Chair of Computer Graphics and Visualization

Insitute for Software and Multimedia Technology

Prof. Dr. Stefan Gumhold

Content

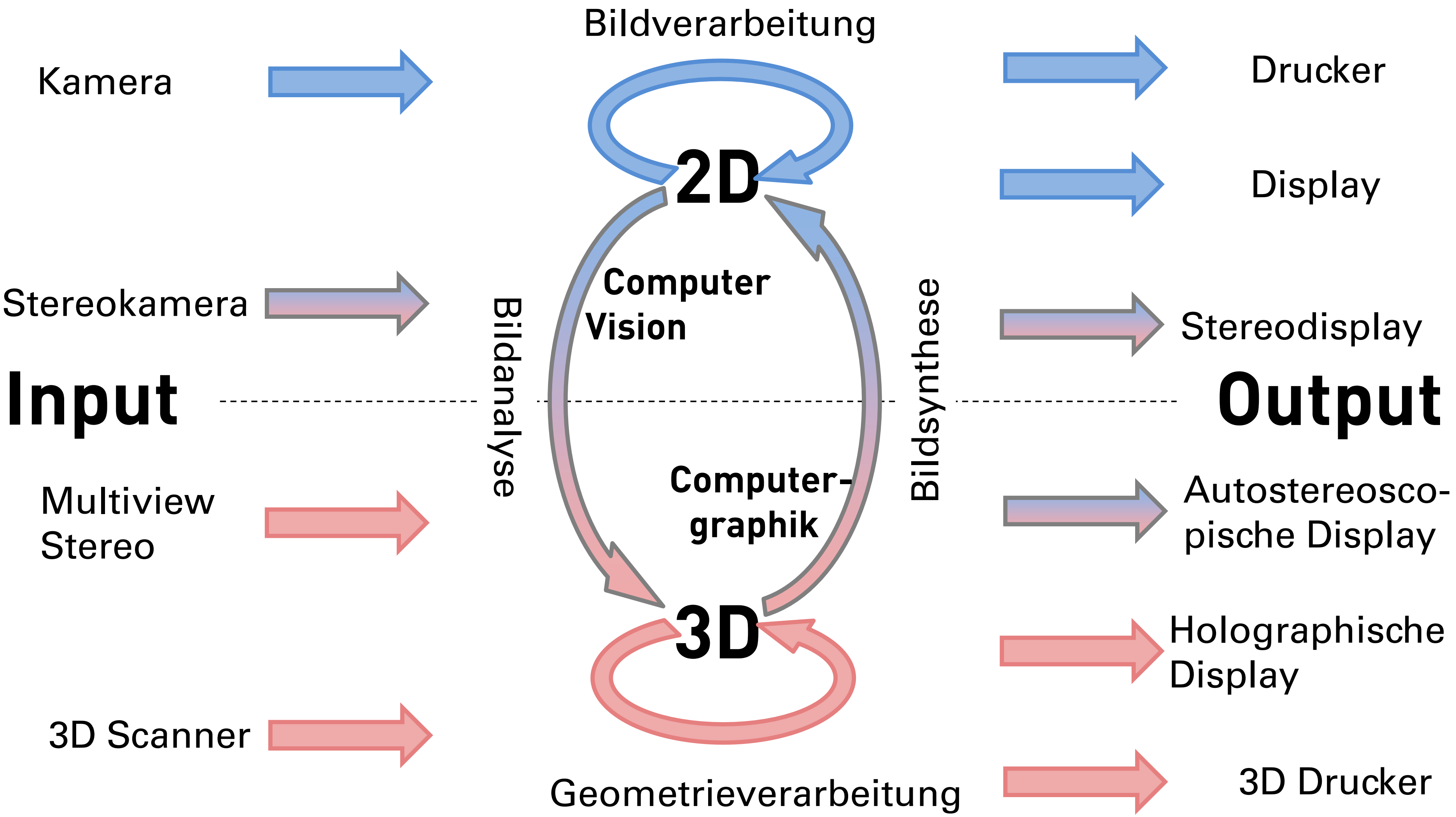
Computer Graphics and Visualization

- Overview
  - Computer Graphics
  - Scientific Visualization
  - Research Areas at CGV
  - Courses at CGV
- Example Research Questions
  - Normal Estimation for 3D Scanning
  - Efficient Particle Rendering for Visualization
  - Object Pose Estimation
- Oral Exam

OVERVIEW CG, SCIVIS, RESEARCH AND COURSES

Einordnung und Begriffsbestimmung
Übersicht der GDV

Computergraphik und Visualisierung



CG Areas and CGV Research

Computer Graphics and Visualization

Computer Graphics

Modelling | Rendering / Image Synthesis | Animation

Application Domains

Visualization	Engineering	Interaction	Entertainment
Natural Sciences	CAD/CAM	Operating Systems	Games
Medicine	Simulation	User Interaction	Special Effects
Geoinformatics	Round Trip	virtual Reality	Cartoons
Information	3D Scanning	augmented Reality	

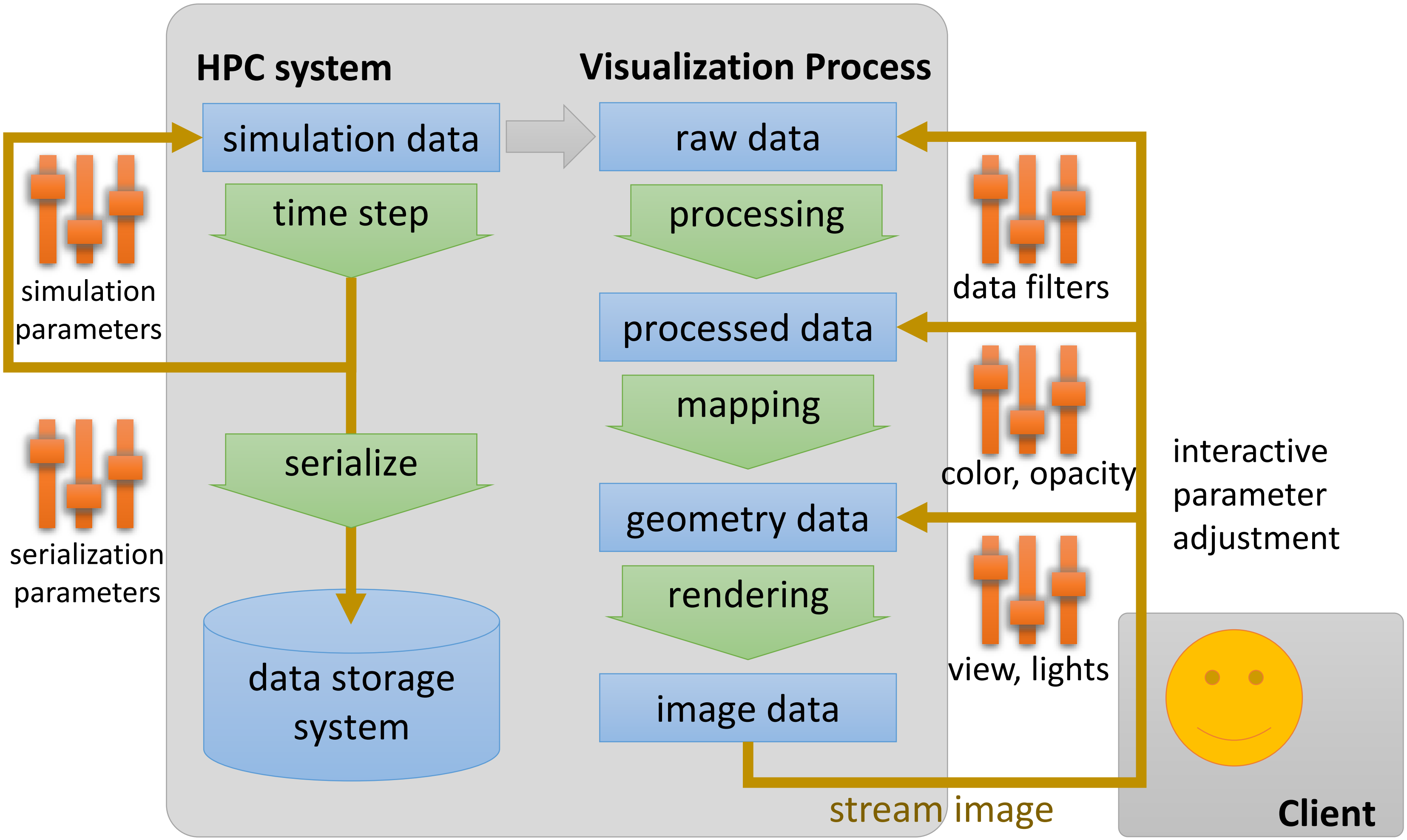
CGV research areas

Sample CG Research

Computer Graphics and Visualization

© Chai et al., [A Reduced Model for Interactive Hairs](), [SIGGRAPH 2014]()

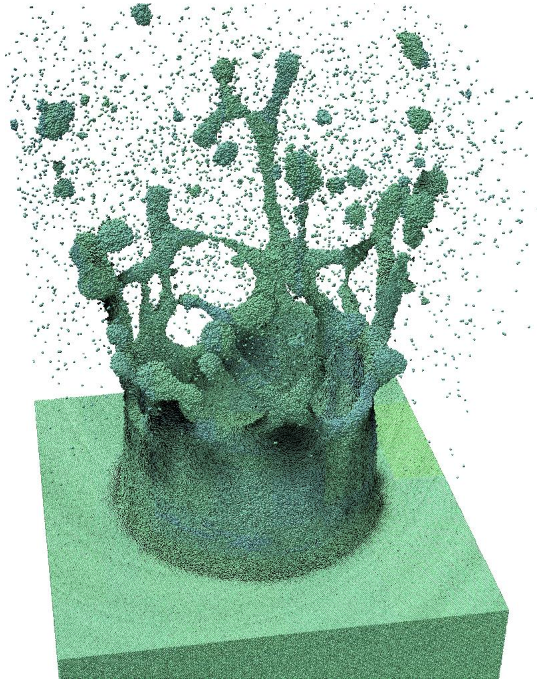
Scientific Visualization Pipeline



Sample Visualizations

Volumes
© Bruckner et al. 2005

Vector Fields
© Günther et al. 2013

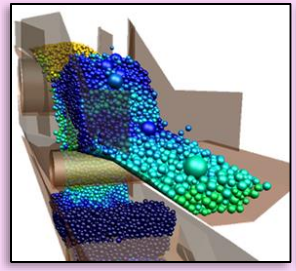
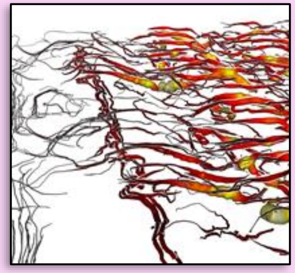


Particles
© Grottel 2014

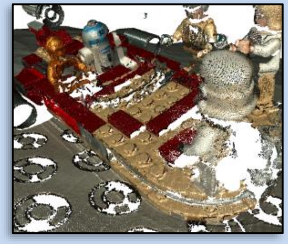
Overview – Research Areas at CGV

Geometry Processing

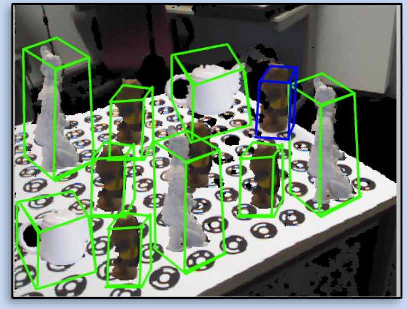

Scientific Visualization

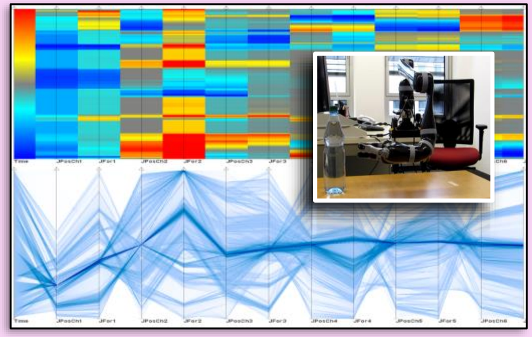
Scene Acquisition



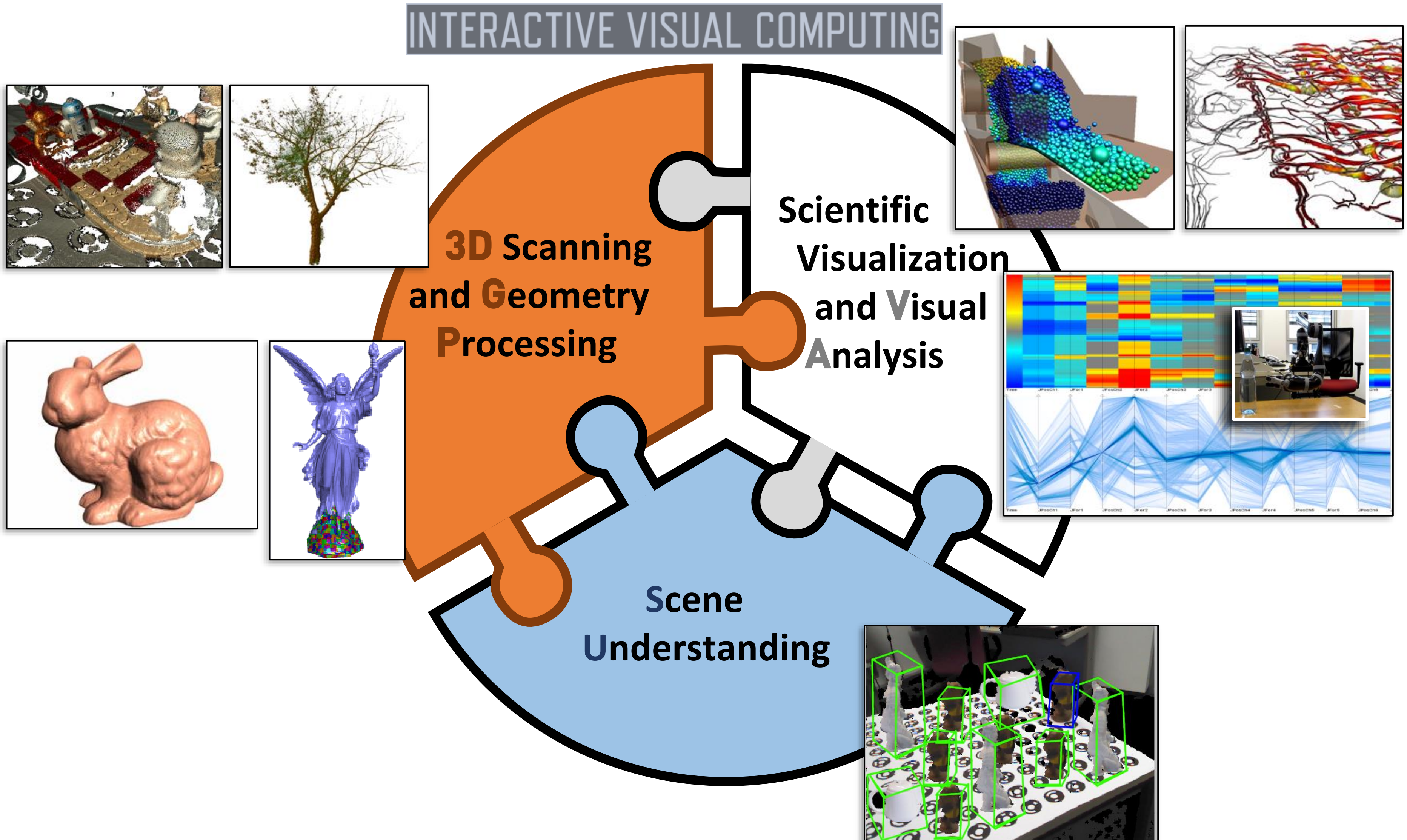

Scene Understanding



Visual Analysis



Overview – Research Areas at CGV



Overview – Course at CGV

	WS	SS	BAS-7	VERT7	VMI-8	INF-PM-FOR/ANW
ECG		X				
CG1	X		X			
CG2		X	X	X	X	X
CG3	X			X	X	X
WV	X		X			
CPP4CG		X	X			

- ECG … Einführung Computergraphik
- CG1 … Computer Graphics 1 – enabling technologies
- CG2 … Computer Graphics 2 – acquisition & geometry processing
- CG3 … Computer Graphics 3 – physical based graphics
- WV … Wissenschaftliche Visualisierung
- CPP4CG … C++-Programmierung für Computergraphik

Course Content

Computer Graphics and Visualization

- ECG
  - Modellierung
  - Rasterisierung
  - Transformationen
  - Basic GPU Prog.
  - Kurven
  - Beleuchtung
  - Raytracing
- CG4CPP
  - usage of IDE
  - polymorphism
  - Templates, stl
  - exceptions
  - advanced features
- CG1
  - Grand Tour
  - Modern GPU Prog.
  - Mesh Processing
  - Acceleration-DS
  - Optimization
- CG2 – Geometry
  - Representations (implicit surfaces subdivision surf.)
  - Inverse Kinematics
  - Rigging & Skinning
  - 3D Scanning
- SciVis
  - Foundations
  - Preprocessing
  - Mapping
  - Volumes
  - Vector Fields
  - Particles
- CG3 - Physical
  - Physics for CG
  - Global Illumination
  - Rigid Bodies
  - Fluids

EXAMPLE RESEARCH QUESTIONS

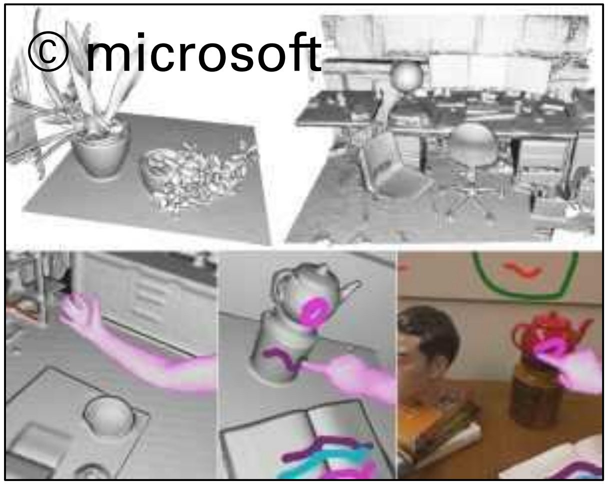
Normal Estimation for 3D Scanning

Efficient Particle Rendering for Visualization

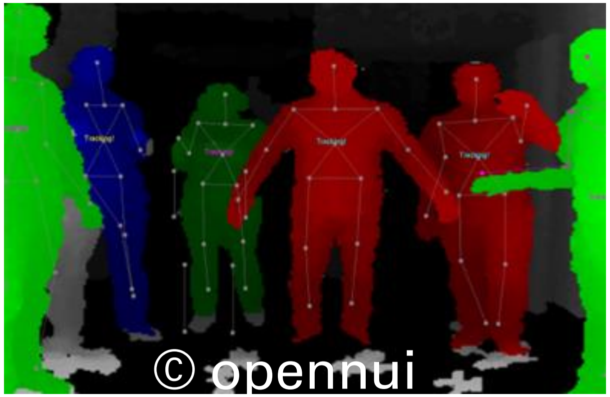
Object Pose Estimation

3D Acquisition

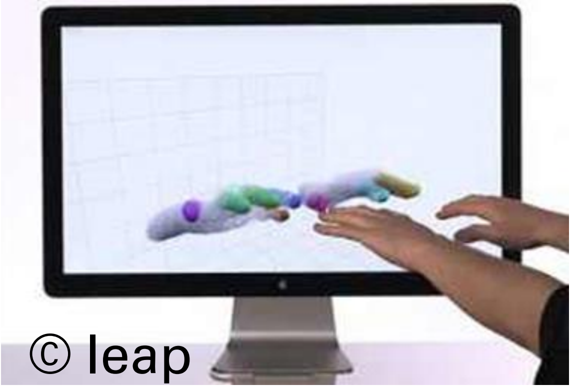
commercial realtime 3D scanning devices



kinect fusion



skeleton tracking

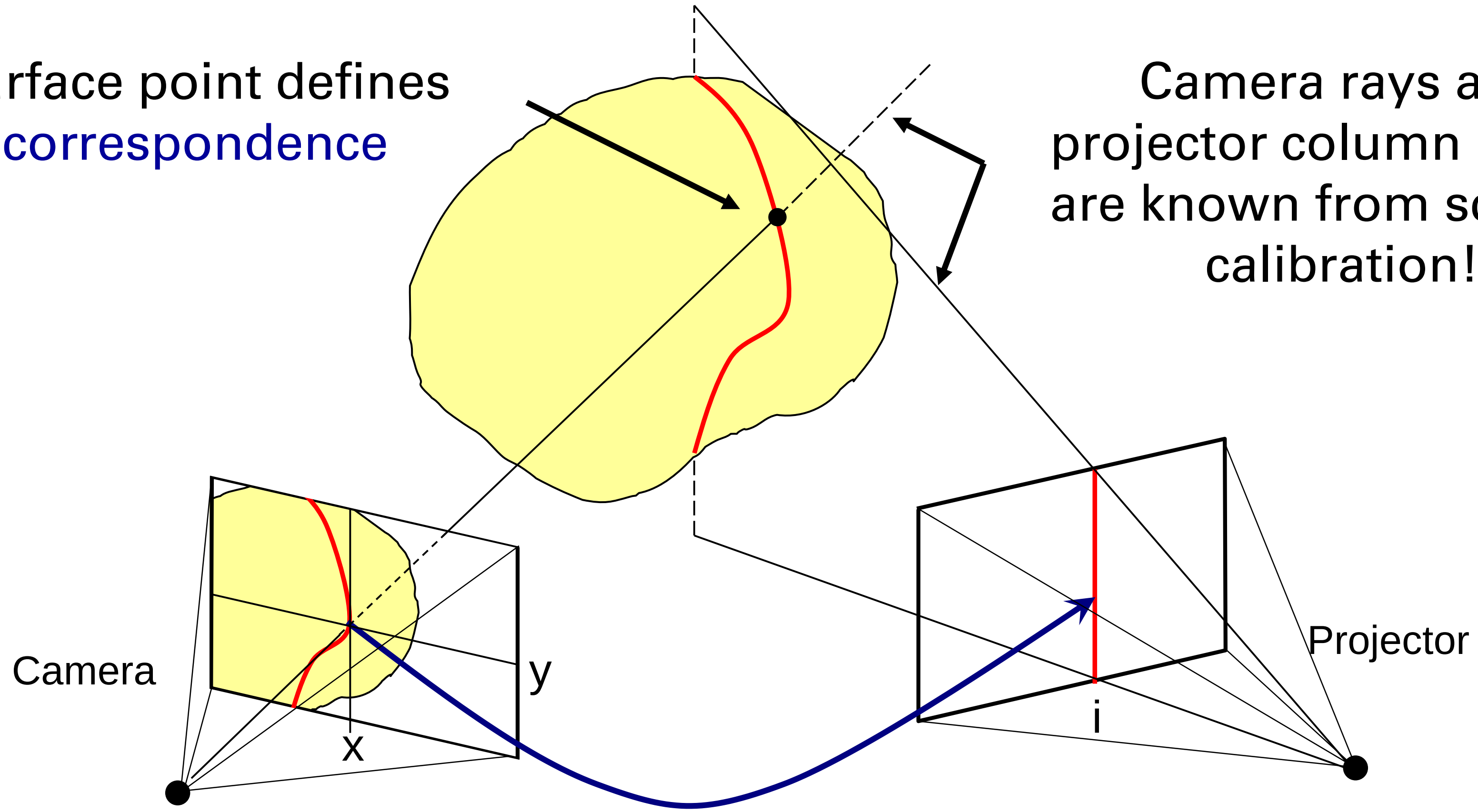


hand tracking

Structured Light Setup

Surface point defines correspondence

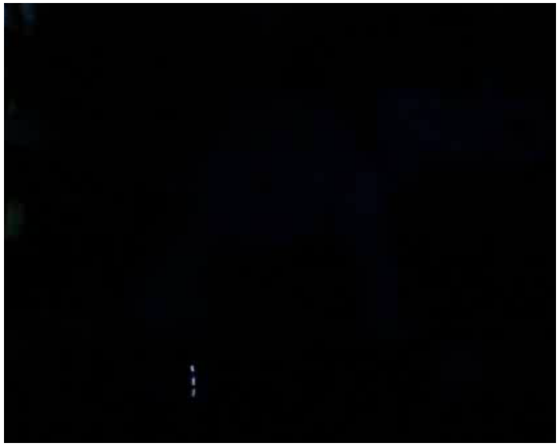
Camera rays and projector column planes are known from scanner calibration!



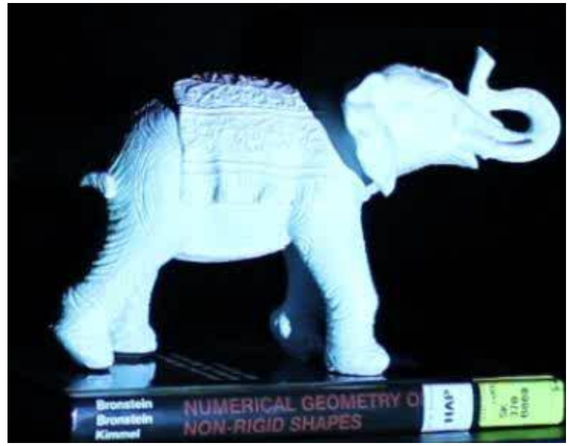
Reconstruction: Solve Correspondence problem Camera pixel → projector column + Ray-Plane Intersection

Column Coding Techniques

Computer Graphics and Visualization



lineshift

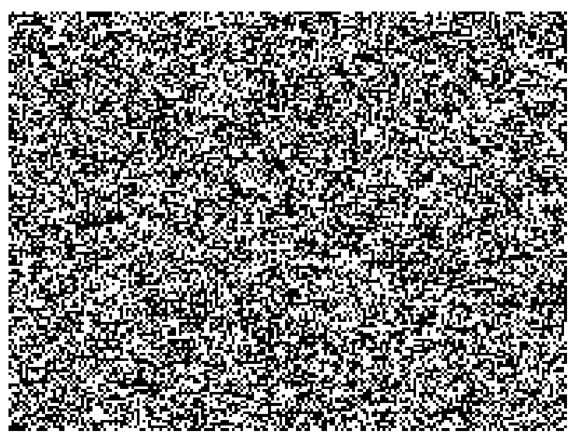


binary code

hierarchical phaseshift

Motion Compensation for Dynamic Case

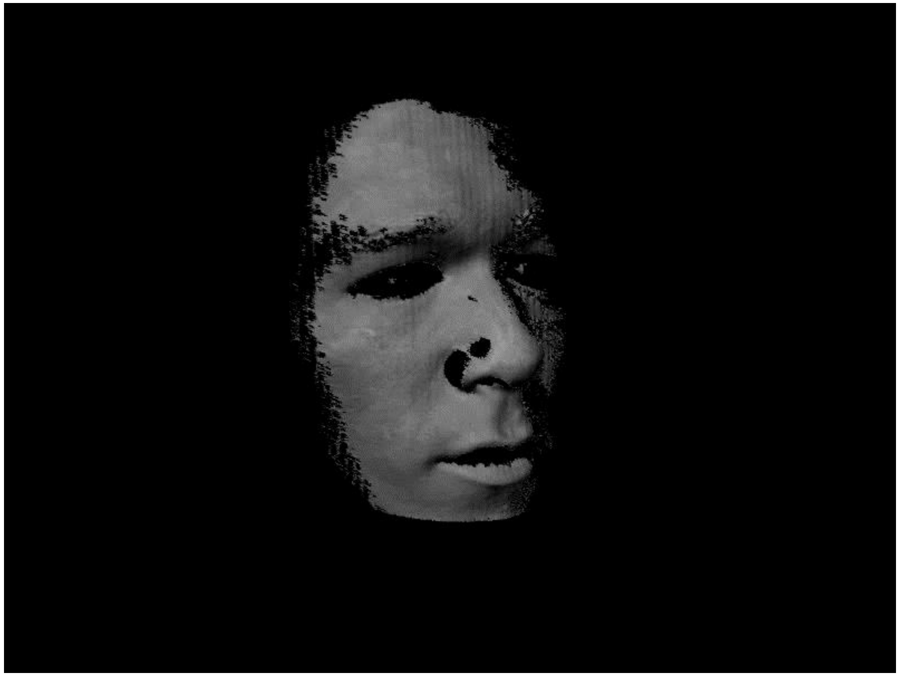
no compensation



motion compensation pattern

with compensation

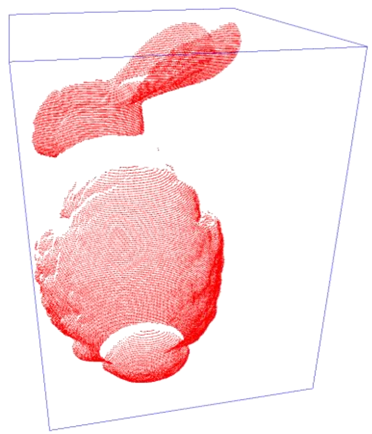
Sample Dynamic 3D Scans



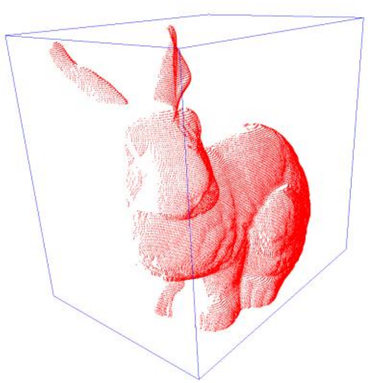
dynamic face scane

dynamic hand scan

Shape Reconstruction Pipeline

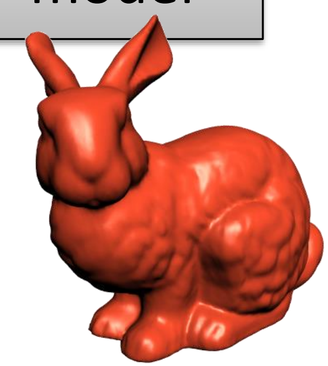


multiple 3D scans



- filter noise & outliers
- register scans
- estimate & orient normals
- reconstruct or fuse
- map colors or reconstruct BRDF

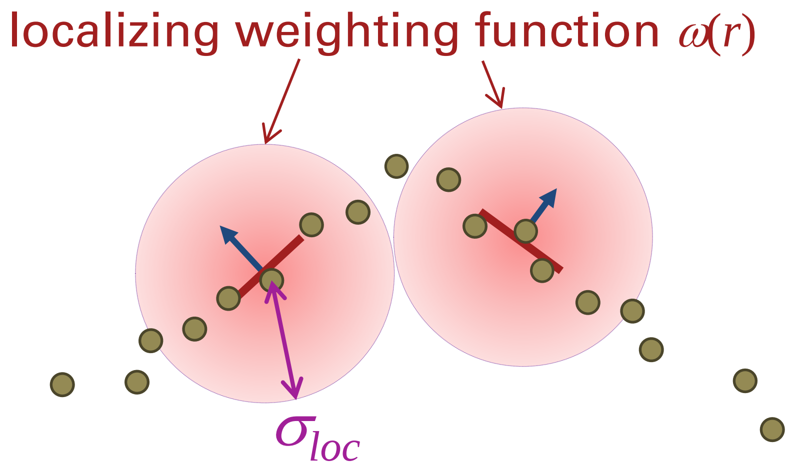
3D shape model



Estimation of Tangent Plane

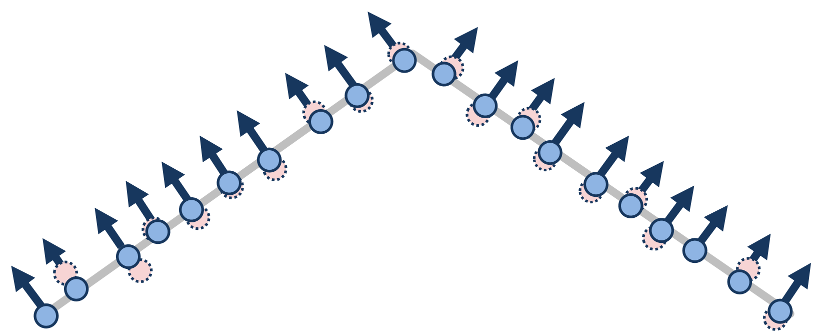
Localized Fit

- For each point in 3D scan estimate a local tangent plane by some weighted fitting procedure
- Crutial is to choose the correct width σ_{loc} of the weighting function

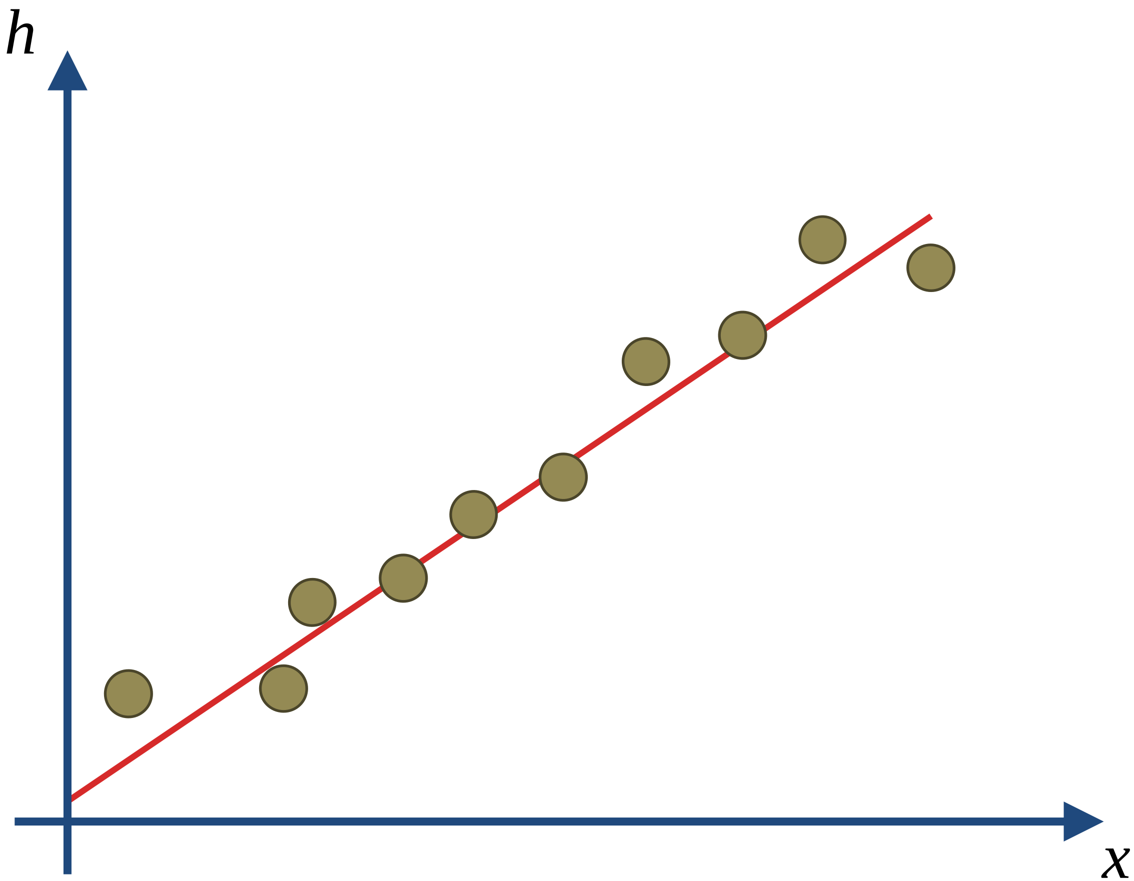


Denoising

- Project each point onto ist local tangent plane for denoising



Fit Line to Heights



data are k pairs:

$$(x_i, h_i)$$

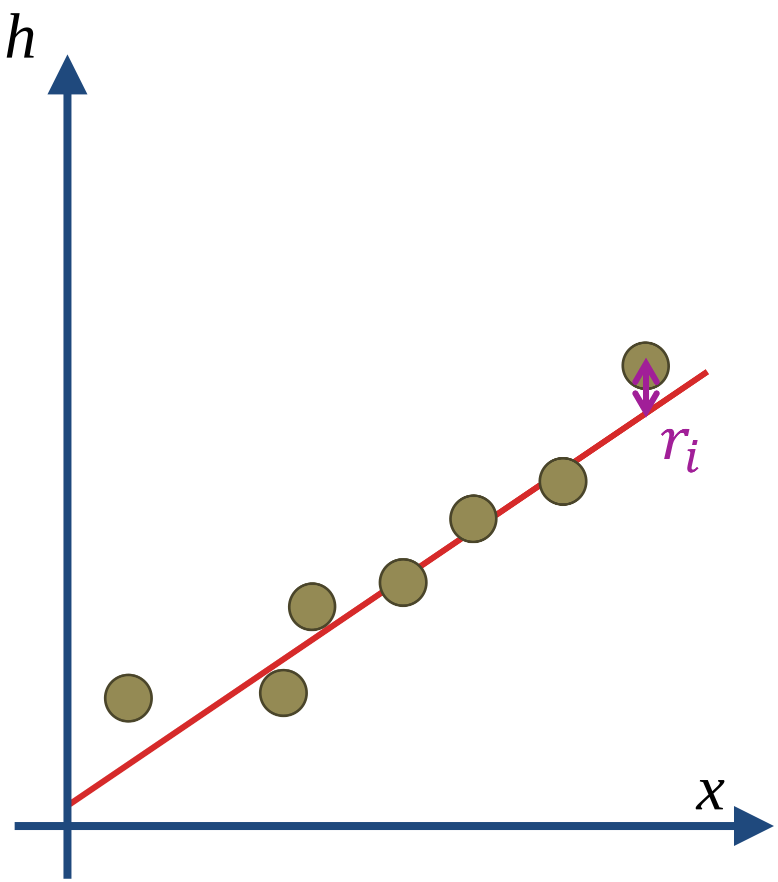
model is line:

$$h(x) = mx + c$$

- Goal: find model parameters $\theta = (m, c)$ of line that best fits the data

Least Squares of Height Differences

Computer Graphics and Visualization



data are k pairs: (x_i, h_i)

model is line: $h(x) = mx + c$

residuals form heights

$$r_i = h(x_i) - h_i$$

fit energy from sum of squares

$$E(m,c) = \sum_i r_i^2$$

least squares approach

$$\theta^* = (m^*, c^*) = \text{minarg}_{(m,c)} E(m,c)$$

Normal Equations

Computer Graphics and Visualization

data are k pairs: (x_i, h_i)

model is line: $h(x) = mx + c$

residuals: $r_i = mx_i + c - h_i$

fit energy: $E(m,c) = \sum_i r_i^2$

normal equations:

$$0 = \frac{\partial}{\partial m}E(m,c) = \sum_i 2r_i x_i = 2\sum_i (mx_i^2 + cx_i - h_i x_i)$$

$$0 = \frac{\partial}{\partial c}E(m,c) = \sum_i 2r_i = 2\sum_i (mx_i + c - h_i)$$

some shortcuts:

$$S_x = \sum_i x_i \qquad S_h = \sum_i h_i \qquad S_{xx} = \sum_i x_i^2$$

$$\bar{x} = \frac{1}{k}S_x \qquad \bar{h} = \frac{1}{k}S_h \qquad S_{hx} = \sum_i h_i x_i$$

$$0 = mS_{xx} + cS_x - S_{hx}$$

$$0 = S_x + kc - S_h$$

normal equations

Transform – Centering of Average

Computer Graphics and Visualization

data are k pairs: (x_i, h_i)

model is line: $h(x) = mx + c$

residuals: $r_i = mx_i + c - h_i$

fit energy: $E(m,c) = \sum_i r_i^2$

normal equations:

$$0 = mS_{xx} + cS_x - S_{hx}$$

$$0 = S_x + kc - S_h$$

compute averages:

$$\bar{x} = \tfrac{1}{k}S_x \qquad \bar{h} = \tfrac{1}{k}S_h$$

transform data:

$$(x'_i, h'_i) = \left(x_i - \bar{x}, h_i - \bar{h}\right)$$

sums vanish

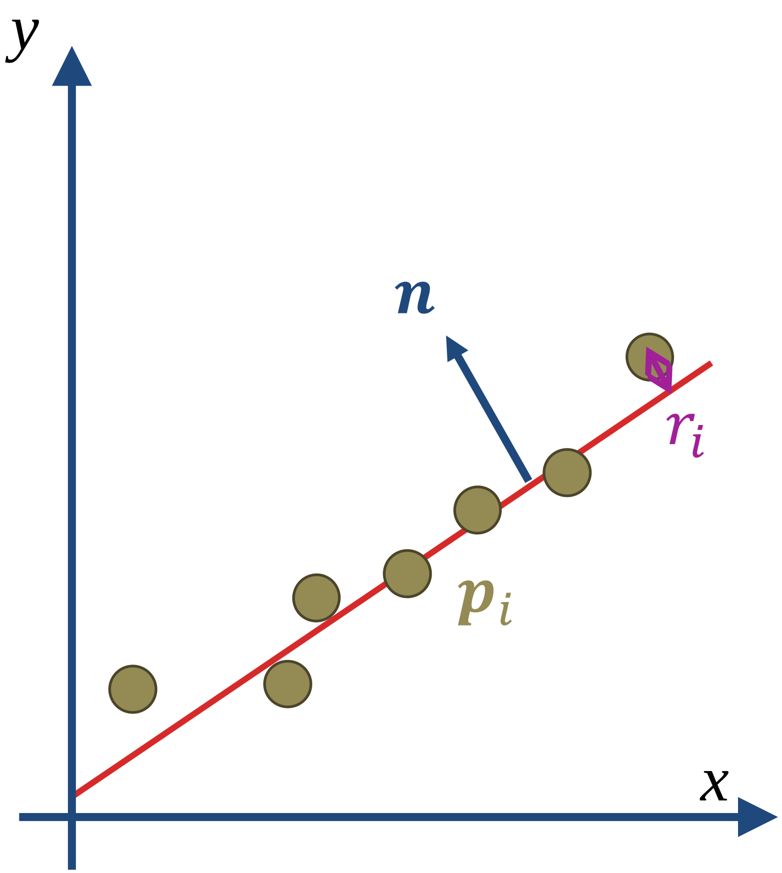
$$S_{x'} = 0 \qquad S_{h'} = 0$$

normal equations become:

$$c' = 0 \qquad m' = \frac{S_{h'x'}}{S_{x'x'}}$$

➡ by centering to averages constant term c‘ vanishes

Orthogonal distances



data are k points: $\boldsymbol{p}_i = (x_i, y_i)$

Hessian form with normal $\boldsymbol{n}$:

$$0 = \langle \boldsymbol{n}, \boldsymbol{p} \rangle + c$$

$$\|\boldsymbol{n}\| = 1$$

residuals form orthogonal dist.:

$$r_i = \langle \boldsymbol{n}, \boldsymbol{p}_i \rangle + c$$

transformation: $\boldsymbol{p}'_i = \boldsymbol{p}_i - \bar{\boldsymbol{p}}$

without proof ➡ $c' = 0$

least squares approach

$$\boldsymbol{n}'^* = \operatorname{minarg}_{\boldsymbol{n}' | \|\boldsymbol{n}'\|=1} \underbrace{E(\boldsymbol{n}')}_{\sum_i r'^2_i}$$

Solve Orthogonal Least Squares Fit

Computer Graphics and Visualization

k data points: $\boldsymbol{p}_i = (x_i, y_i)$

transformed: $\boldsymbol{p}'_i = \boldsymbol{p}_i - \overline{\boldsymbol{p}}$

reduced model: $0 = \langle \boldsymbol{n}', \boldsymbol{p}' \rangle$, $\|\boldsymbol{n}'\| = 1$

residuals: $r_i = \langle \boldsymbol{n}', \boldsymbol{p}'_i \rangle$

$$S_{\boldsymbol{p}'} = \sum_i \boldsymbol{p}'_i$$

$$S_{\boldsymbol{p}'\boldsymbol{p}'} = \sum_i \boldsymbol{p}'_i \boldsymbol{p}'^T_i$$

fit energy $$E(\boldsymbol{n}') = \sum_i r_i^2 = \sum_i \langle \boldsymbol{n}', \boldsymbol{p}'_i \rangle^2 = \boldsymbol{n}'^T S_{\boldsymbol{p}'\boldsymbol{p}'} \boldsymbol{n}'$$

$$\langle \boldsymbol{n}', \boldsymbol{p}'_i \rangle^2 = \boldsymbol{n}'^T \boldsymbol{p}'_i \boldsymbol{p}'^T_i \boldsymbol{n}'$$

$$\boldsymbol{p}'_i \boldsymbol{p}'^T_i = \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} \begin{pmatrix} x'_i & y'_i \end{pmatrix} = \begin{pmatrix} x'^2_i & x'_i y'_i \\ x'_i y'_i & y'^2_i \end{pmatrix}$$

least squares fit: $\boldsymbol{n}'^* = \text{minarg}_{\boldsymbol{n}' | \|\boldsymbol{n}'\| = 1} \boldsymbol{n}'^T S_{\boldsymbol{p}'\boldsymbol{p}'} \boldsymbol{n}'$

⇨ $\boldsymbol{n}'^*$ is eigenvec. to smallest eigenval.

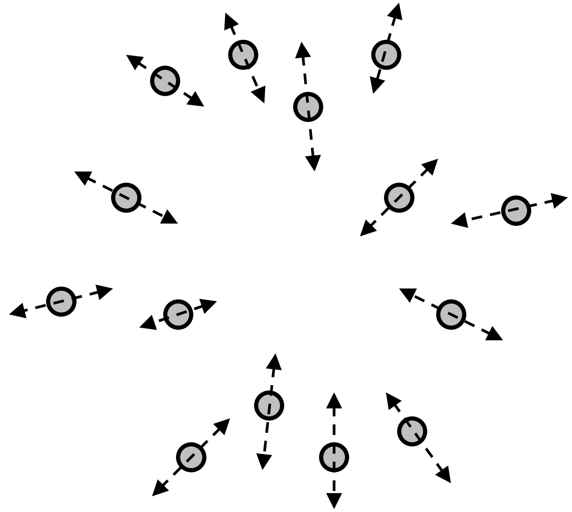
$$\boldsymbol{n}'^* = \pm EV\left(S_{\boldsymbol{p}'\boldsymbol{p}'}\right)_0$$

Computer Graphics and Visualization

Discussion of Orthogonal Fit

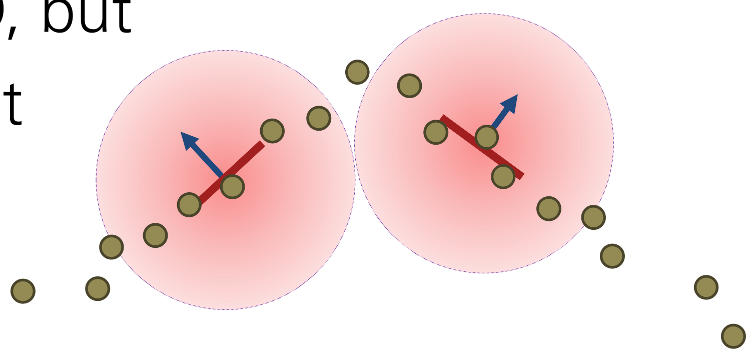
Normal Orientation

- sign of normal is called orientation
- orientation defines inside of shape
- globally consistent orientation needed
- scanner location defines orientation



3D Case

- all generalizes nicely to 3D or nD, but
- single line / plane is not sufficient
- we need some technique to localize the fit



Weighted Least Squares

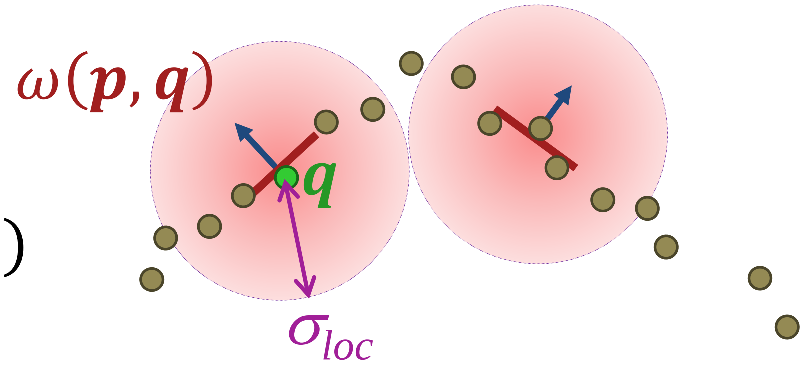
Computer Graphics and Visualization

reference point: $\boldsymbol{q}$

weight function: $\omega(\boldsymbol{p},\boldsymbol{q})$

$\omega_i(\boldsymbol{q}) = \omega(\boldsymbol{p}_i,\boldsymbol{q})$

influence scale: σ_{loc}



often Gaussian weights:

$$\omega(\boldsymbol{p},\boldsymbol{q}) = \theta(\|\boldsymbol{p}-\boldsymbol{q}\|),\ \theta(d) = \exp\left(-d^2/\sigma_{\mathrm{loc}}^2\right)$$

weighted fit energy: $E(\boldsymbol{n}'(\boldsymbol{q})) = \sum_i \omega_i(\boldsymbol{q}) r_i^2$

only few changes:

$$S_\omega = \sum_i \omega_i\,,\quad S_{\boldsymbol{p}} = \sum_i \omega_i \boldsymbol{p}_i\,,\quad \bar{\boldsymbol{p}} = \frac{S_{\boldsymbol{p}}}{S_\omega},\quad S_{\boldsymbol{pp}} = \sum_i \omega_i \boldsymbol{p}_i \boldsymbol{p}_i^T$$

Discussion of Localized Fit

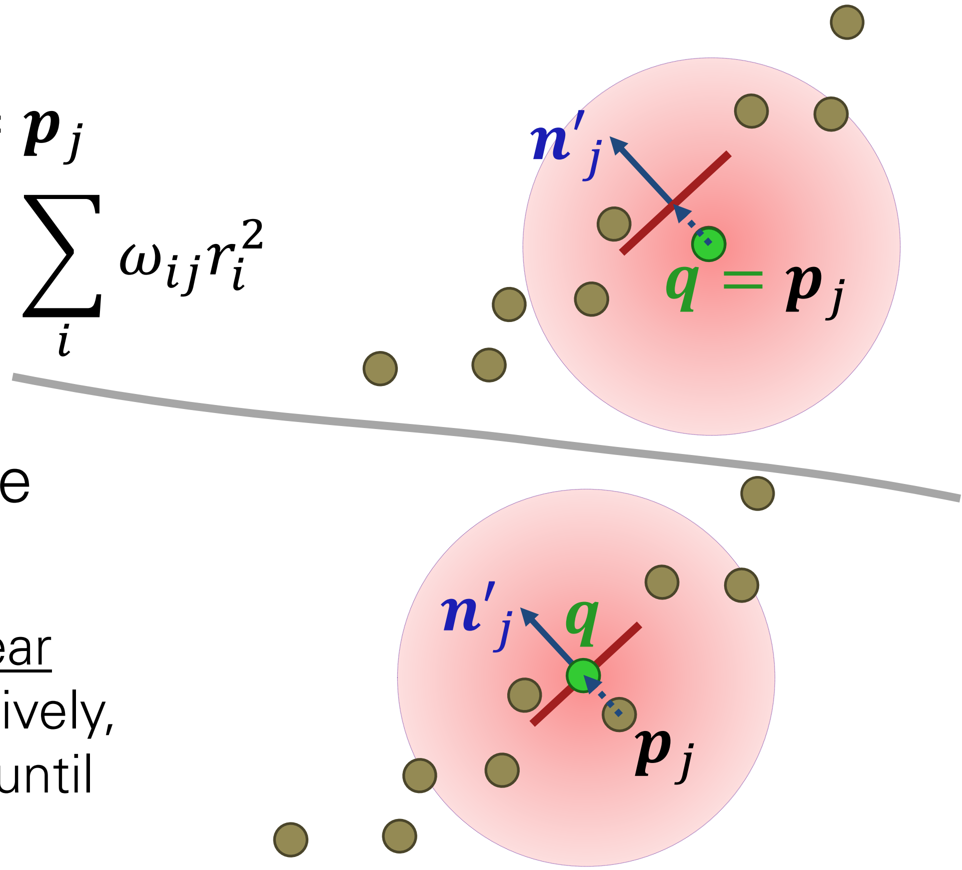
Computer Graphics and Visualization

choice of reference point

- choose each data point $\boldsymbol{q} = \boldsymbol{p}_j$

$$\omega_{ij} = \omega(\boldsymbol{p}_i, \boldsymbol{p}_j), \quad E(\boldsymbol{n}'_j) = \sum_i \omega_{ij} r_i^2$$

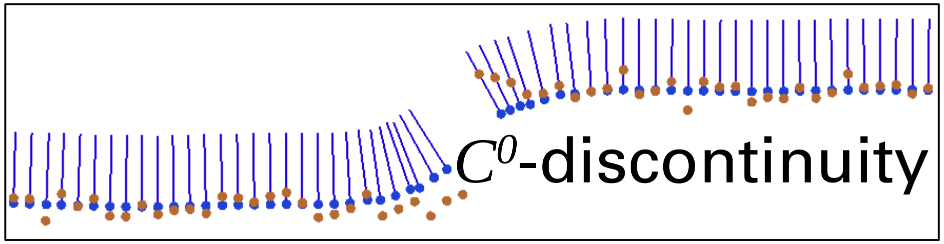
- choose projection $\boldsymbol{q}$ of data point $\boldsymbol{p}_j$ onto resulting plane as reference point
  - fit problem becomes non-linear and needs to be solved iteratively, by recomputing the weights until convergence
  - besides $\boldsymbol{p}_j$ one can choose any initial point $\boldsymbol{p}$
  - all points that project onto themselves form a C^∞-contineous surface called the Moving Least Squares (MLS) surface



Discussion of Localized Fit

Problems

- outliers and C^0-discontinuities significantly influence fit

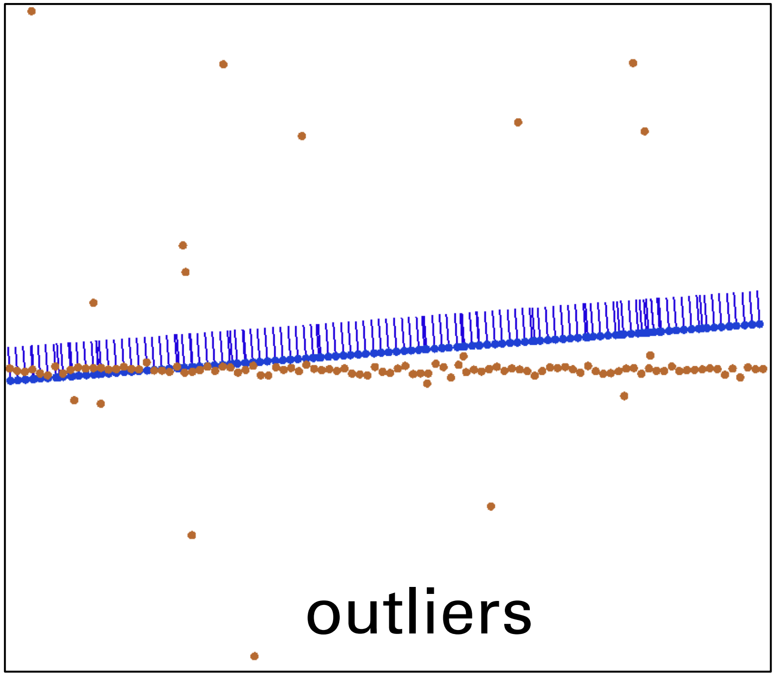


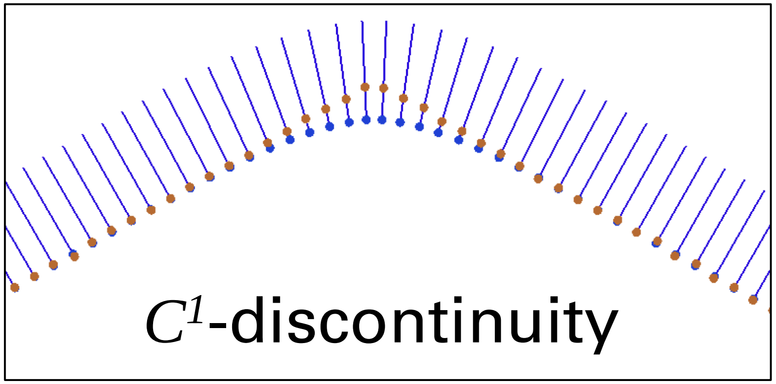
- C^1-discontinuities are smoothed out

Solution?

- use robust norm $\rho(r)$

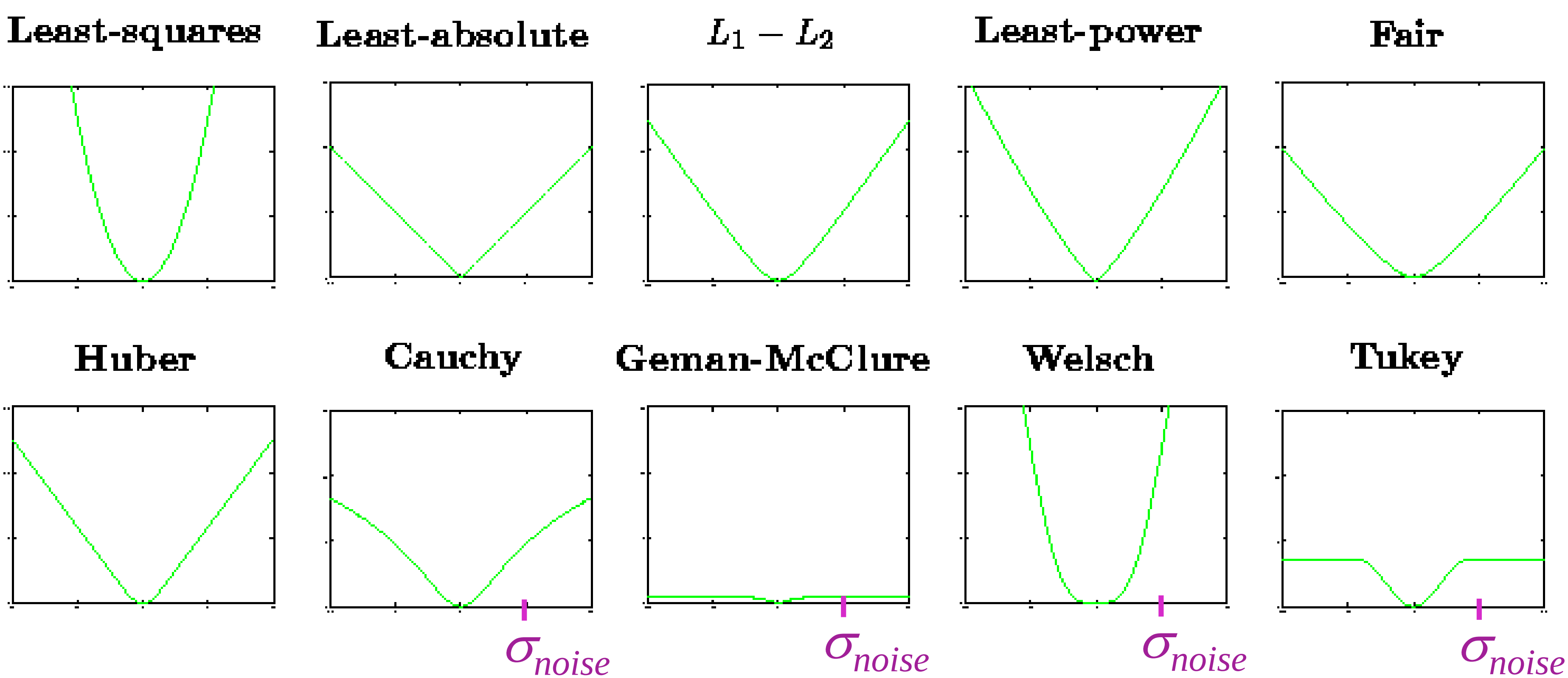
$$E(\mathbf{n}') = \sum_i \rho(r_i)$$





Zoo of Robust Norms

- for Gaussian noise, the L_2-norm $\rho(r) = r^2$ is optimal
- Otherwise a large number of norms can be chosen from, which need to be scaled by the noise scale σ_{noise}:



M-estimation with Iterated Re-weighted Least Squares

Computer Graphics and Visualization

- Idea: choose weights not for localization but to emulate robust norm with weighted least squares fit

robust norm: $\rho(r)$ influence function: $\psi(r) = \frac{\partial\rho}{\partial r}(r)$

Weighted LS

$$E = \frac{1}{2}\sum_i \varpi_i r_i^2$$

$$\frac{\partial E}{\partial\theta} = \sum_i \varpi_i r_i \frac{\partial r_i}{\partial\theta}$$

Robust LS

$$E = \sum_i \rho(r_i)$$

$$\frac{\partial E}{\partial\theta} = \sum_i \frac{\partial\rho}{\partial r}(r_i)\frac{\partial r_i}{\partial\theta}$$

$\varpi_i r_i = \psi(r_i)$ ➡ weight function: $\varpi_i = \frac{\psi(r_i)}{r_i}$

IRWLS:

$$WLS(\varpi_i^0 \equiv 1) \Rightarrow r_i^0 \Rightarrow \varpi_i^1 = \psi(r_i^0)/r_i^0$$

$$WLS(\varpi_i^l) \Rightarrow r_i^l \Rightarrow \varpi_i^{l+1} = \psi(r_i^l)/r_i^l$$

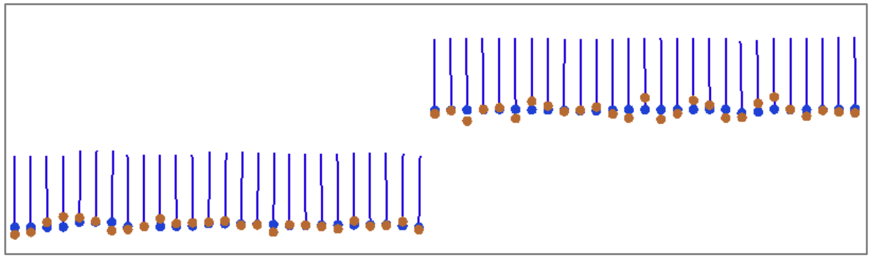
$l \leftarrow l+1$

typically $l = 3..5$ iterations suffice

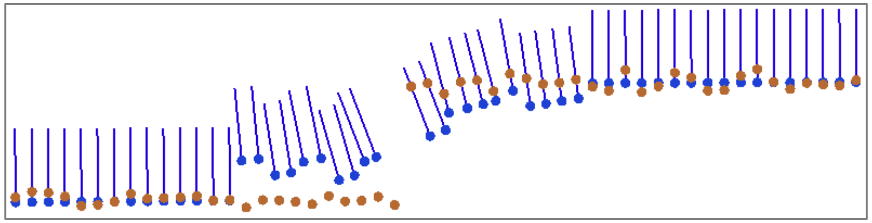
Robust Least Squares – Discussion

- localization of robust fit similar to weighted LS
- most robust norms do not help much with outliers, C^0 and C^1-discontinuities
- Example C^0-discontinuity with best robust norm being German-McClury:

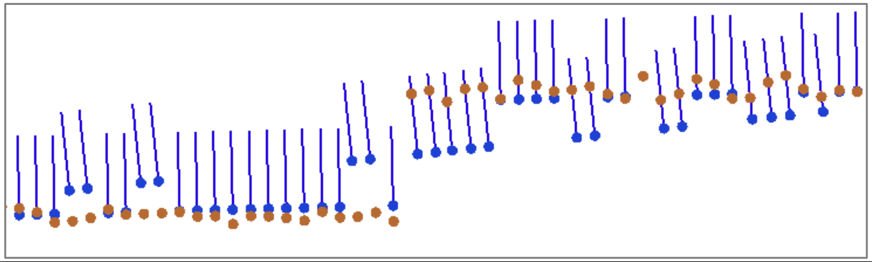
good choices for σ_{loc} and σ_{noise}:



for σ_{loc} too large:



for σ_{noise} too large:



Current Research Questions

Computer Graphics and Visualization

- How to locally estimate σ_{loc} and σ_{noise} from the data?

➔ maximize local goodness of fit over σ_{loc} and σ_{noise}

- How to support C^1-discontinuities?

➔ add weight that depends on difference in estimated normals

- How to get real-time performance?

➔ work on hierarchical redundant grid partition of data

EXAMPLE RESEARCH QUESTIONS

Normal Estimation for 3D Scanning

Efficient Particle Rendering for Visualization

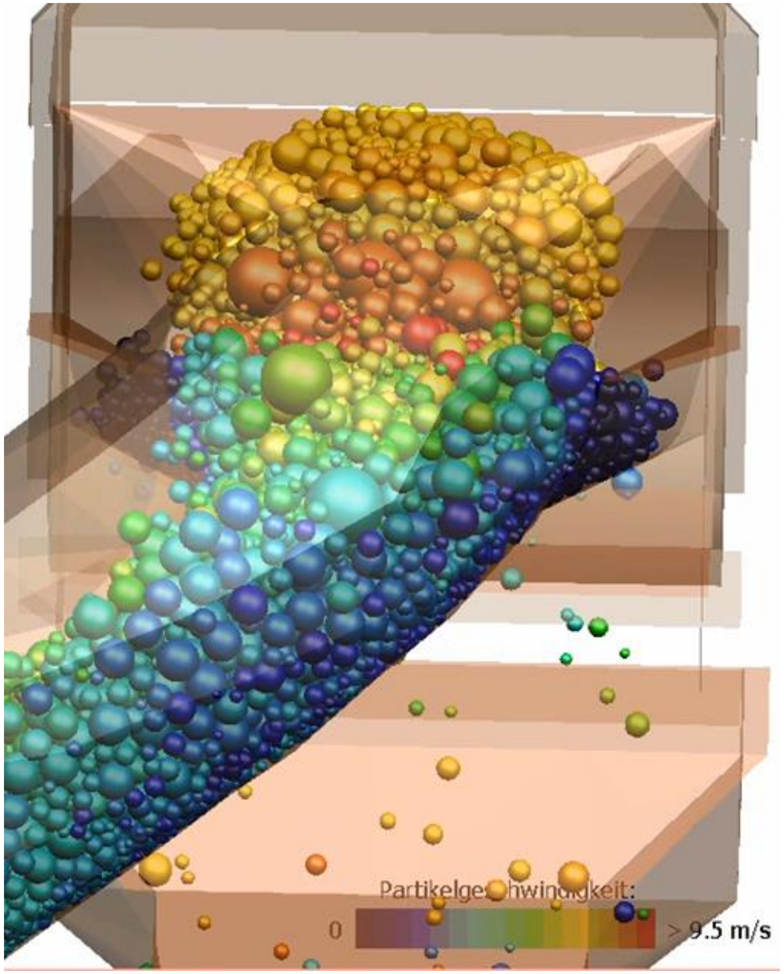
Object Pose Estimation

Particle Based Visualization

Computer Graphics and Visualization

Discrete Element Method

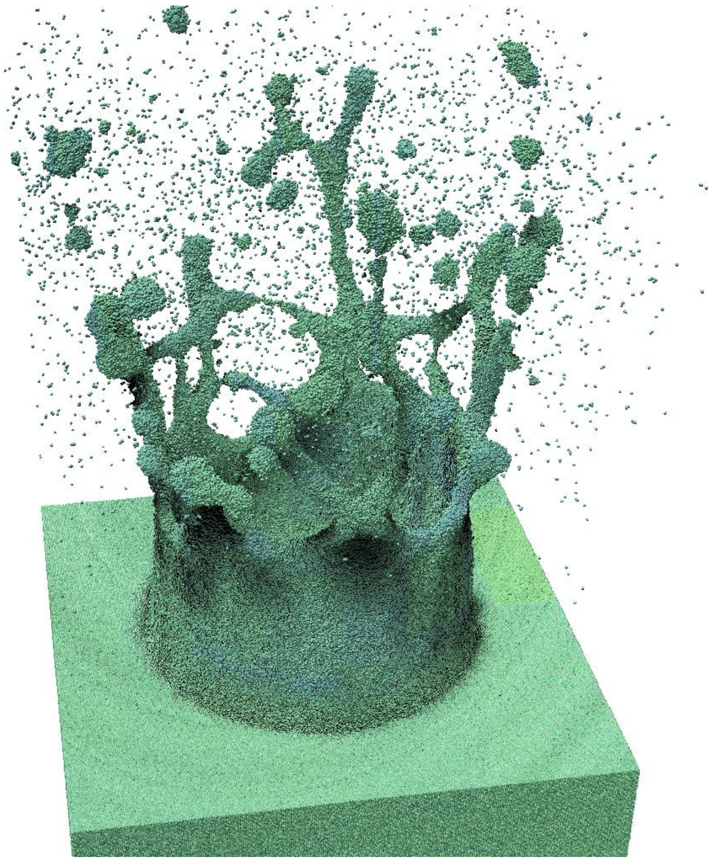
- Applications
  - liquids and solutions, for instance of sugar or proteins;
  - bulk materials in storage silos, like cereal;
  - granular matter, like sand;
  - powders, like toner.
  - Blocky or jointed rock masses
- Simulation
  - Discretize matter on particles
  - Model particle particle interaction forces



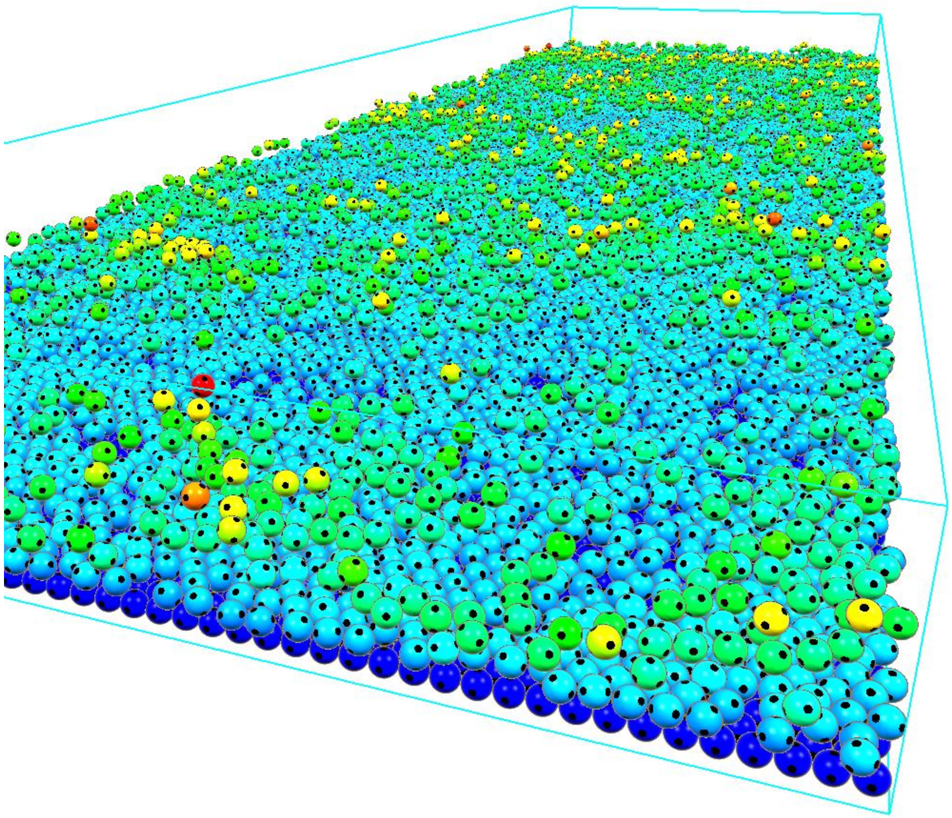
Particle Based Visualization

Computer Graphics and Visualization

- Molecular Dynamics Simulation of Laser Ablation



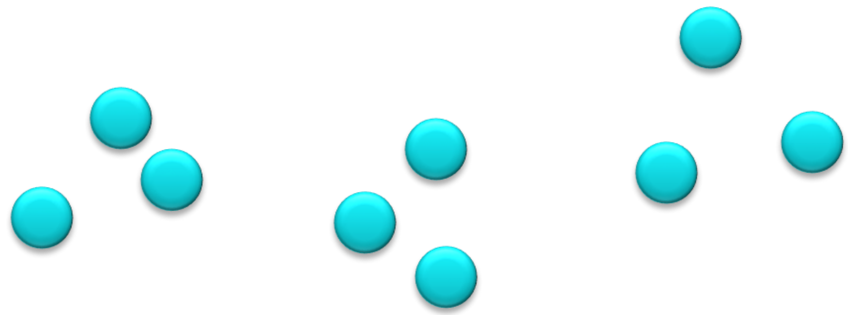
- Simulation of Sedimentation with rotating particles



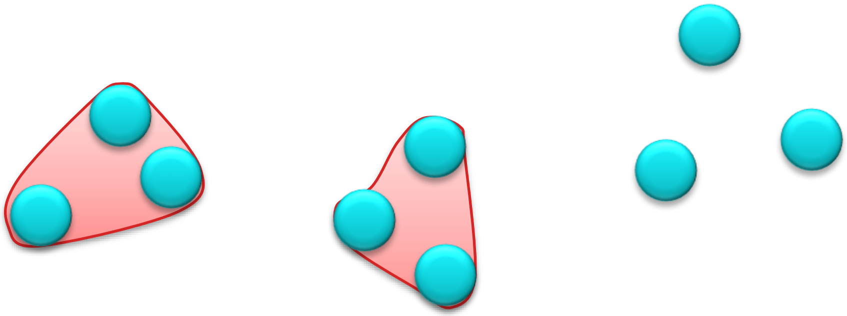
Visual Analysis of Particle Data

Computer Graphics and Visualization

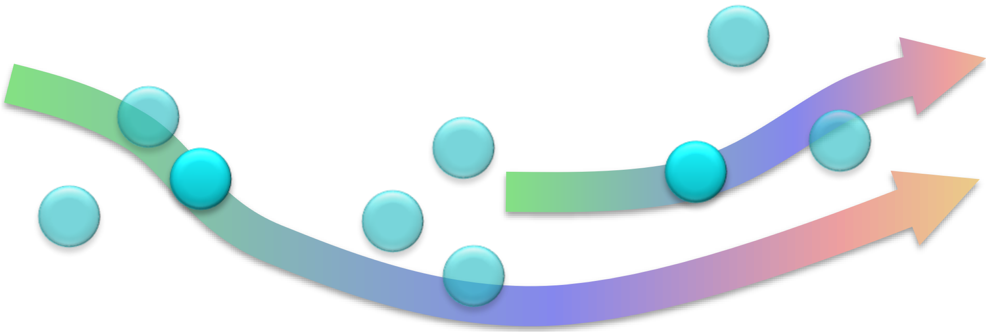
- show animation of particles



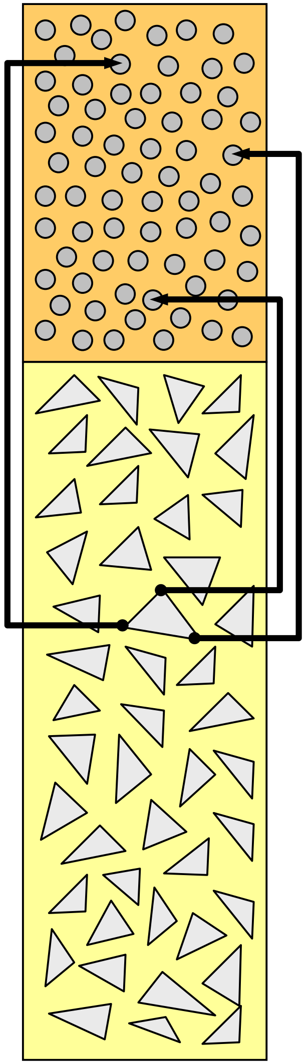
- cluster particle and trace clusters over time

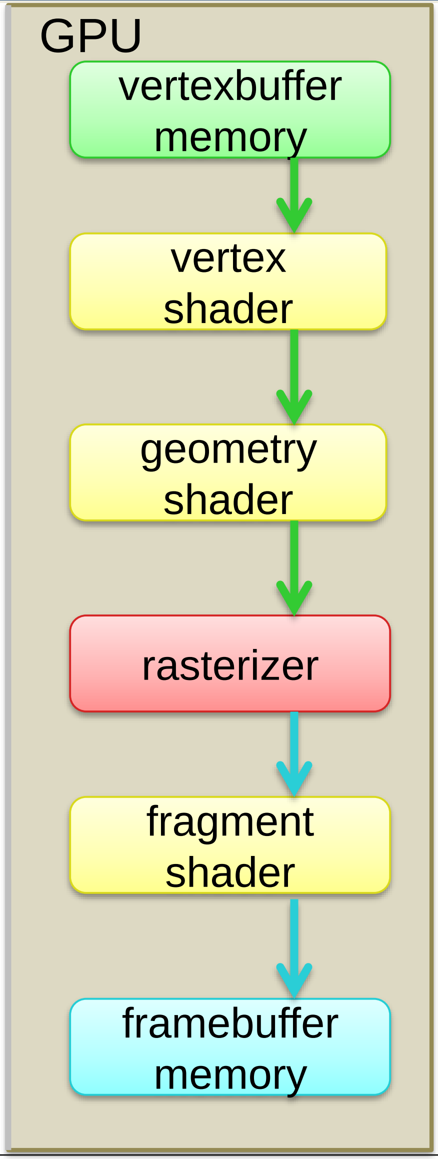


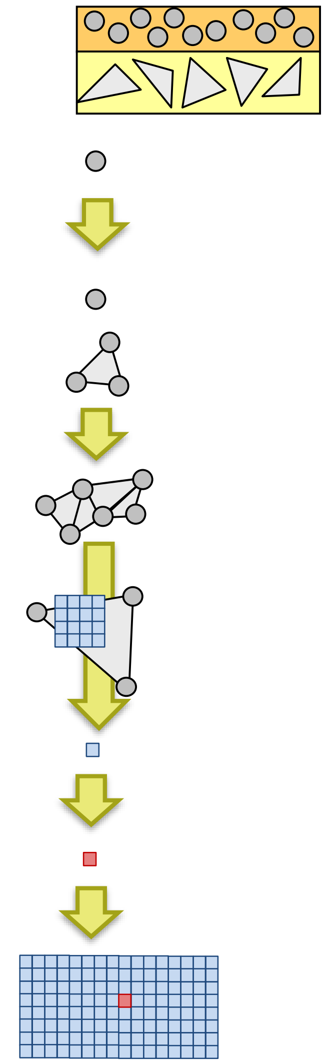
- extract and filter particle traces



Indexed mesh rendering

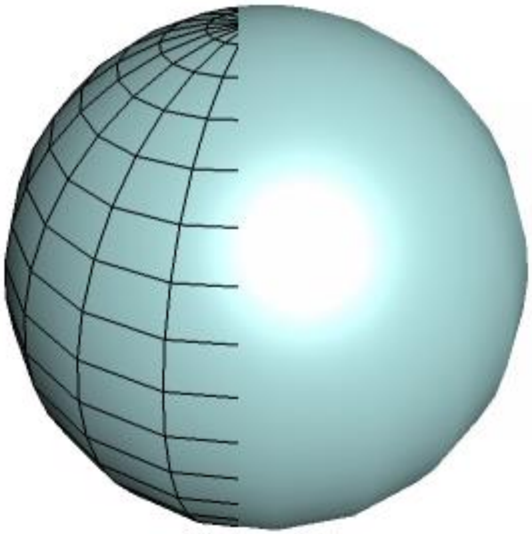




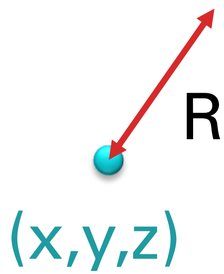


Problem with Pipeline

- points, lines and triangles are only supported graphics primitives
- all other primitives must be tesselated



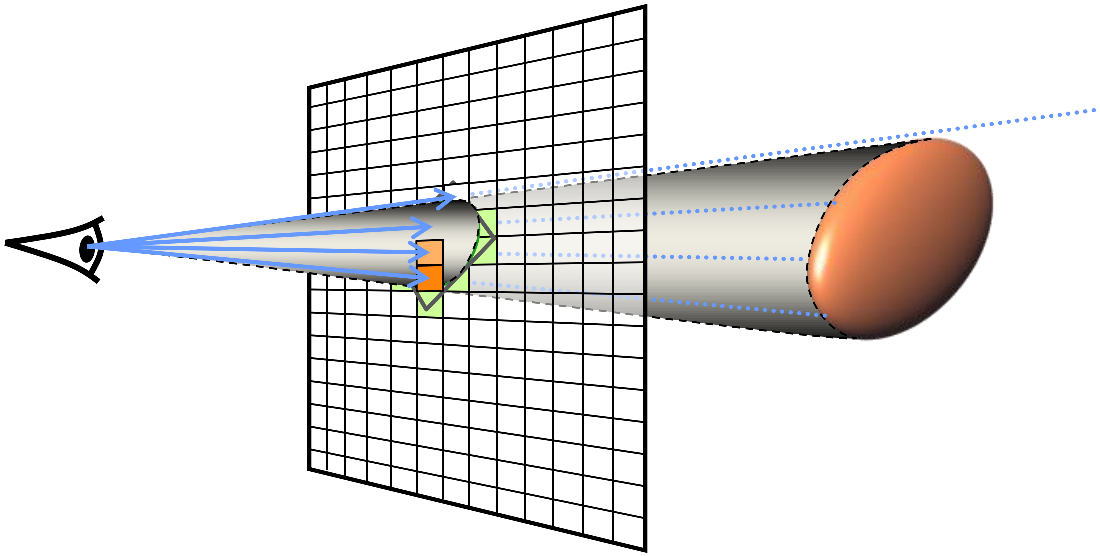
20x20 = 400 vertices plus 800 triangles



vs 4 floats

Particle Raycasting

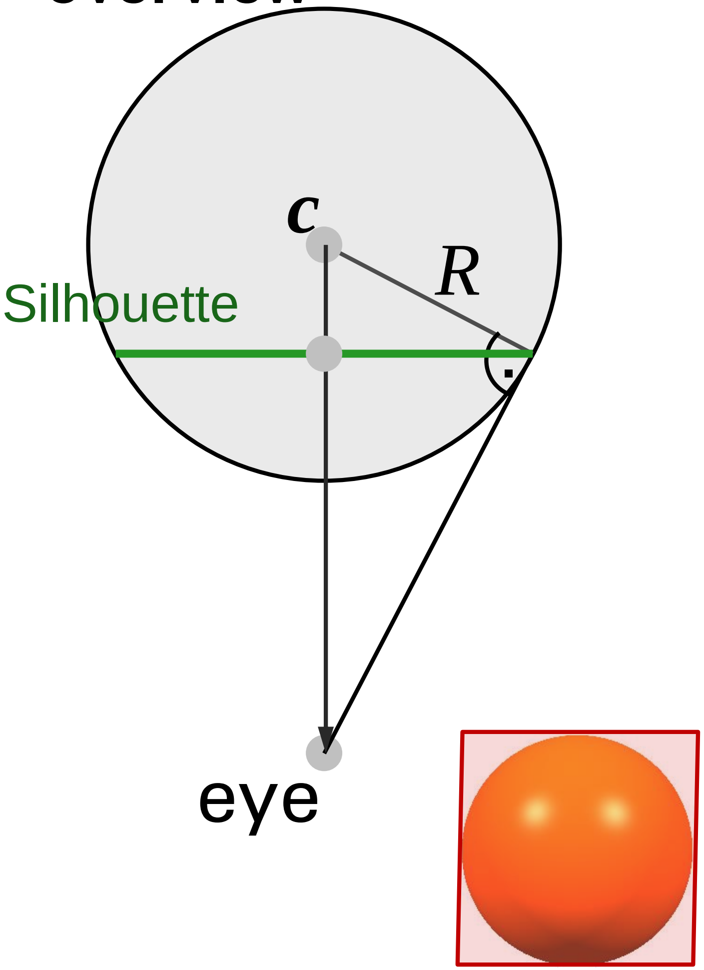
- compute projected silhouette
- cover silhouette with graphics primitive[s]

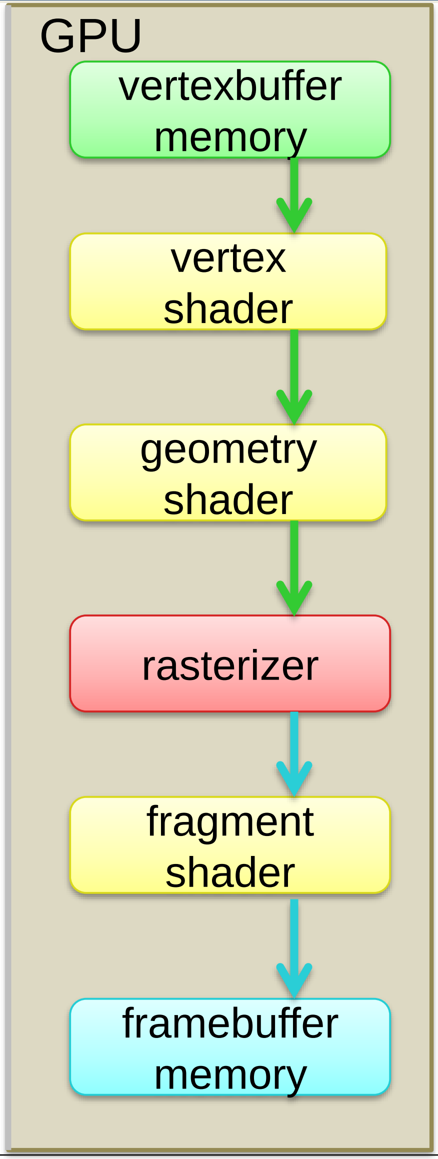


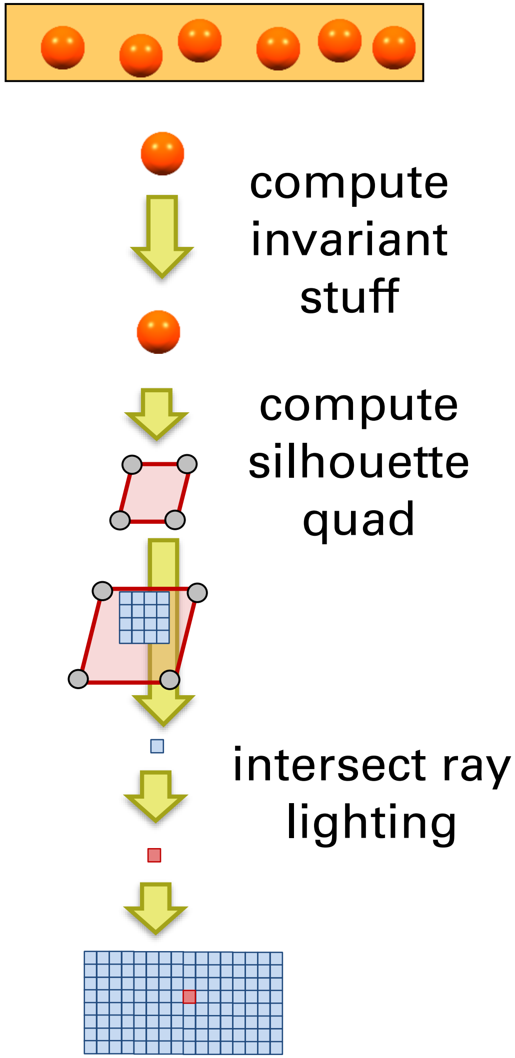
- per fragment ray intersect with particle
- compute illumination

Graphics Pipeline

Particle rendering Overview





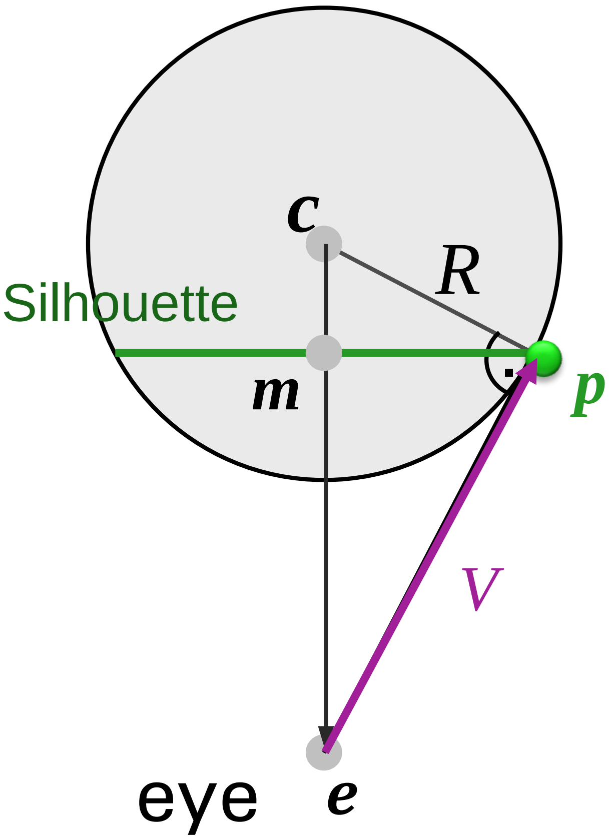


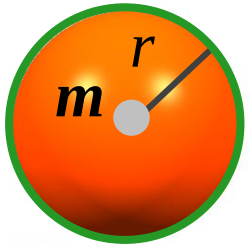
Particle Silhouette

- condition for particle points $\boldsymbol{p}$ on silhouette:

$$\boldsymbol{p} - \boldsymbol{c} \perp V$$

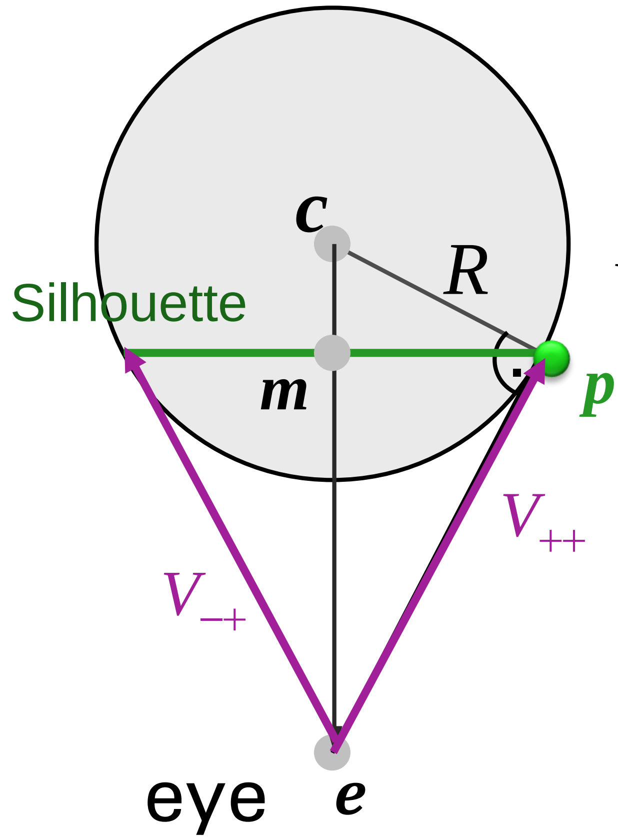
- silhouette points form a circle of some radius $r < R$ around $\boldsymbol{m}$

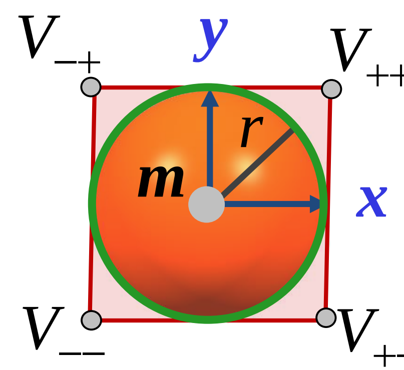




Silhouette Quad (Geometry Shader)

- the particle can be completely covered, by covering the silhouette circle with a quad as graphics primitive
- for this $\boldsymbol{m}$ and the directions $\boldsymbol{x}$ and $\boldsymbol{y}$ are computed
- The 4 quad corners are
  $$\boldsymbol{V}_{\pm\pm} = \boldsymbol{m} \pm \boldsymbol{x} \pm \boldsymbol{y}$$
- $\boldsymbol{x}$ and $\boldsymbol{y}$ parametrize all quad location $\boldsymbol{q}$:
  $$\boldsymbol{q} \epsilon [-1,1]^2$$





simple test for intersection:
$$\|\boldsymbol{q}\|^2 \leq 1$$

Ray Intersection (Fragment Shader)

- rasterizer interpolates $\boldsymbol{V}_{\pm\pm}$ over quads to $\boldsymbol{V}$
- equations for ray-sphere intersection:

$$\boldsymbol{p} = \boldsymbol{e} + \lambda \boldsymbol{V}$$

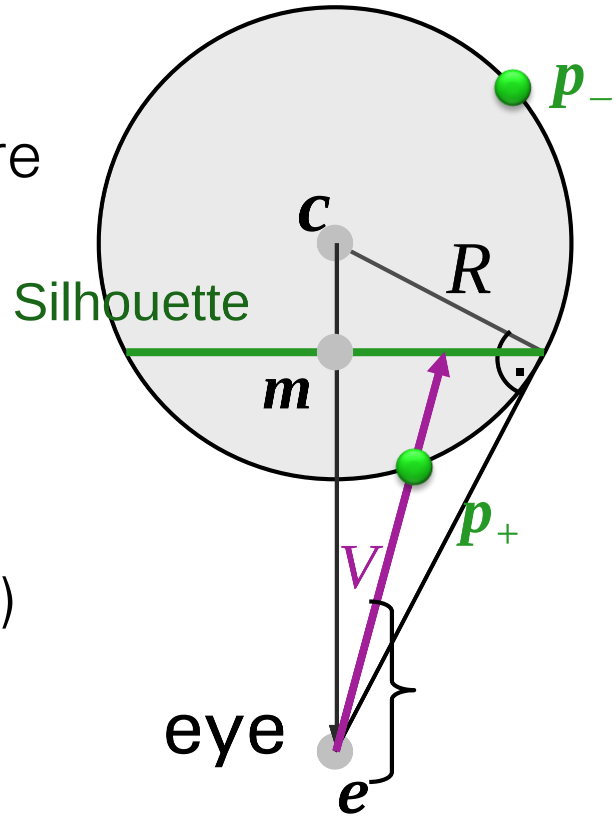
$$\|\boldsymbol{p} - \boldsymbol{c}\|^2 = R^2$$

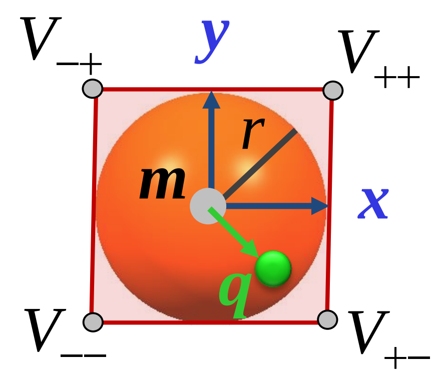
- solution (as homework)

$$\boldsymbol{p}_{\pm} = \boldsymbol{e} + \lambda_{\pm} \boldsymbol{V}$$

$$\lambda_{\pm} = \frac{1}{1 \pm \beta}$$

$$\beta = \frac{R}{\|\boldsymbol{e}-\boldsymbol{c}\|}\sqrt{1 - \|\boldsymbol{q}\|^2}$$





$\boldsymbol{q} \epsilon [-1,1]^2$

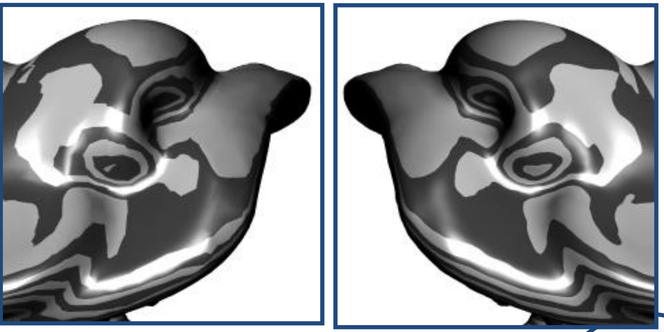
important consequence:
λ_+ is always first intersection

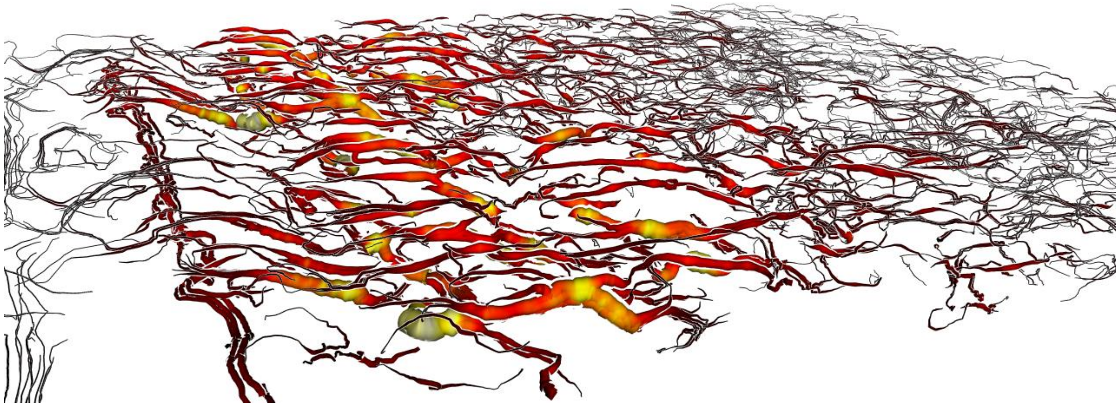
Discussion

- Some more details needed to allow implementation of shaders (see CG1)
- Ellipsoids can be supported very easily by considering it as an affinely transformed sphere

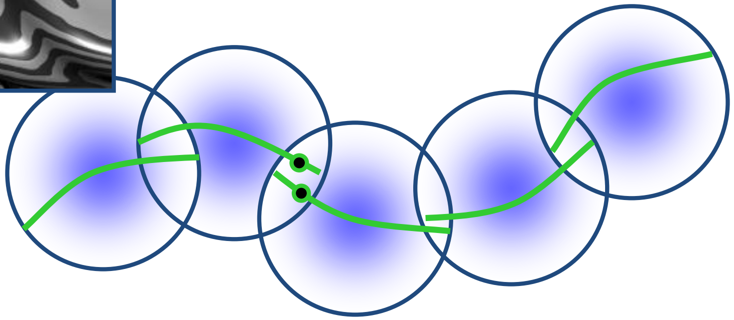
further primitives

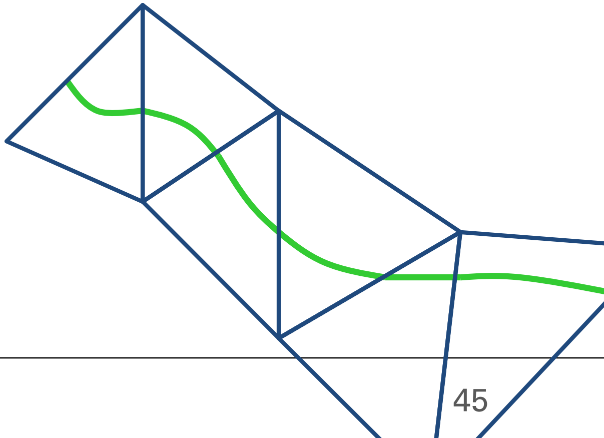
- tubes





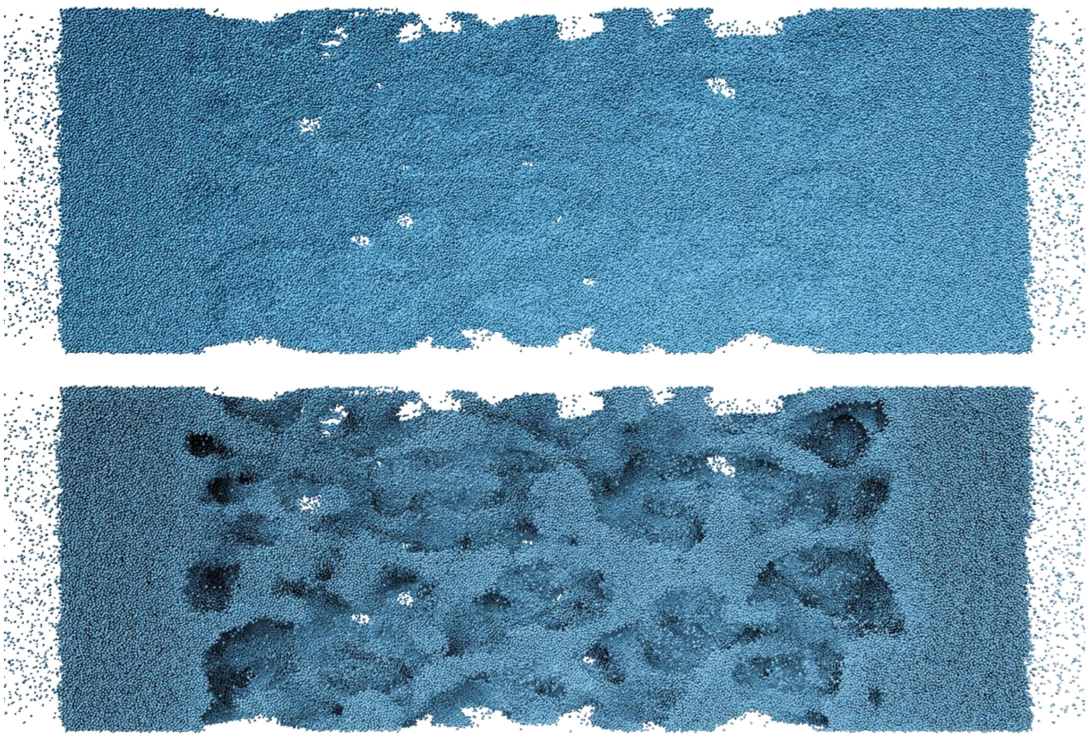
- locally smooth surfaces



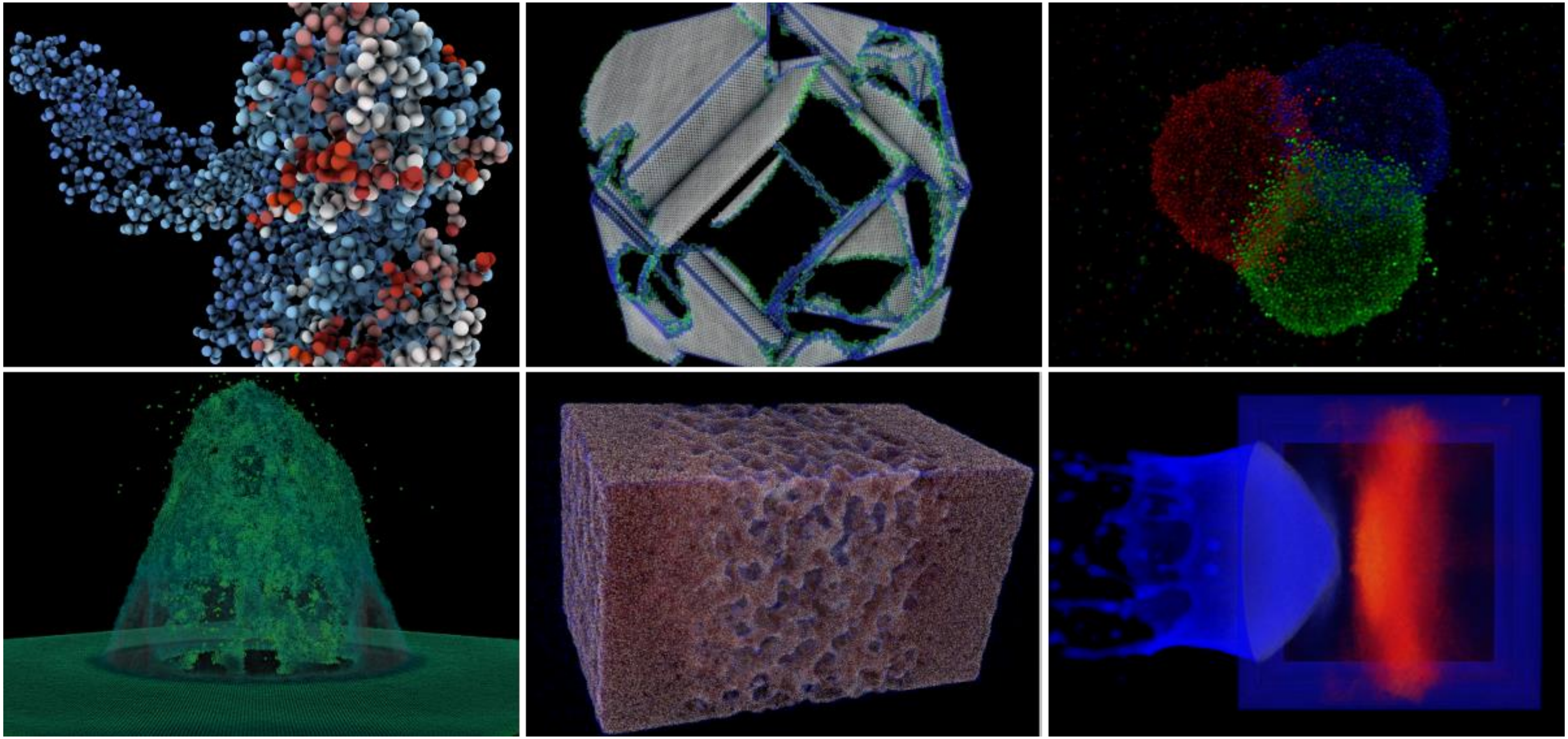


Ambient Occlusion

- Ambient Occclusion approximates incoming light per point through the fraction of hemisphere that sees environment
- Fast Approximation by resampling particle data on volume



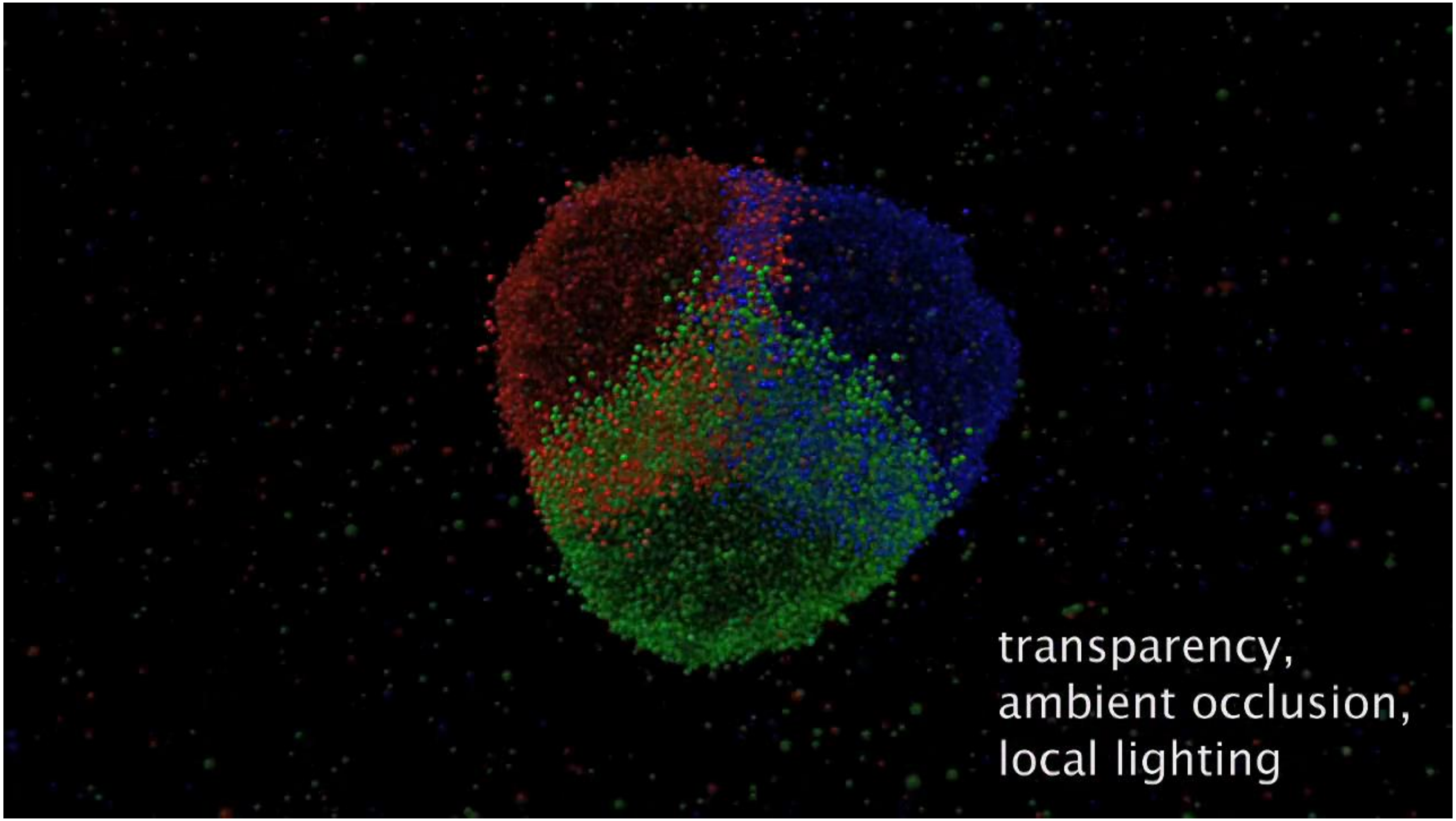
Recent Research



J. Staib, Sebastian Grottel, S. Gumhold, *Visualization of Particle-based Data with Transparency and Ambient Occlusion,* EuroVis 2015

Recent Research

Computer Graphics and Visualization



YouTube DE

J. Staib, Sebastian Grottel, S. Gumhold, *Visualization of Particle-based Data with Transparency and Ambient Occlusion,* EuroVis 2015

EXAMPLE RESEARCH QUESTIONS

Normal Estimation for 3D Scanning

Efficient Particle Rendering for Visualization

Object Pose Estimation

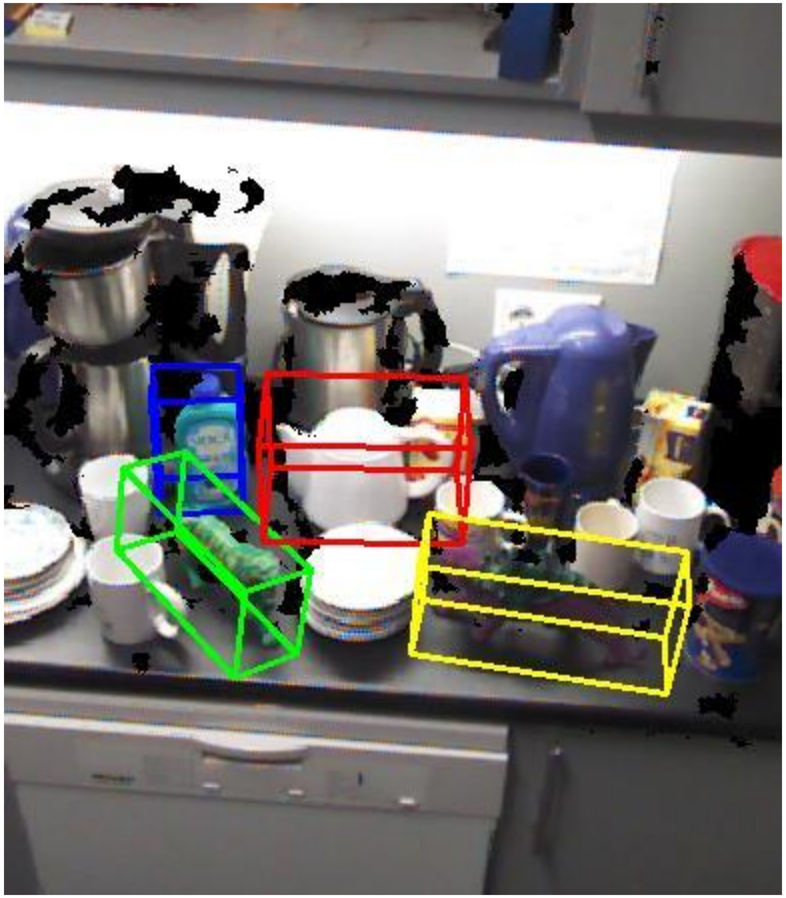
Object Recognition & Pose Estimation
Problem Statement

Given a single RGBD frame

- recognize objects present
- segment objects
- estimate full 6 DoF poses

Challenges

- small objects with occlusion
- cluttered, unknown background
- changing lighting conditions



Object Recognition & Pose Estimation Related Work – Template Based

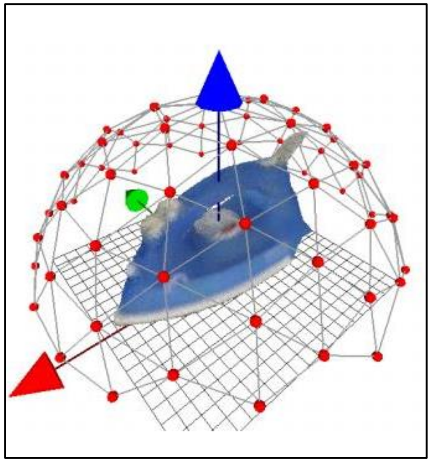
Computer Graphics and Visualization

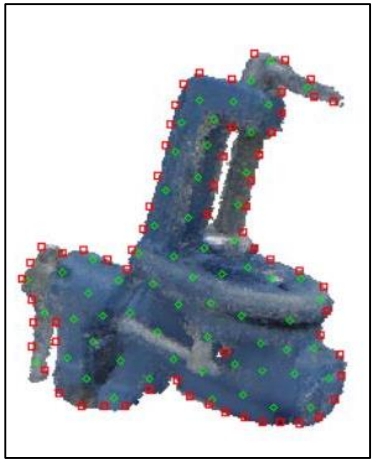
Learning

- acquire 3D-scan of to be recognized model
- synthesize RGB-D views from a dense sampling of hemisphere
- extract and store gradient & color features in templates

Recognition

- sliding window based template matching
- ICP with 3D model

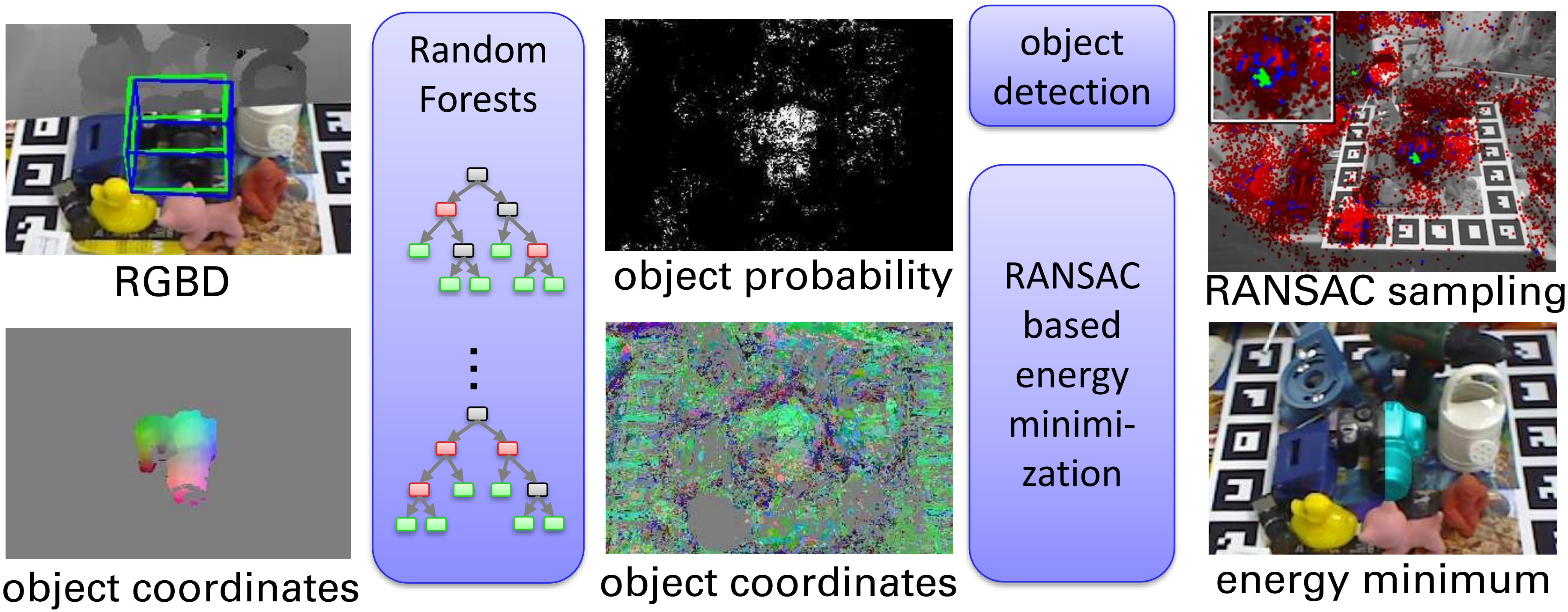




Hinterstoisser 2011, 2012; Rios-Cabrera 2013

Object Recognition & Pose Estimation
Overview of Out Approach

- define object coordinate system
- train random forest to predict object coordinates
- use forest output to detect objects and to initialize energy minimization



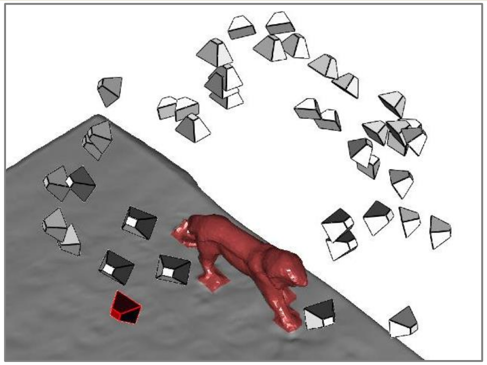
Object Recognition & Pose Estimation
Data Acquisition – 2 Approaches

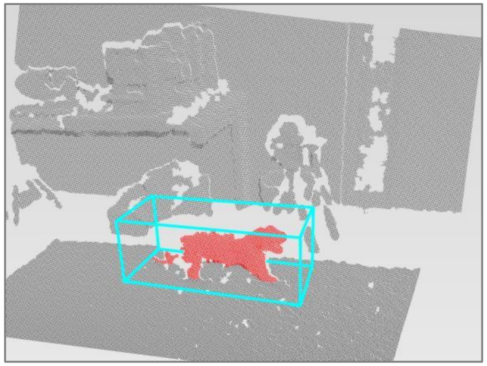
3D Scan Approach

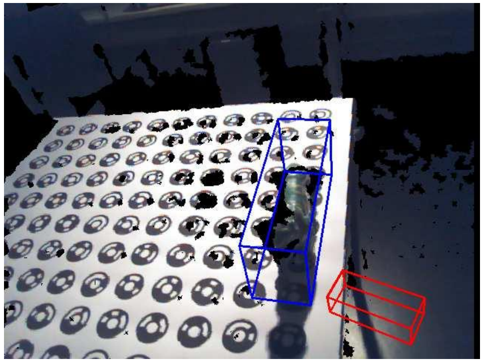
- perform 3D Scan
- define object coordinates
- regularly sample views
- render ground truth with GPU

Kinect Fusion Approach

- scan with Kinect Fusion
- ensure good sampling
- define object coordinates from fusion result
- repeat acquisition for several lighting conditions







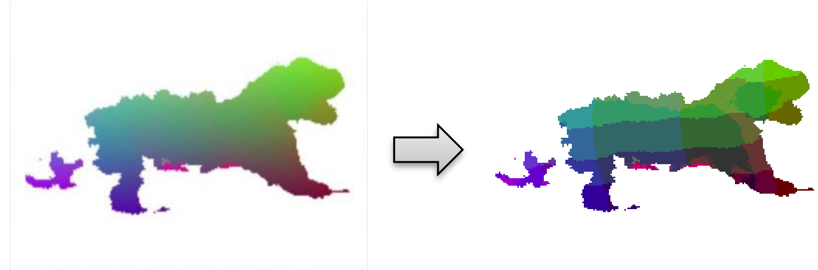
Object Recognition & Pose Estimation
Dataset of 20 Models

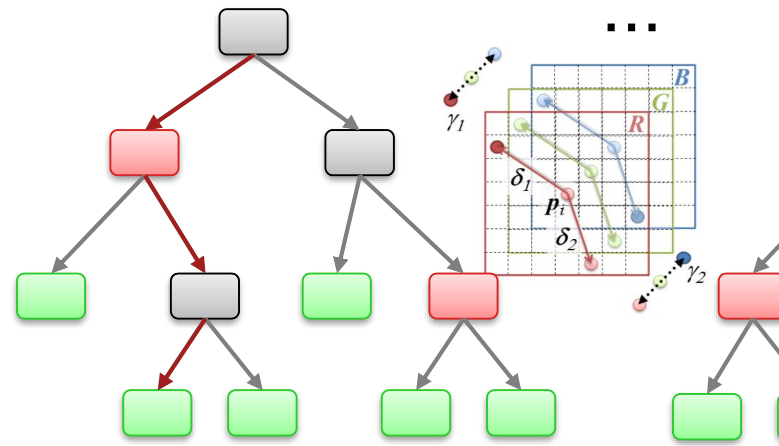
Object Recognition & Pose Estimation
Random Forest

Computer Graphics and Visualization

Building the Forest

- quantize object coordinates
- use simple depth and color features
- use information gain to select features
- in each leaf store probability of objects and per object one object coordinate





p_c^l, y_c^l

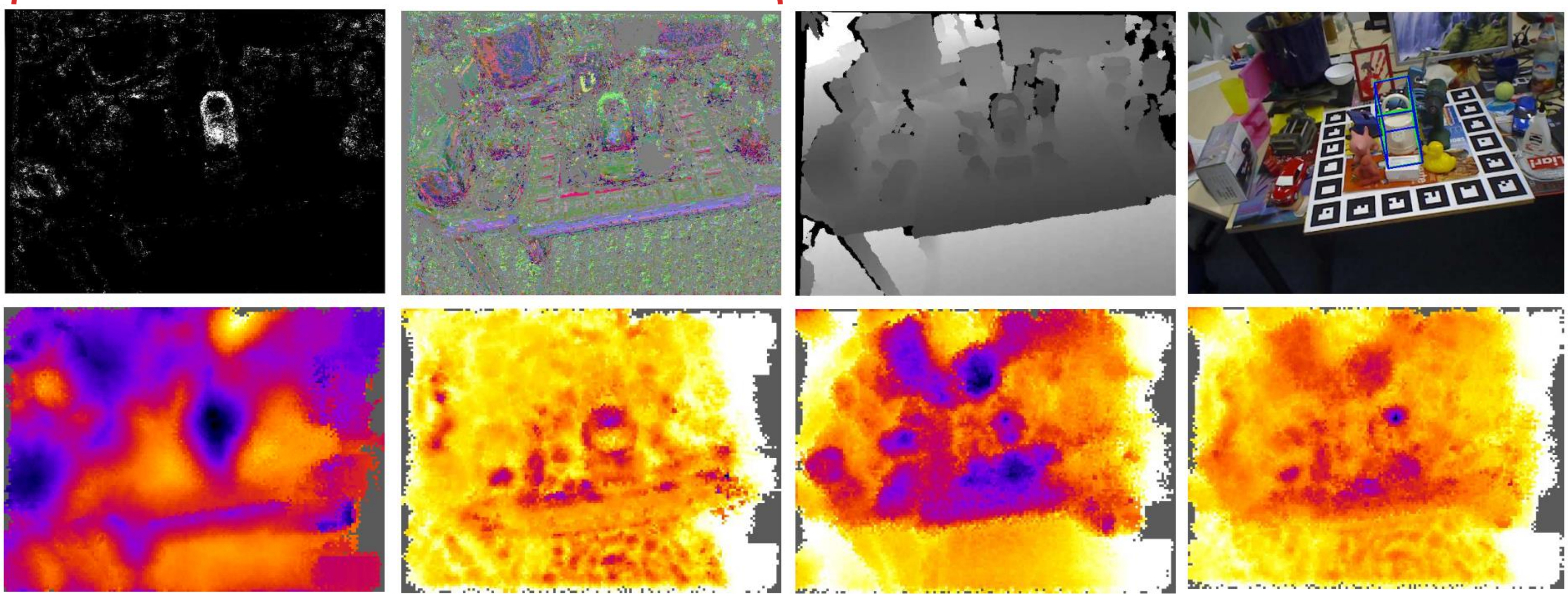
Usage of Forest

- traverse per pixel all trees and average object probabilities and coordinates

$p_{c,i}^l, y_{c,i}^l$

Object Recognition & Pose Estimation
Energy Function

From Random Forest **From Rendering**

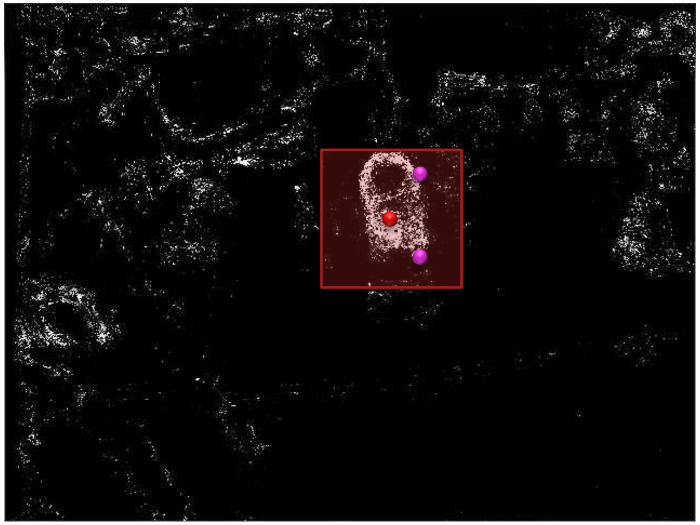


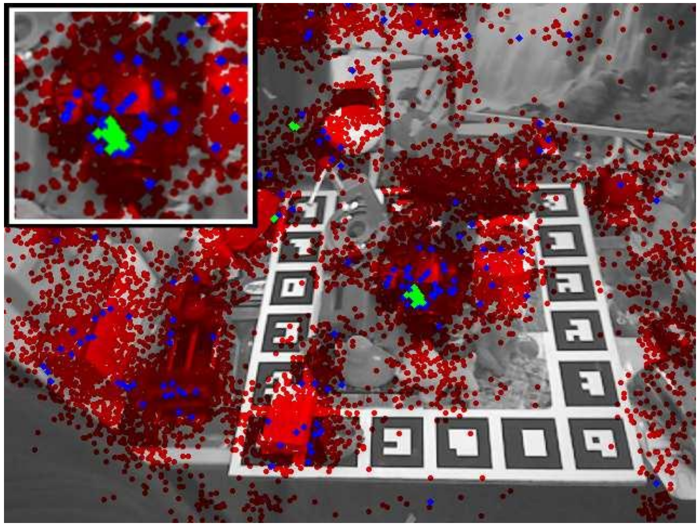
object probabilities — object coordinates — depth comparison — final energy

$p_{c,i}^{l}$ $y_{c,i}^{l}$

input: per pixel object probability and coordinate

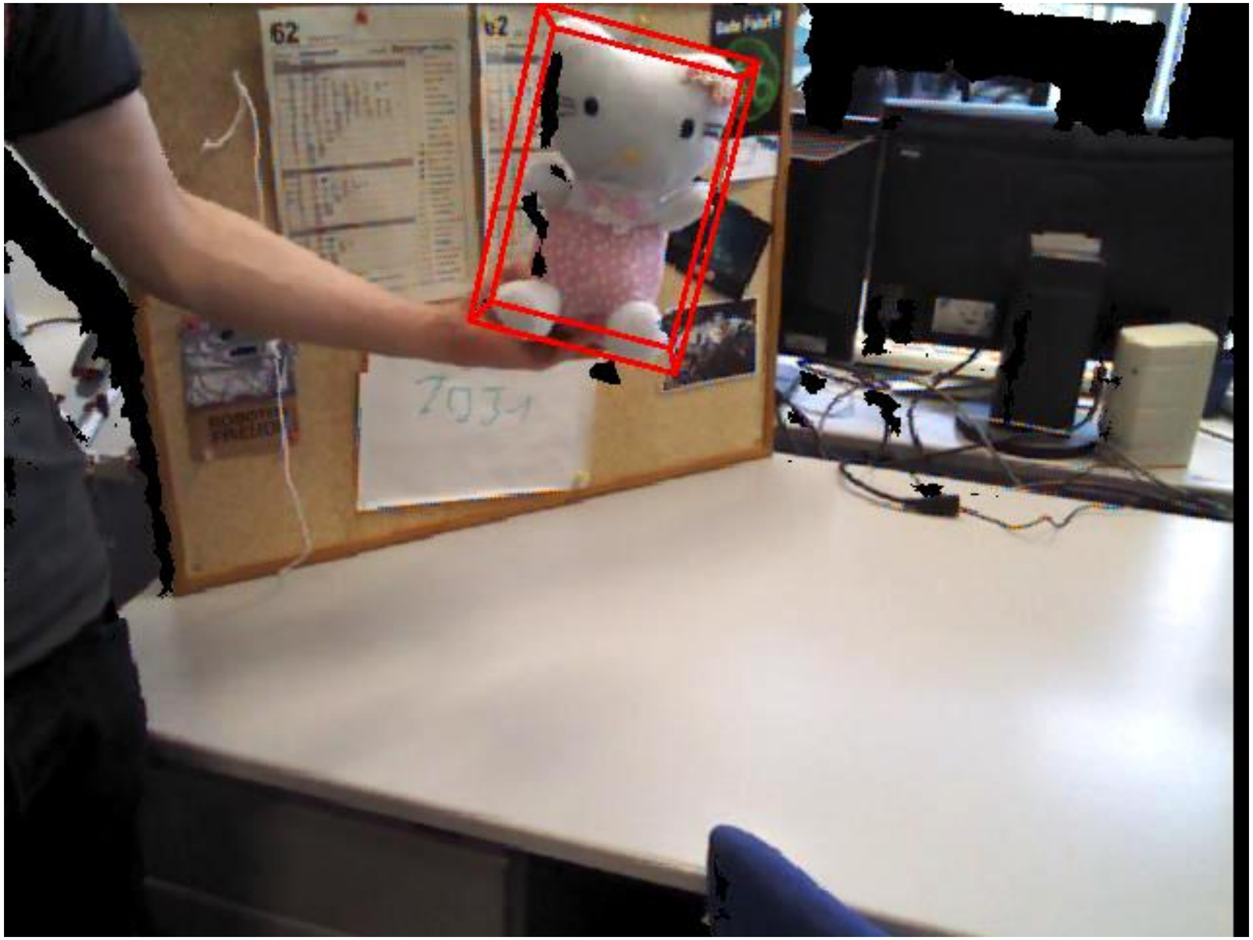
- draw pixel sample proportional to object probability (red)
- determine size of object in image from pixel depth
- sample two more pixels
- construct hypothesis and discard inconsistent ones
- evaluate energy & discard all but 25 best ones
- do final refinement





Object Recognition & Pose Estimation
Results

Computer Graphics and Visualization



Oral Exam

- Based on these Slides with details only for
  - Normal Estimation for 3D Scanning
  - ~~Efficient Particle Rendering for Visualization~~
- The following publications should be read
  - Consistent Propagation of Normal Orientations in Point Clouds, König, S.; and Gumhold, S. In *Proceedings of Workshop on Vision, Modeling and Visualization*, pages 83--92, 2009.
  - Robust Surface Reconstruction from Point Clouds, König, S.; and Gumhold, S. Technical Report TUD-FI13-03, TU Dresden, 2013.
  - ~~Splatting Illuminated Ellipsoids with Depth Correction, Gumhold, S. In *Proceedings of International Workshop on Vision, Modeling, and Visualization*, pages 245--252, November 2003.~~
  - ~~Incremental Raycasting of Piecewise Quadratic Surfaces on the GPU, Stoll, C.; Gumhold, S.; and Seidel, H. In *Proceedings of IEEE Symposium on Interactive Raytracing*, pages 141--150, 2006.~~