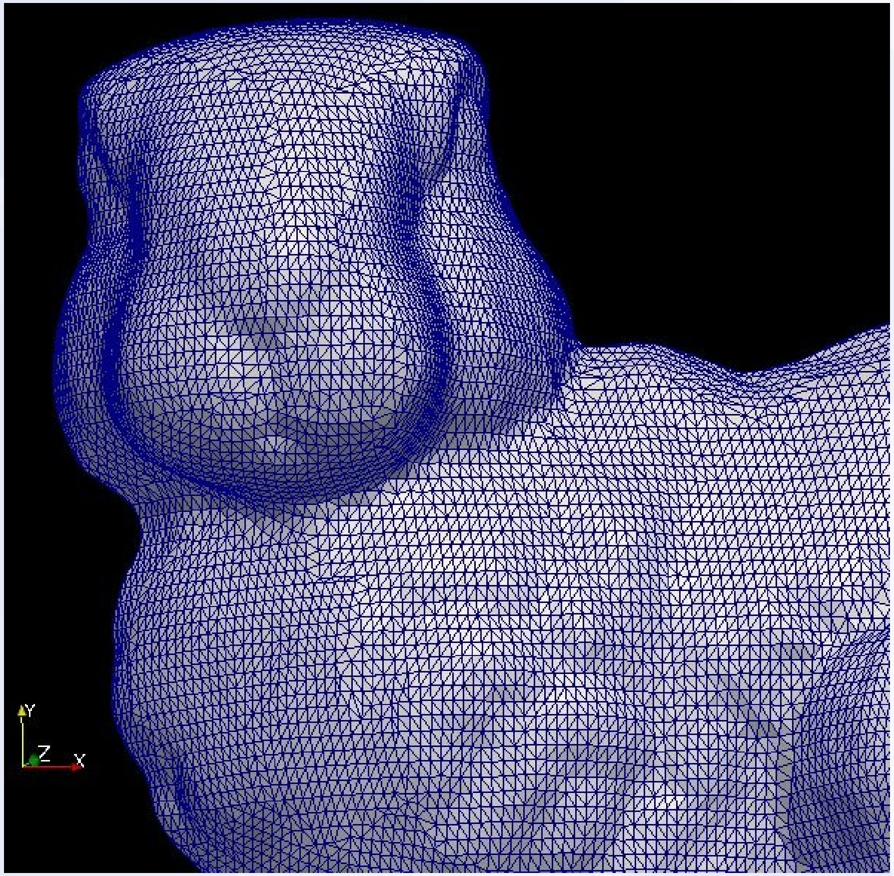
Rangemaps



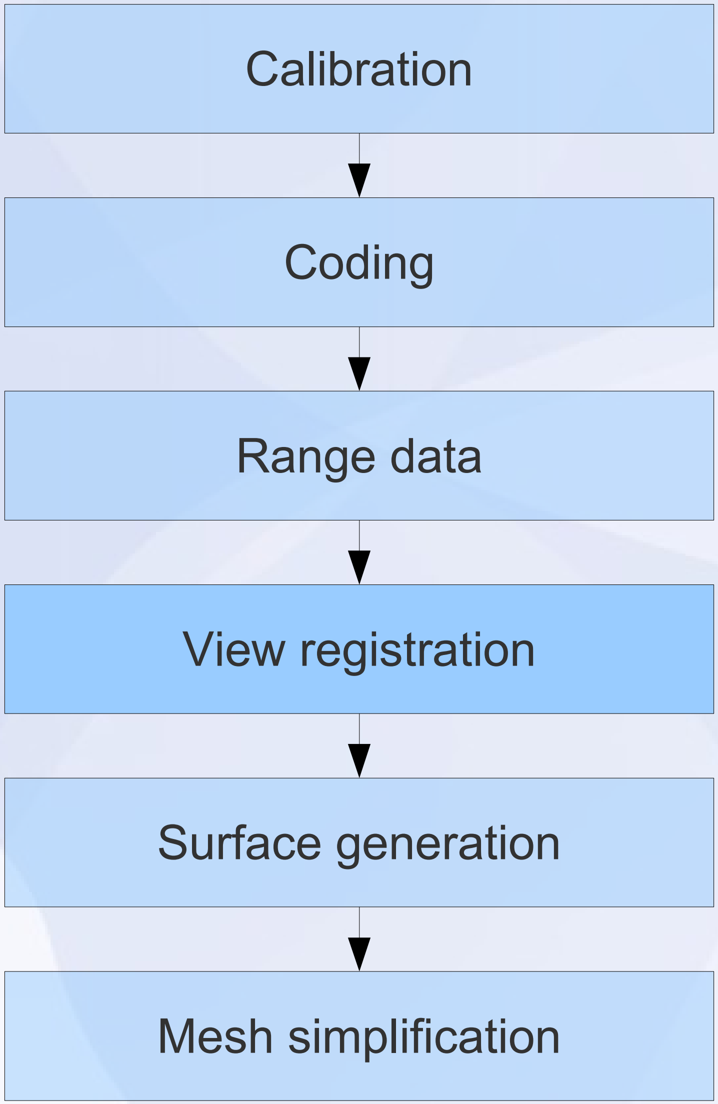
Tessellation

Provide range data with *connectivity* information, in order to give a *structure* to it.

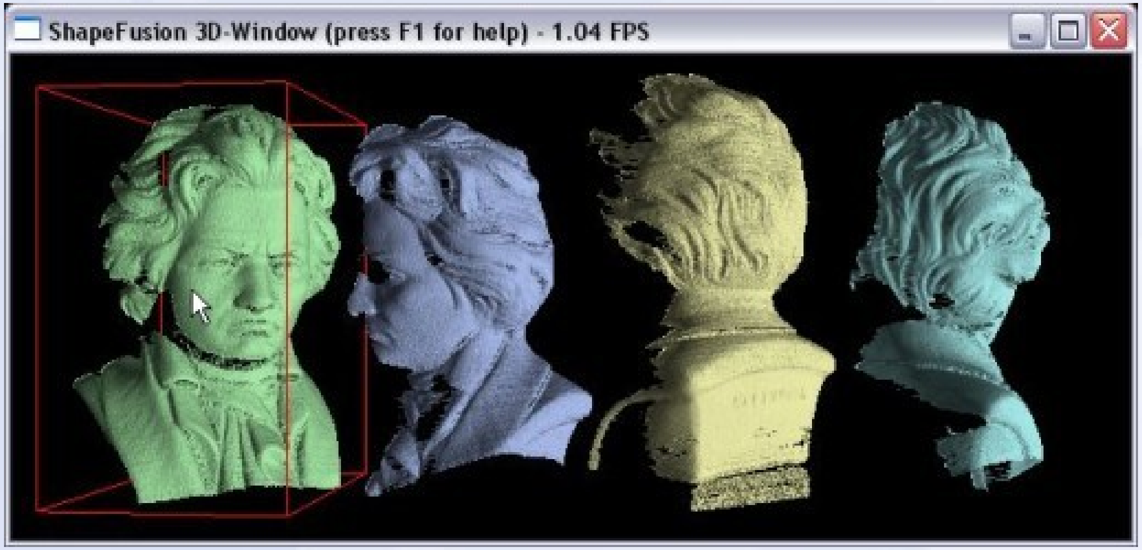




Reconstruction pipeline



Registration (coarse)

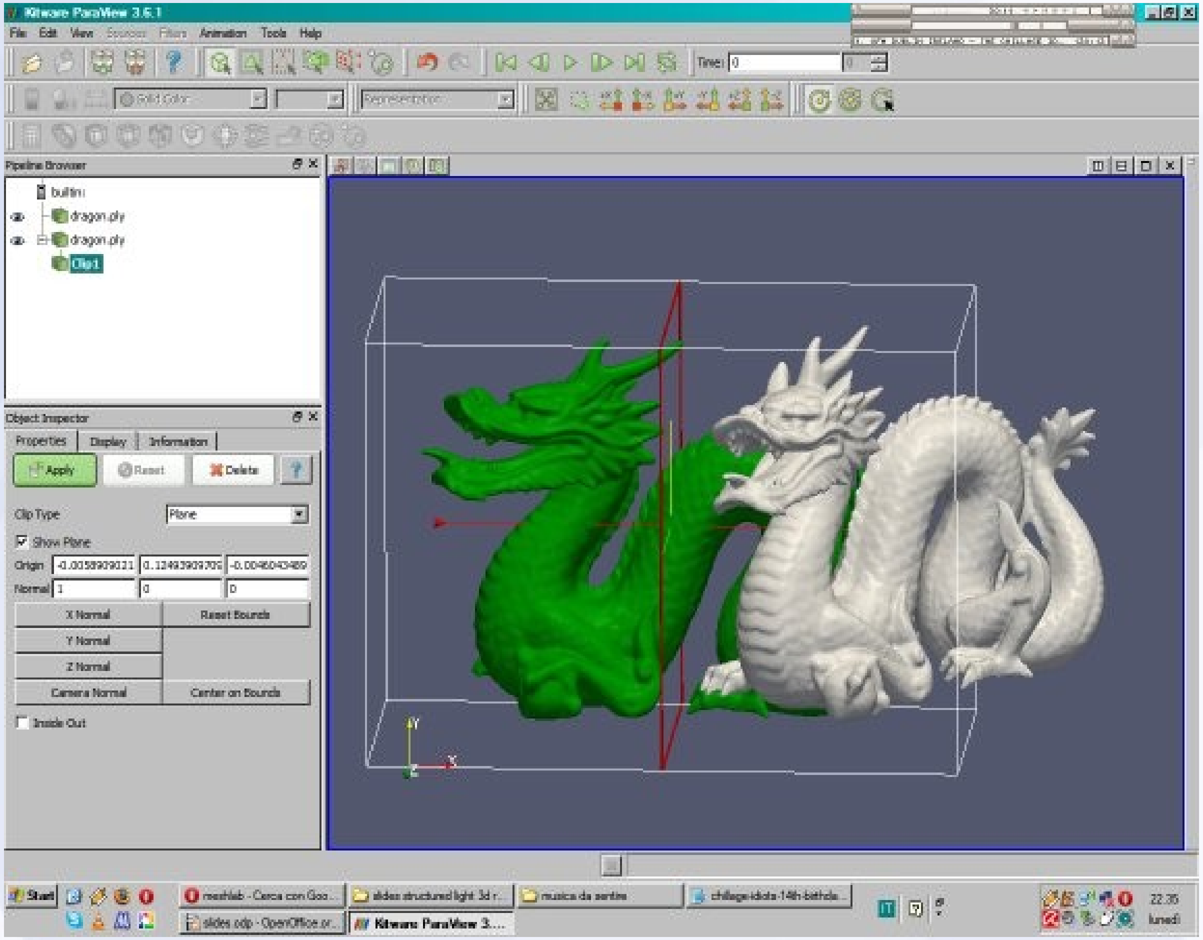


- Every single “range shot” lies in its own *local* coordinate frame
- Range data from the different viewpoints must be brought to a common world of coordinates (a *global* coordinate frame)

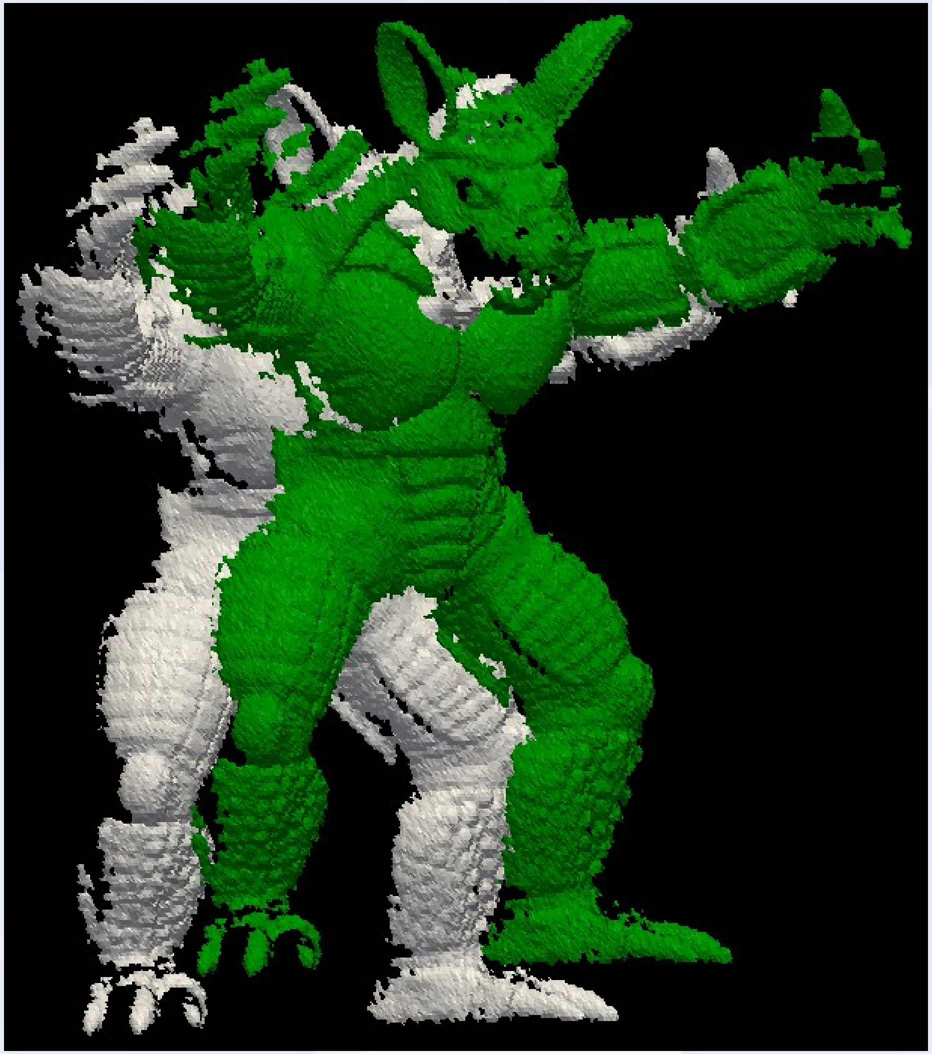
Registration (coarse)

An initial, approximate alignment is called *coarse registration*

- Hints might come from the acquisition process itself (e.g. known rotation among the views)
- Coarse alignment can be performed by hand if unsupervised reconstruction is not required



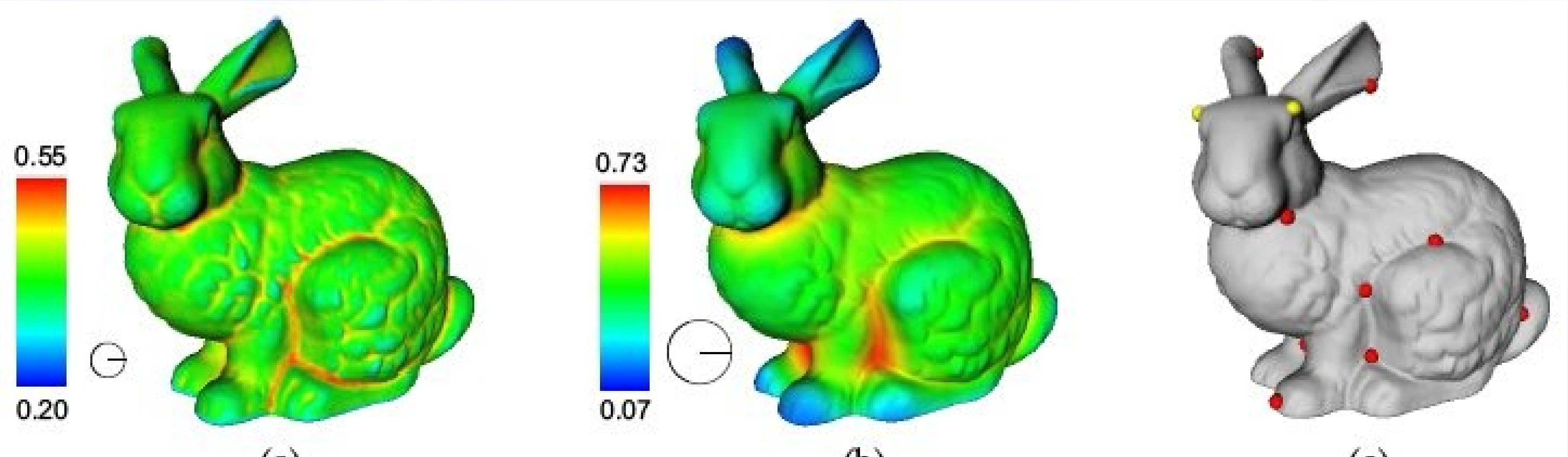
Registration (coarse)



<u>Automatic</u> alignment usually requires:

1. Detection of *interest points*
2. Computation of some *shape descriptor*
3. Establishing *matches* among patches
4. Computation of *optimal motion*

Interest points detection

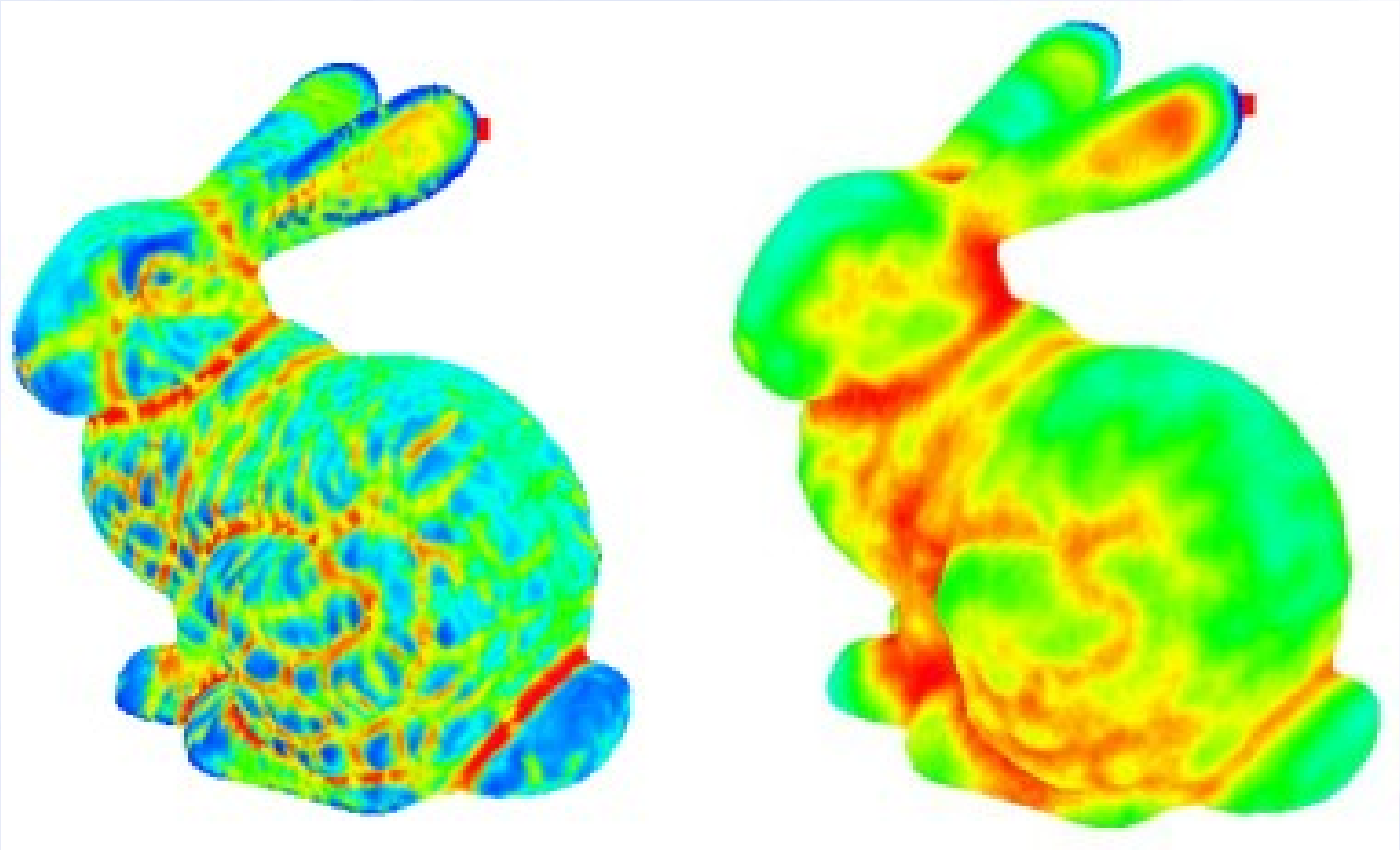


(a) (b) (c)

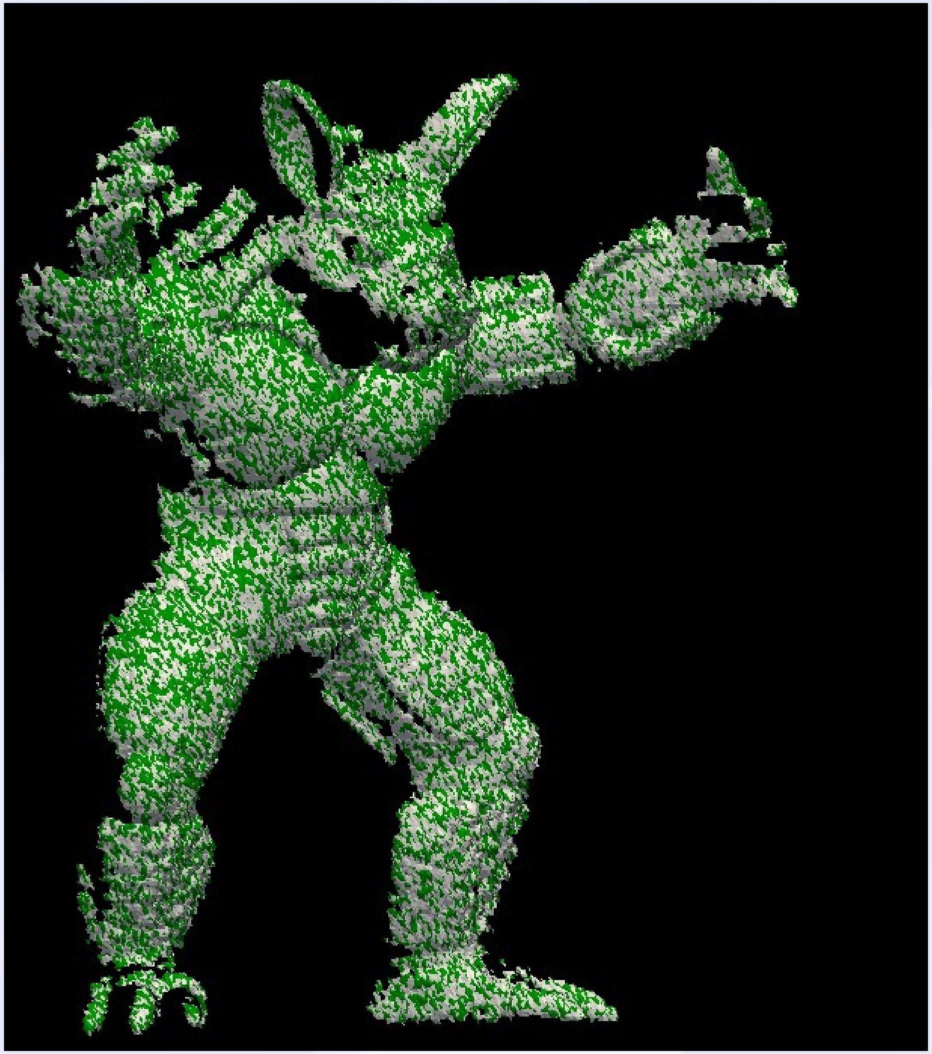
The local surface structure around the interest point is rich in terms of local information contents. Desirable properties are:

- rotation and translation *invariance*
- *robustness* to local and global perturbations, including:
  - scale changes (e.g. due to reconstruction ambiguity)
  - Gaussian noise (e.g. due to the acquisition process)
  - occlusions
- *repeatability*

Shape descriptors



Registration (fine)



Coarse alignment can be made more accurate by means of “fine registration” techniques

- Iterative Closest Point (ICP)
- Chen-Medioni
- Hybrid approaches

ICP

1) Initial transformation
2) For each data point, look for a *mate* on the model mesh
3) Estimate motion bringing the data points to their model mates
4) Apply the estimated motion and compute RMS error
5) If no stop conditions are satisfied, go to step 2)

Cons:

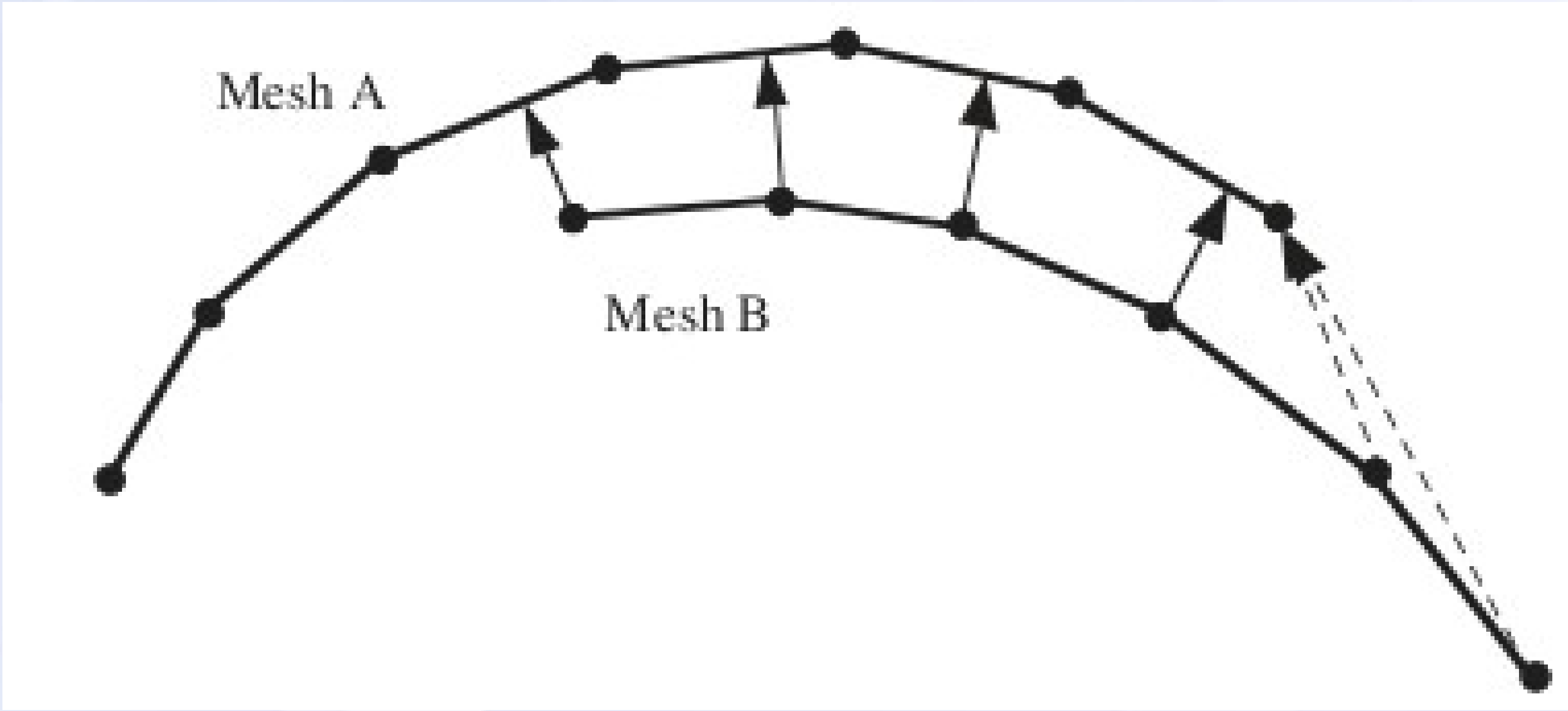
- Convergence time
- Local minima

ICP variations

- Sampling strategy

ICP variations

- Pairs selection / rejection (“mating”)



ICP variations

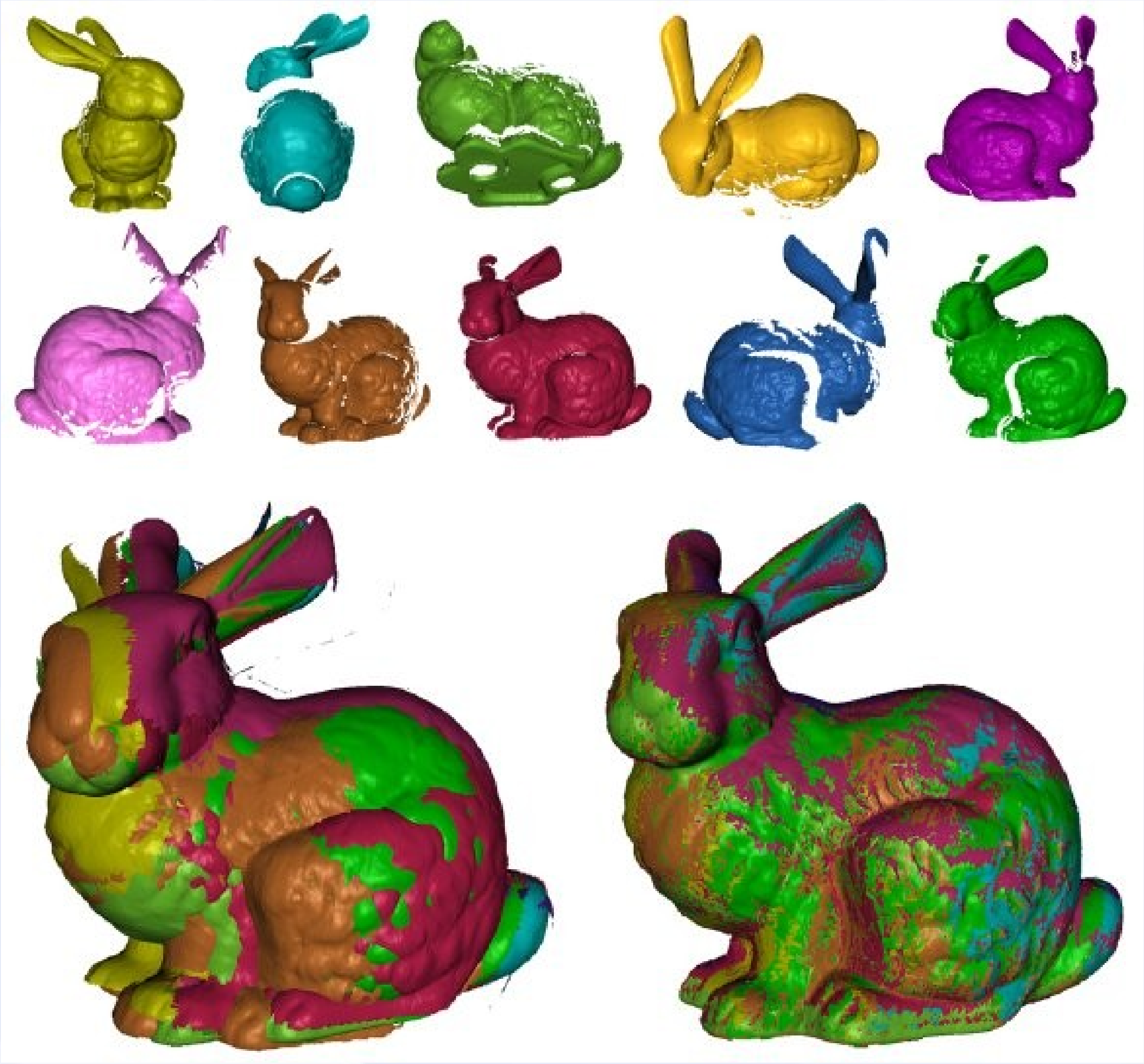
- Error metric to minimize

  Point-point distance, point-plane distance
- Exit conditions

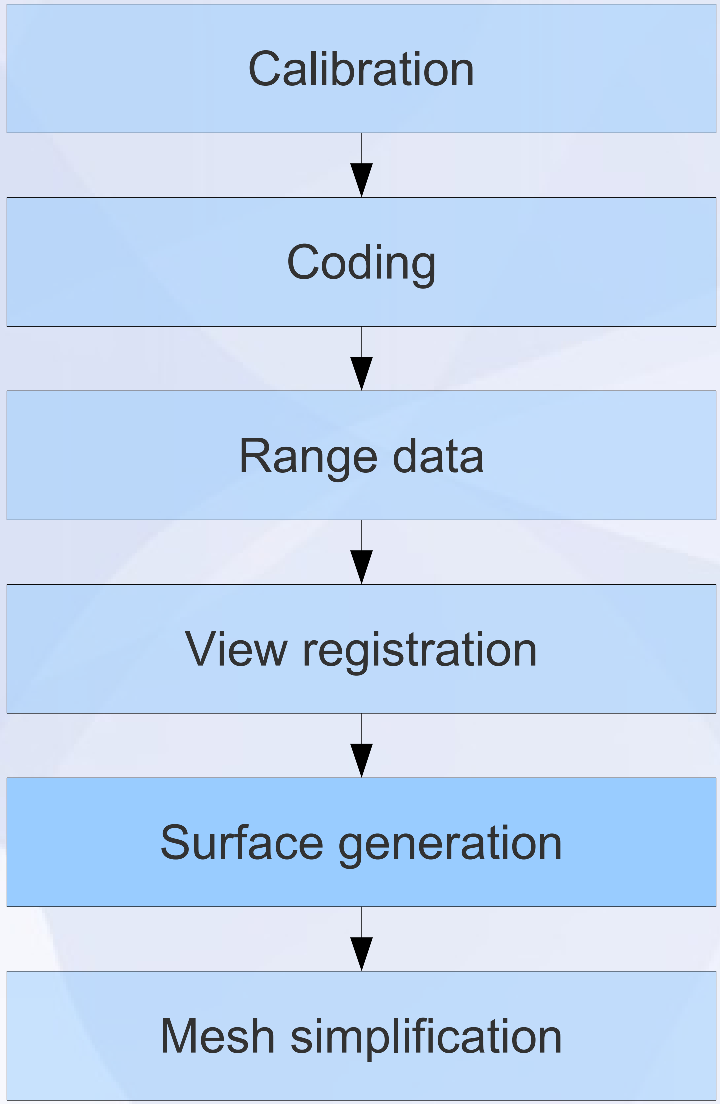
  # iterations, stable error, small error
- Motion computation

  Horn weighted vs non-weighted
- Combinations of the above

Registration



Reconstruction pipeline



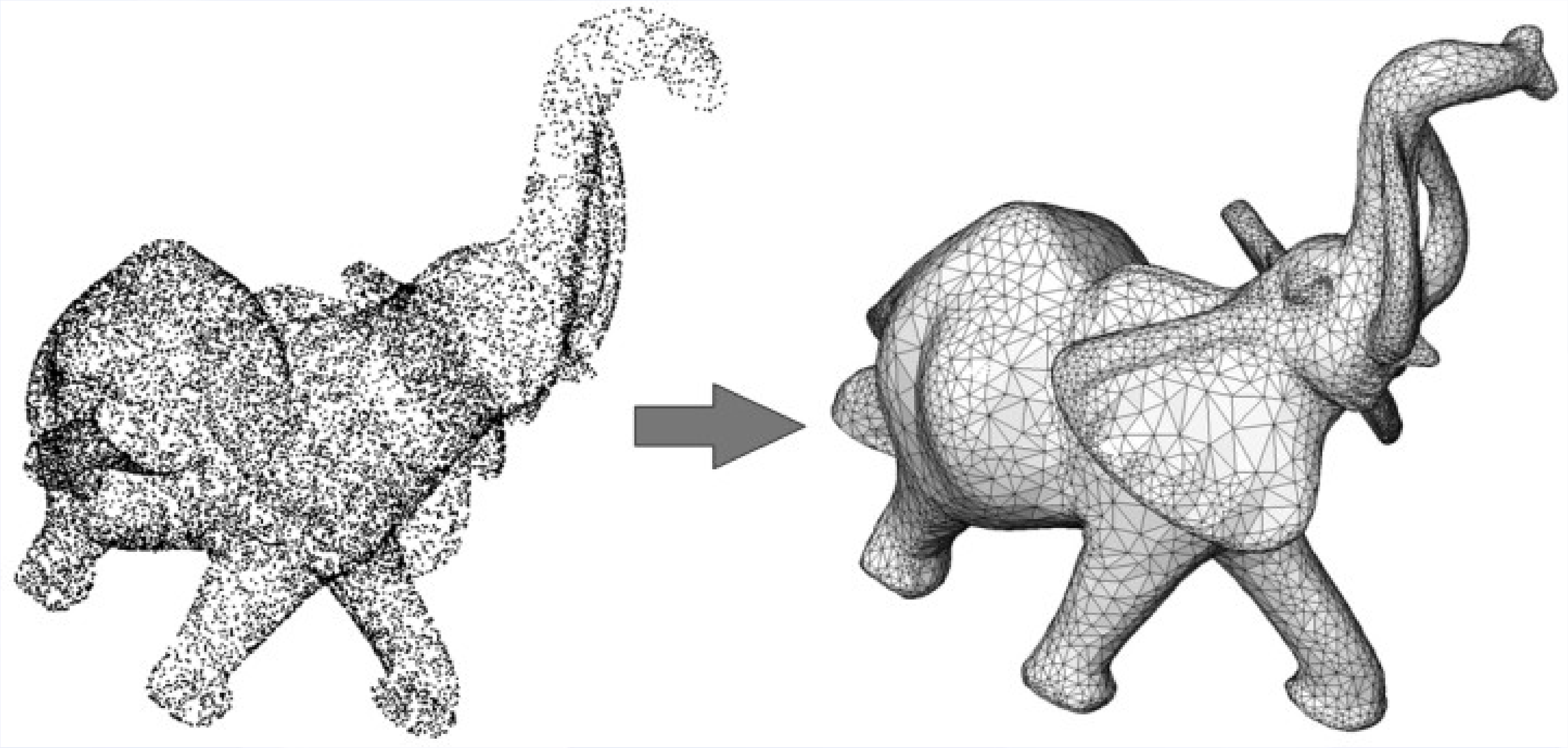
Surface generation

The registered rangemaps are now ready to produce a *single*, *water-tight*, *manifold* mesh.

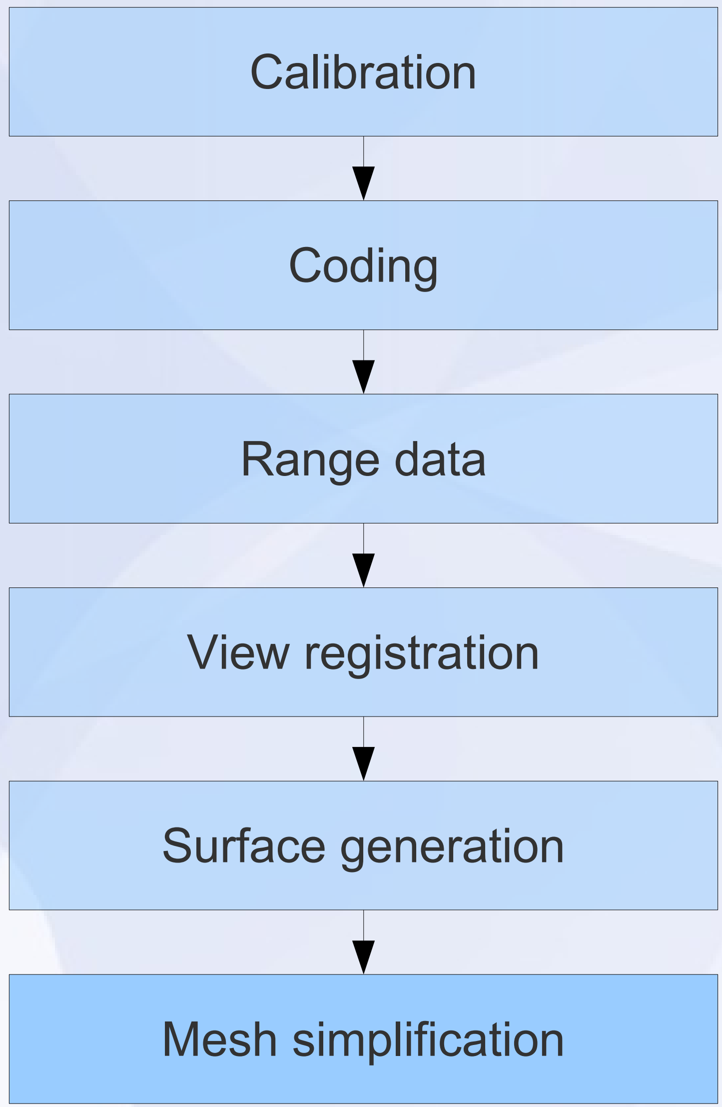
1. Integration
2. Surface extraction
3. Hole filling
4. Clean up

Surface reconstruction

Usually, geometry information is discarded at this stage, and oriented point sets are used.



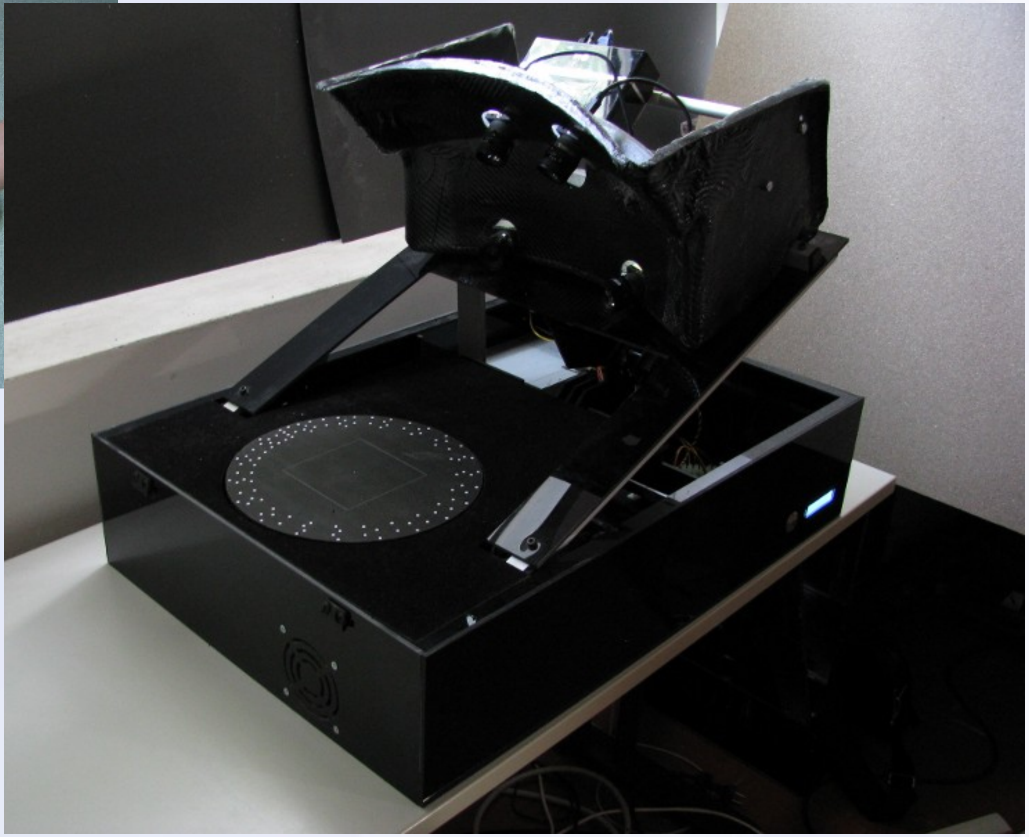
Reconstruction pipeline



Mesh simplification

The process of reducing the number of faces used in the surface while keeping the overall shape, volume and boundaries preserved as much as possible

Our (accurate :)) scanner



Reconstruction of a small object