Robotic 3D Vision

Lecture 20: Map Representations

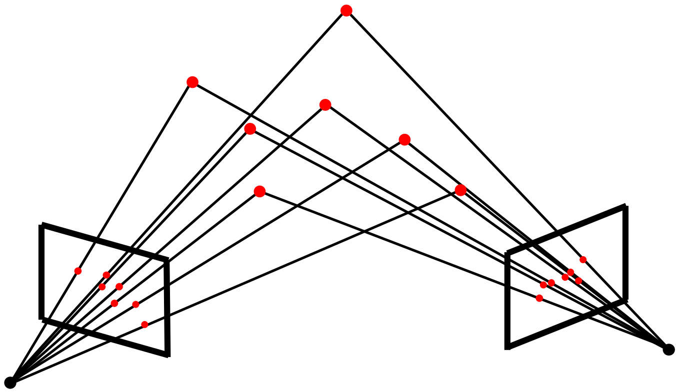
Prof. Dr. Jörg Stückler
Computer Vision Group, TU Munich
http://vision.in.tum.de

What We Will Cover Today

- Dense map representations
- Implicit vs. explicit representations
- Occupancy maps
- Signed distance function maps
- Surfel splat maps

Recap: Dense Depth from Two Views

- So far: triangulation of corresponding interest points between two images to find depth
- How can we obtain depth for all pixels in one of the images?
- Assume relative pose between the camera images known
- Assume intrinsic camera calibration known



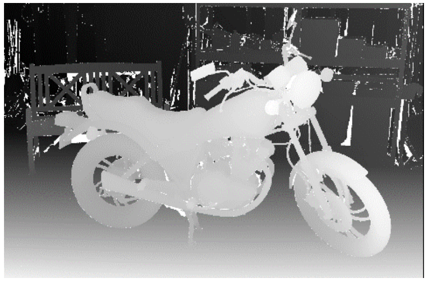


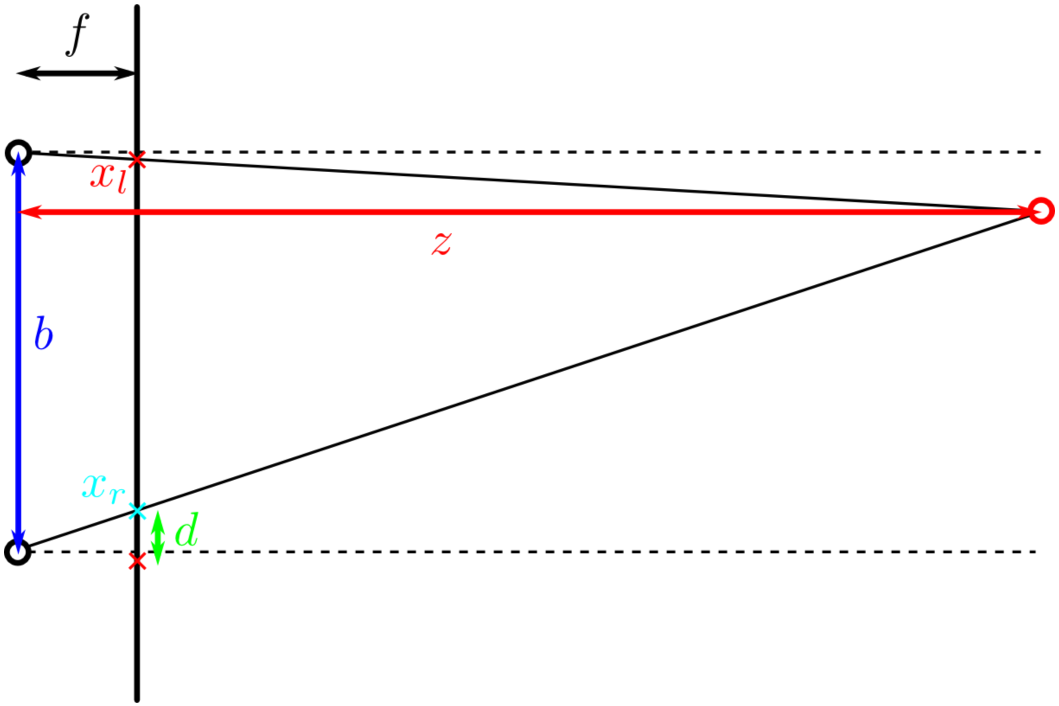
Image source: Scharstein et al., Middlebury stereo benchmark

Recap: Relation of Disparity and Depth

Similar triangles:

$$\frac{b}{z} = \frac{b-d}{z-f}$$

$$\Rightarrow \quad d = \frac{bf}{z}$$



- Disparity is inversely proportional to depth:
  The larger the depth, the smaller the disparity
- Disparity is proportional to the baseline:
  The larger the baseline, the larger the disparity

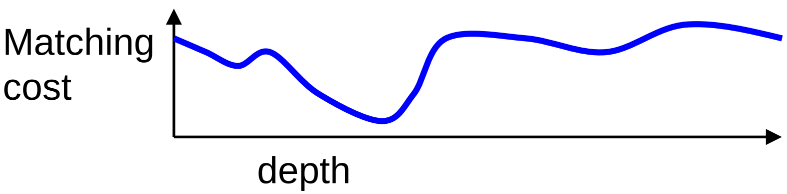
Recap: Dense Stereo Depth Estimation

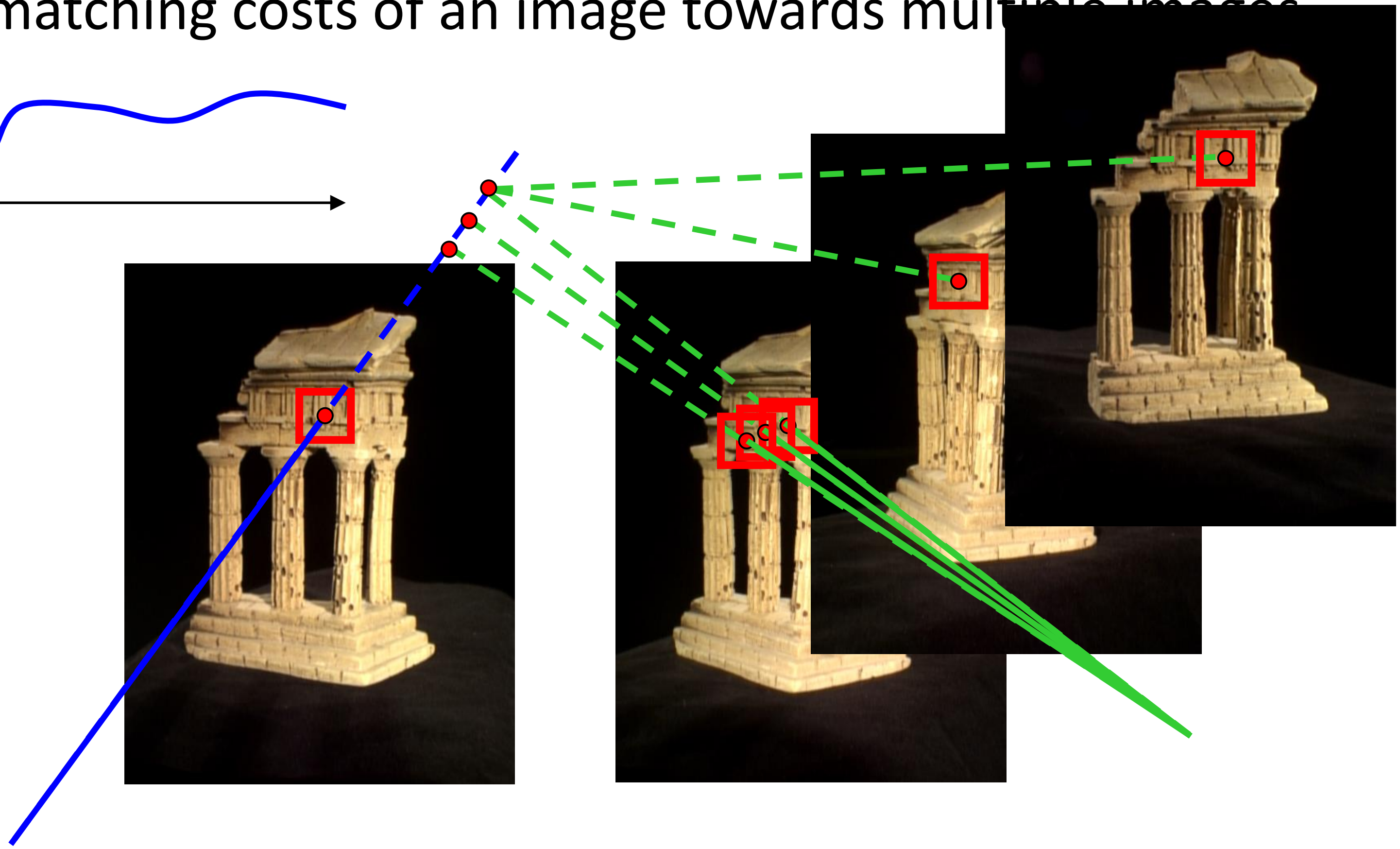
- Better idea: Compare patches (blocks)

- New questions:
  - What are good patch correlation measures?
  - Patch size?
  - etc.

Recap: Dense Depth from Multiple Views

- Straightforward approach: extend two-view matching cost to sum over matching costs of an image towards multiple images

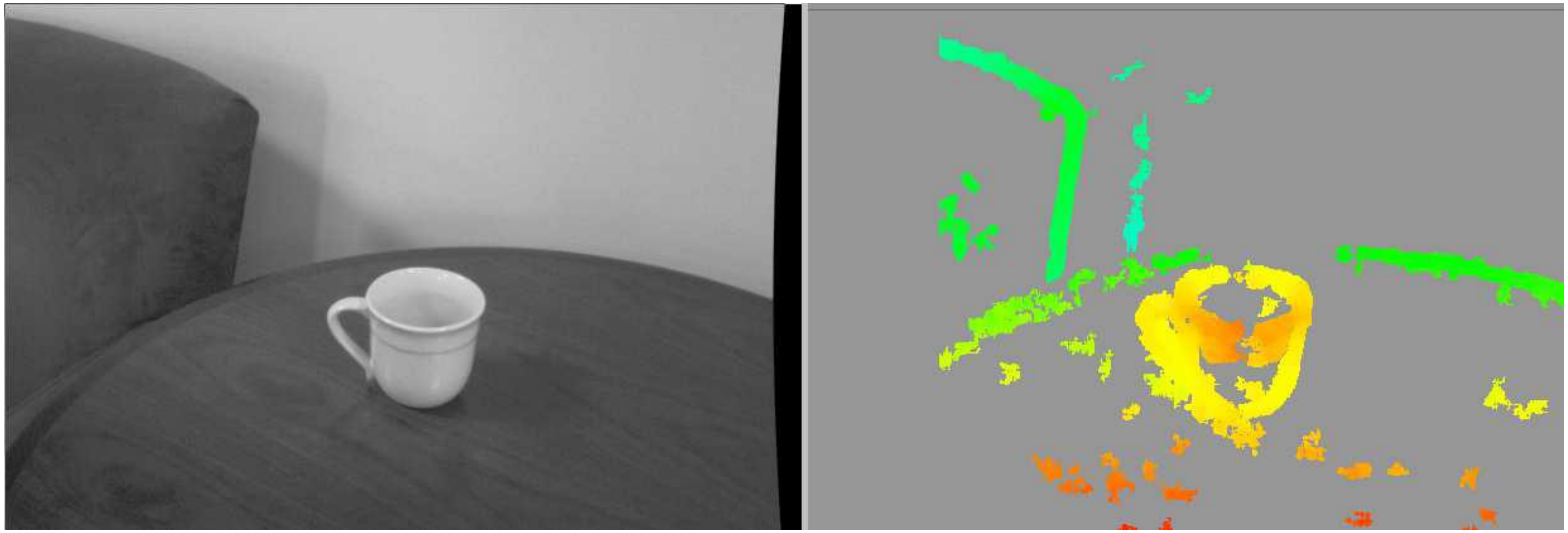




Slide adapted from R. Szeliski

Recap: Active Depth Sensing

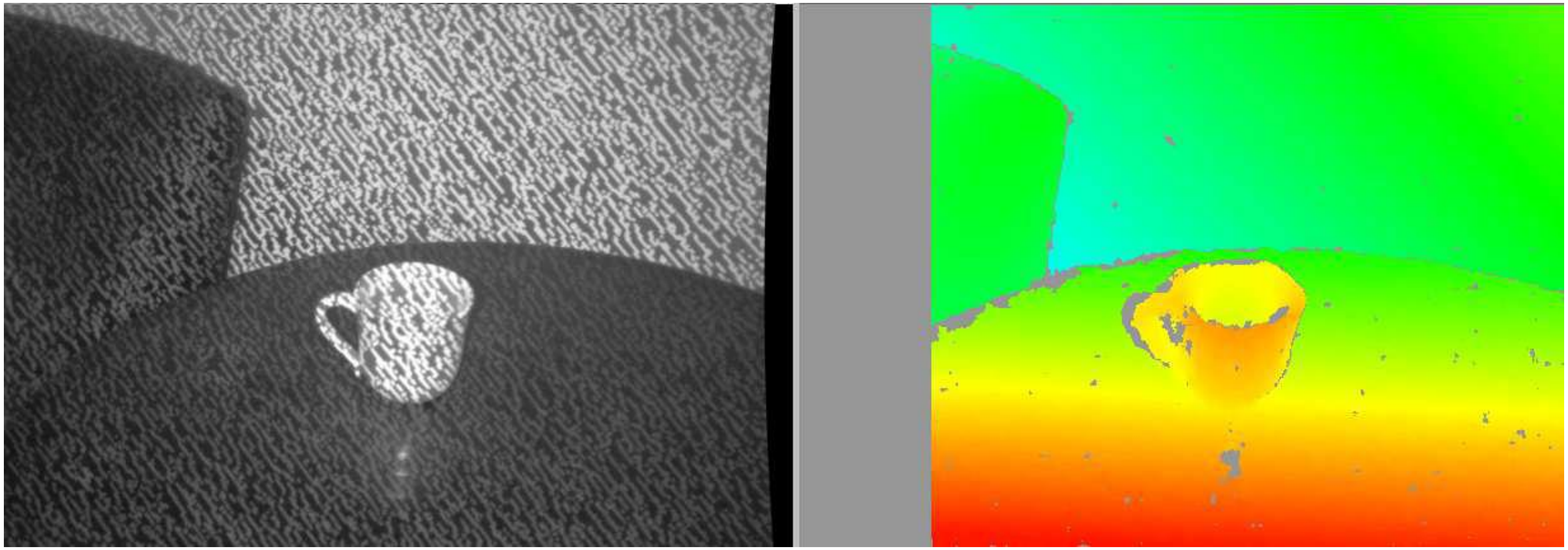
- What can we do about textureless scenes?



Images: J. Sturm

Recap: Active Depth Sensing

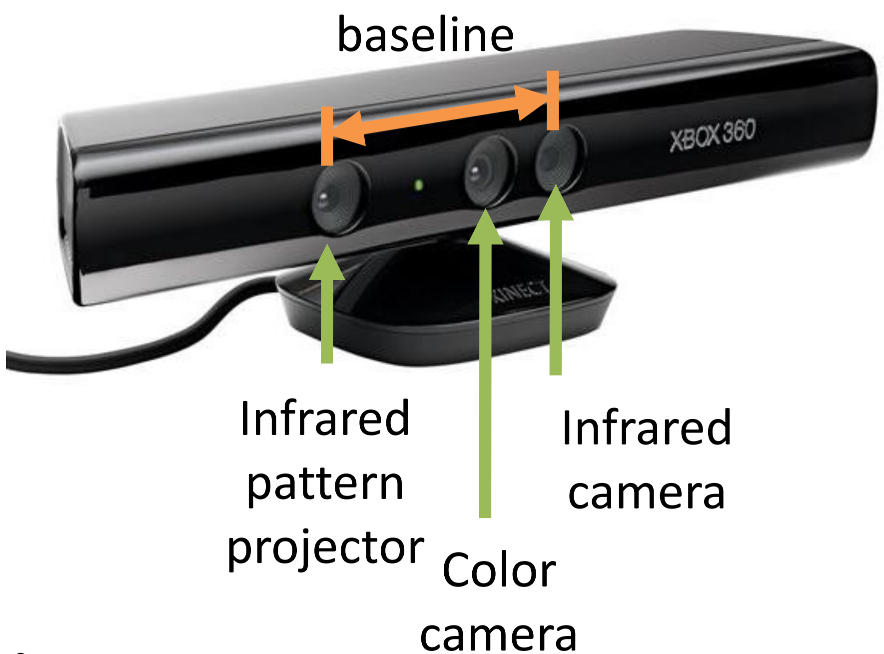
- Idea: Project light/texture



Images: J. Sturm

Recap: Structured Light Measurement Principle

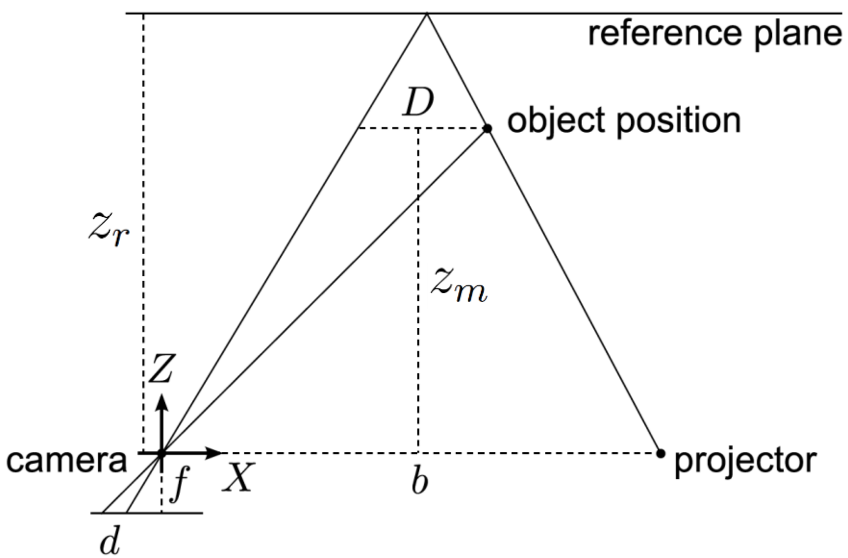
- Project speckle pattern using infrared laser and diffraction element
- Measure infrared speckles using infrared camera
- Measure corresponding RGB image using color camera



Slide adapted from J. Sturm

Recap: Structured Light Measurement Principle

- Use known baseline and reference pattern for depth measurement



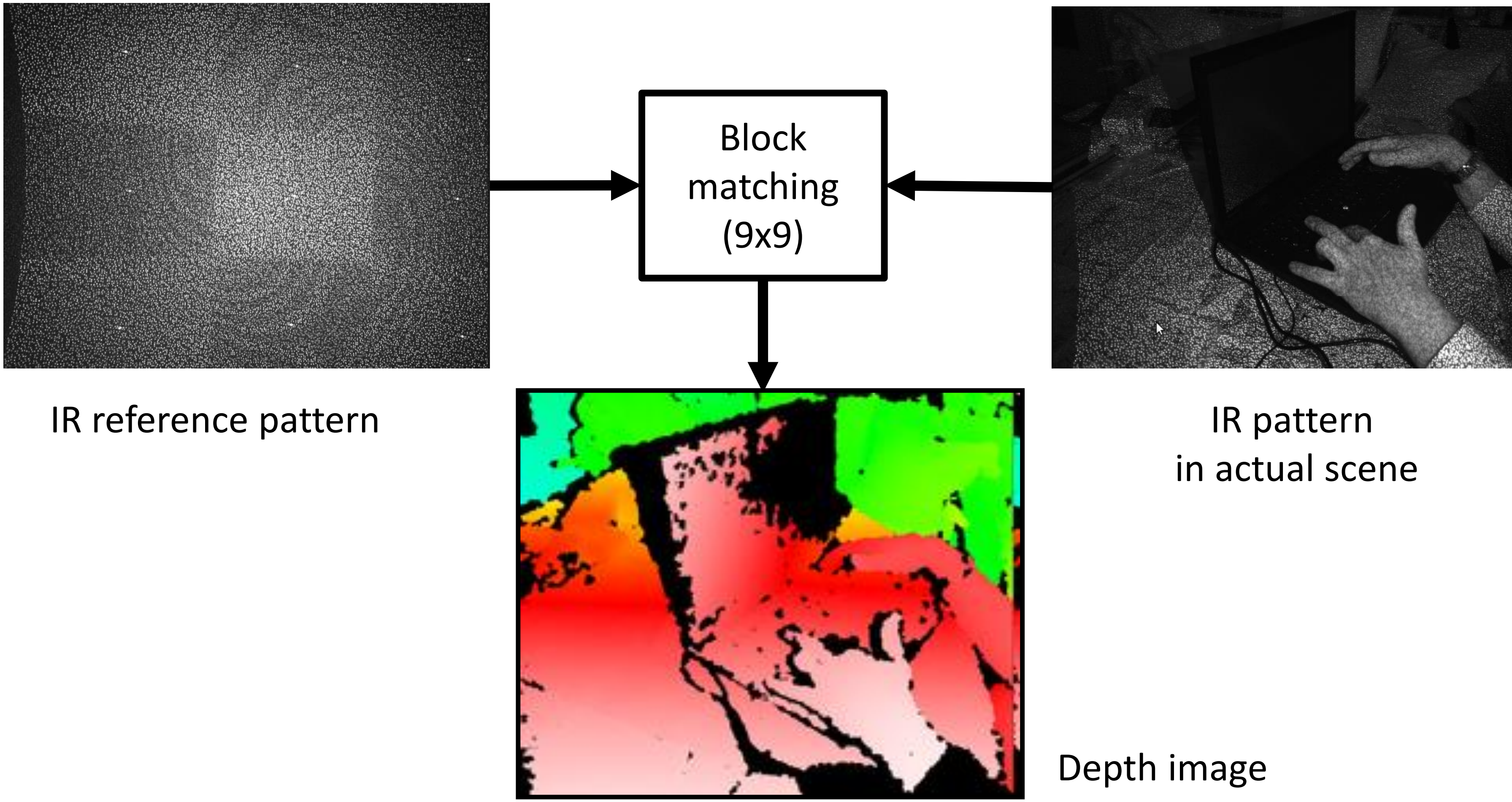
$$\frac{d}{f} = \frac{D}{z_m}$$

$$\frac{D}{z_r - z_m} = \frac{b}{z_r}$$

$$z_m = \frac{z_r}{\frac{d z_r}{b f} + 1}$$

Image: Stückler, 2014

Recap: Structured Light Measurement Principle

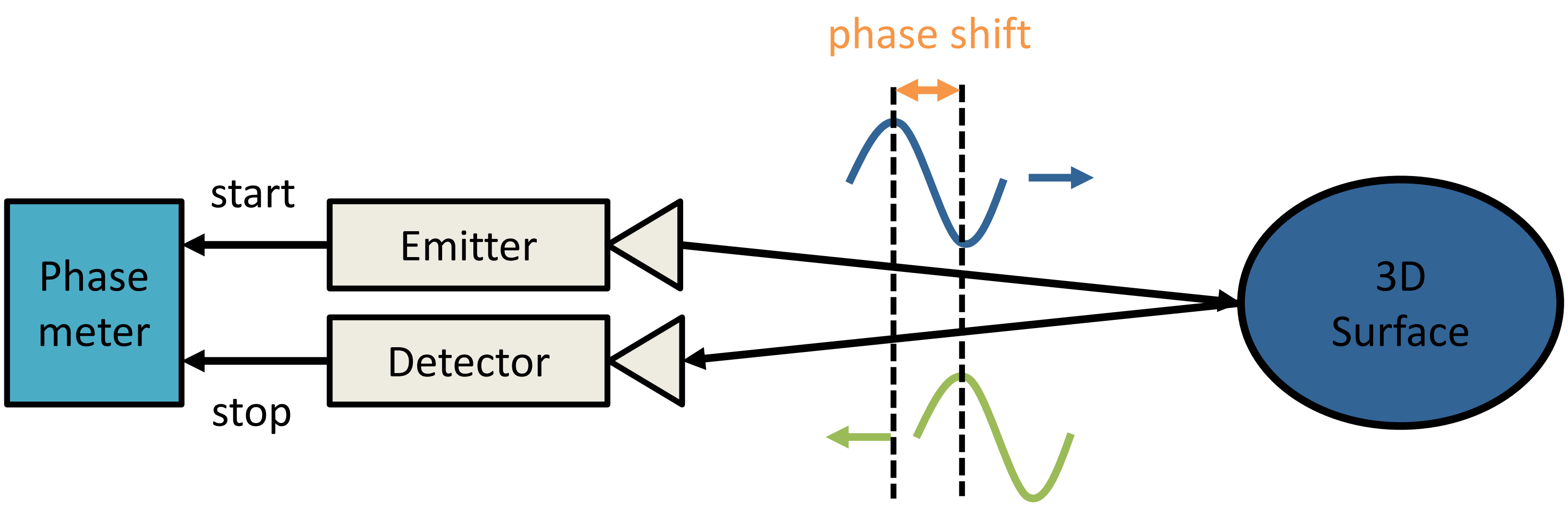


Slide adapted from J. Sturm

Recap: Time-of-Flight Measurement Principle

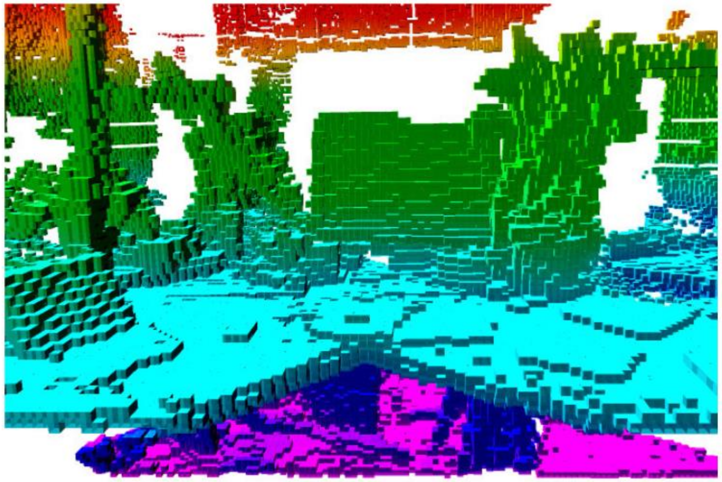
- Idea: emit continous modulated IR wave signal and measure phase shift

- Signal periodicity creates phase ambiguities: use multiple frequencies

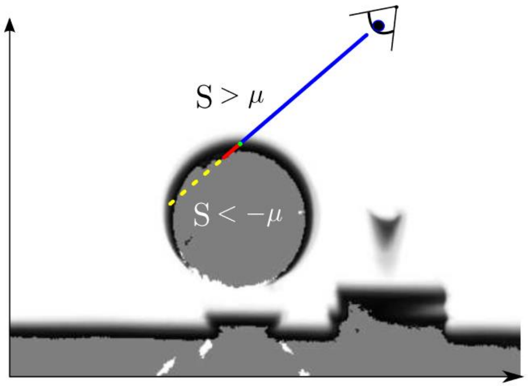


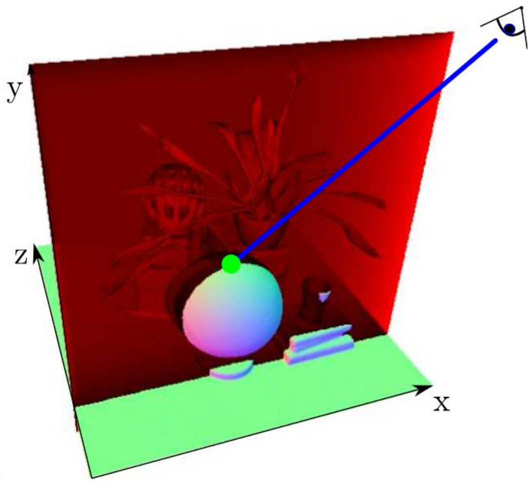
Slide adapted from N. Navab

Dense 3D Map Representations

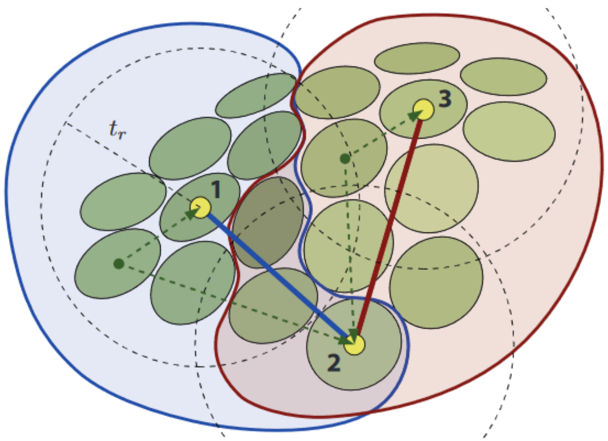


Volumetric Occupancy Maps





Volumetric Signed Distance Functions



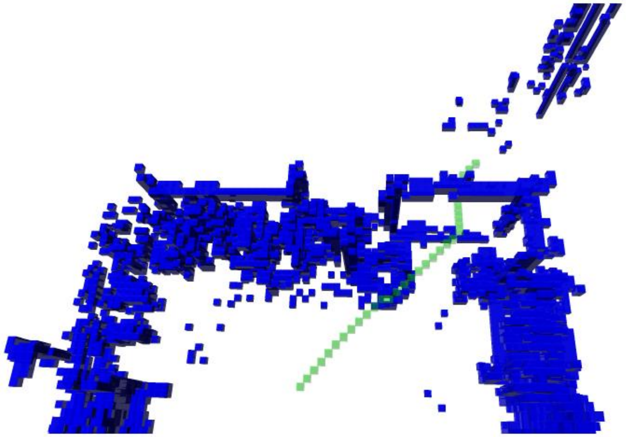
Surfel Splats

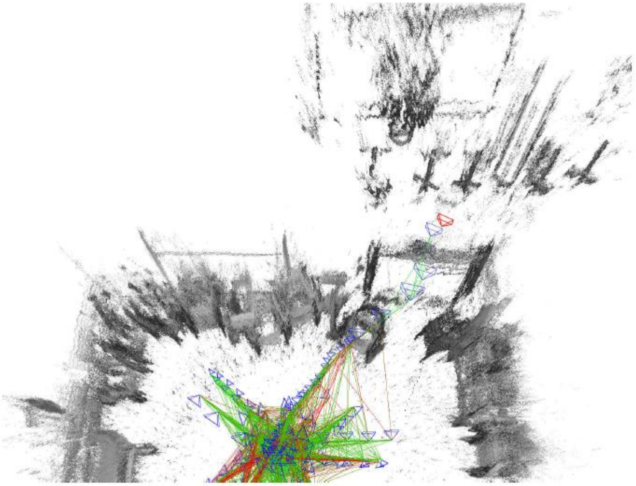
Images: Weise et al., 2011; Wurm et al., 2010; Newcombe et al., 2011

Example Usage of Dense 3D Maps

Augmented and virtual reality

Robot navigation and exploration





Images: von Stumberg et al., 2016; Newcombe et al., 2011

Dense 3D Maps in SLAM

- Tracking and Mapping approaches
  - Drift accumulates in the map

- Fuse map from dense depth images in optimized camera poses
  - Offline integration after sequence recording
  - Online integration requires map modification when poses change

- Full SLAM: dense bundle adjustment
  - Mostly offline approaches
  - ElasticFusion: Joint optimization of camera alignment to surfel map and alignment of corresponding surfels

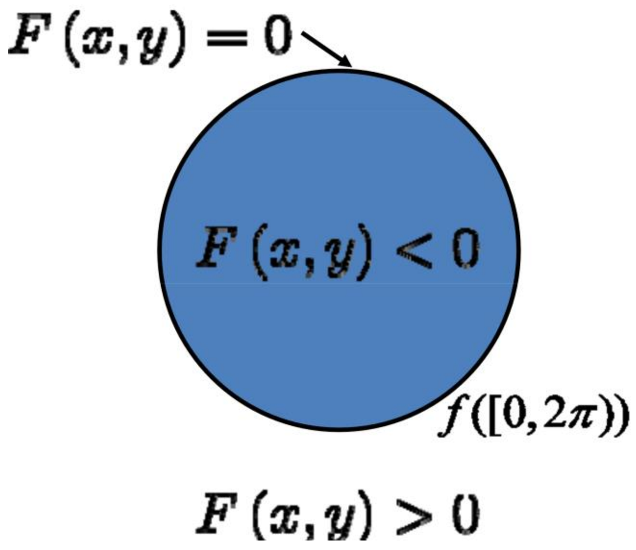
Implicit vs. Explicit Surface Representations

- Explicit:
  - Image of parametrization

$$f(t) = (x(t), y(t)) = (r\cos(t), r\sin(t))$$

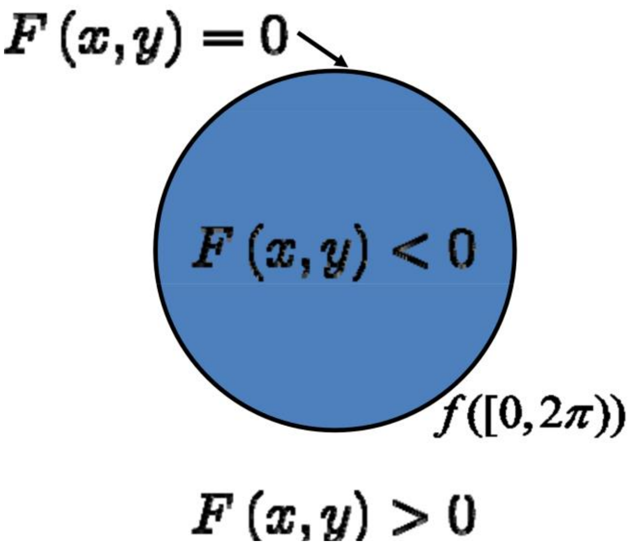
- Implicit:
  - Zero set of distance function

$$F(x,y) = \sqrt{x^2 + y^2} - r$$



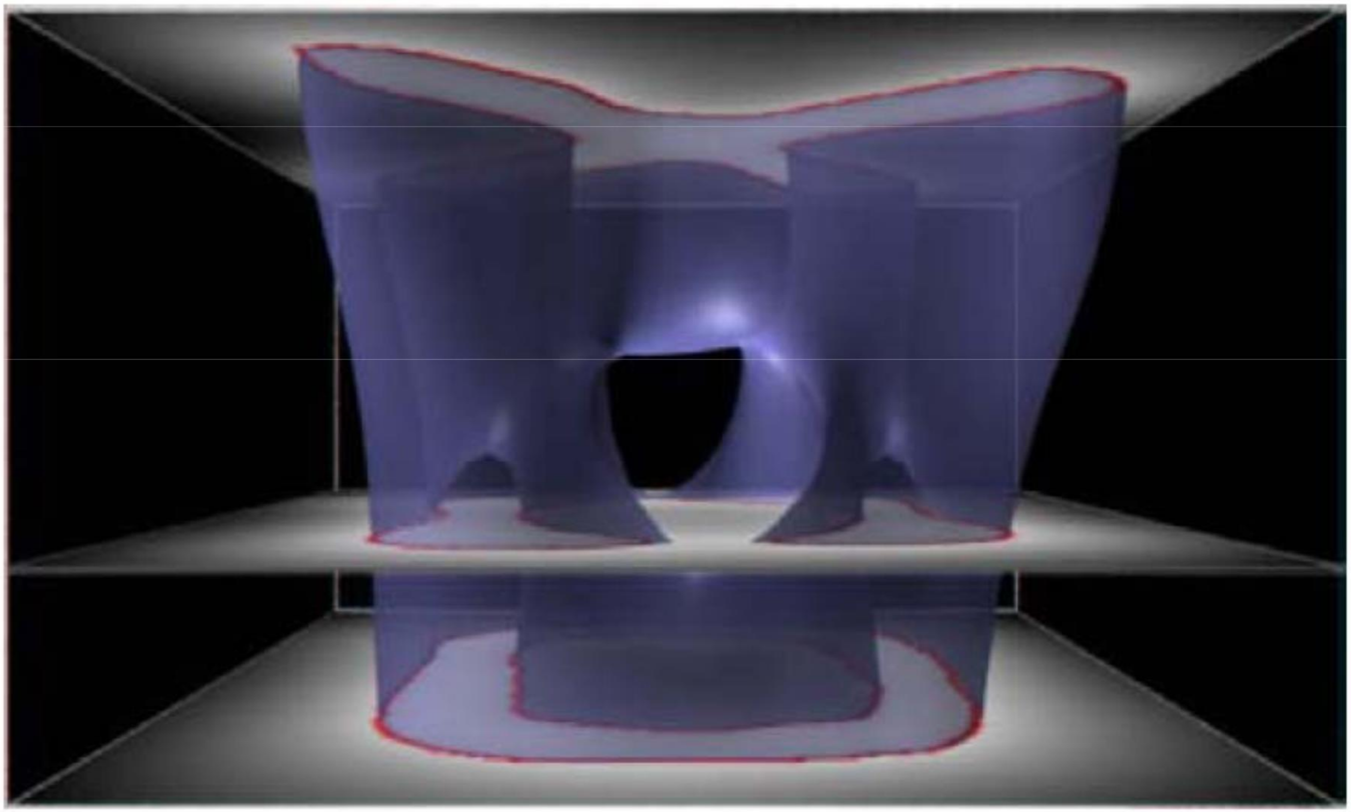
Implicit vs. Explicit

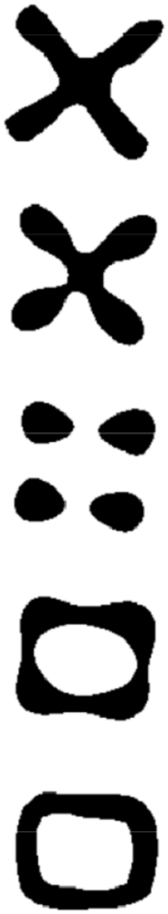
- Explicit:
  - Image of parametrization
  - Easy to find points on surface
  - Can defer problems to param space
- Implicit:
  - Zero set of distance function
  - Easy in/out/distance test
  - Easy to handle different topologies



Implicit Representations

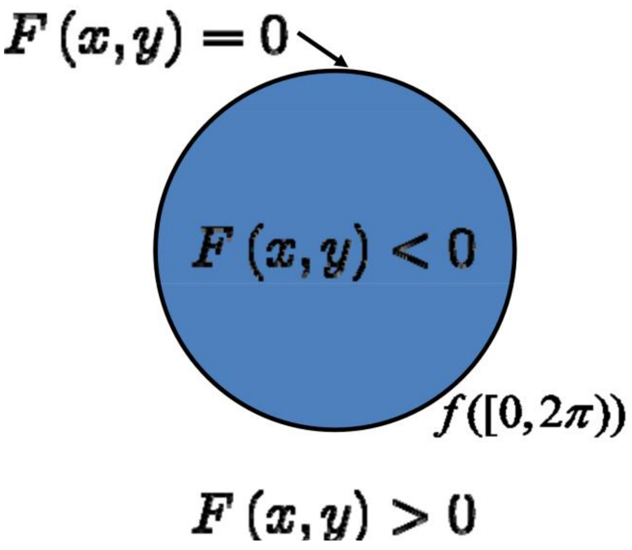
- Easy to handle different topologies



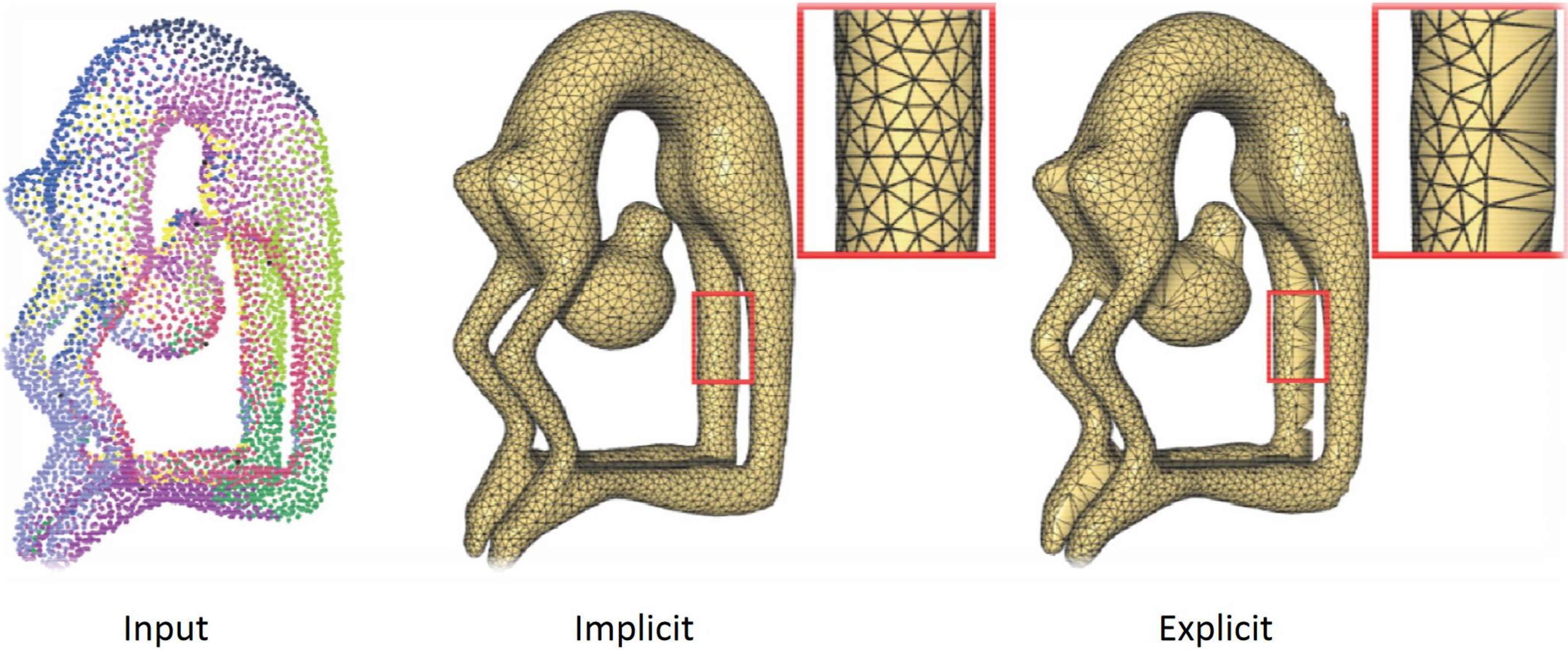


Implicit Representations

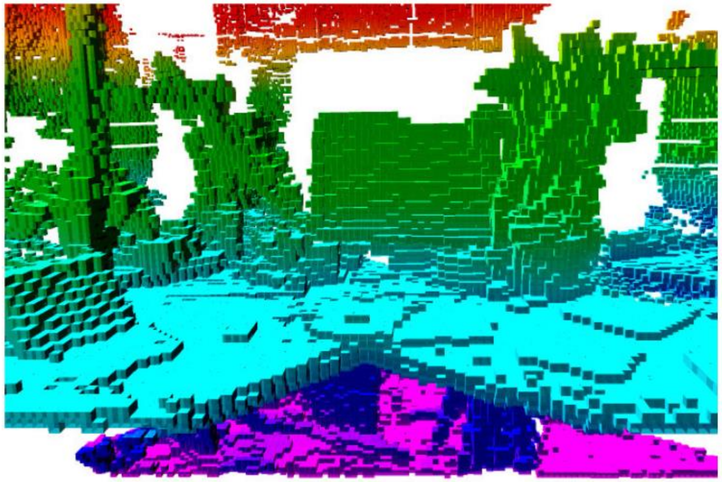
- General implicit function:
  - Interior: $F(x,y,z) < 0$
  - Exterior: $F(x,y,z) > 0$
  - Surface: $F(x,y,z) = 0$



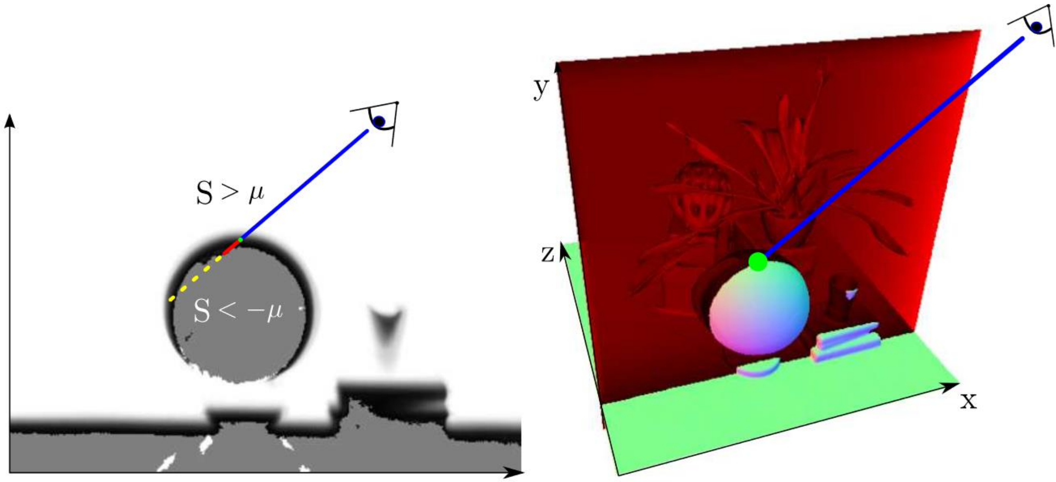
- Special case:
  - Signed distance function (SDF)


Input
Implicit
Explicit

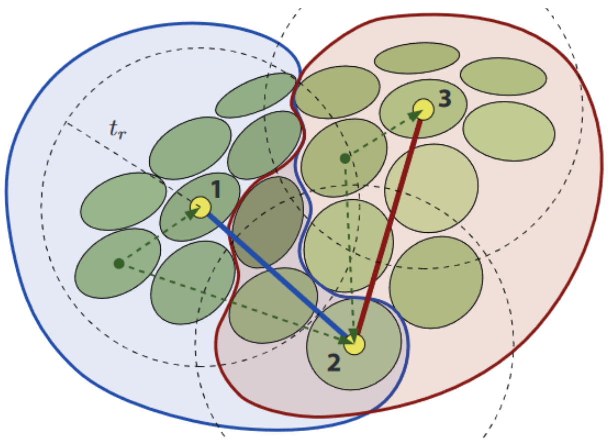
Implicit or Explicit?



Volumetric Occupancy Maps



Volumetric Signed Distance Functions



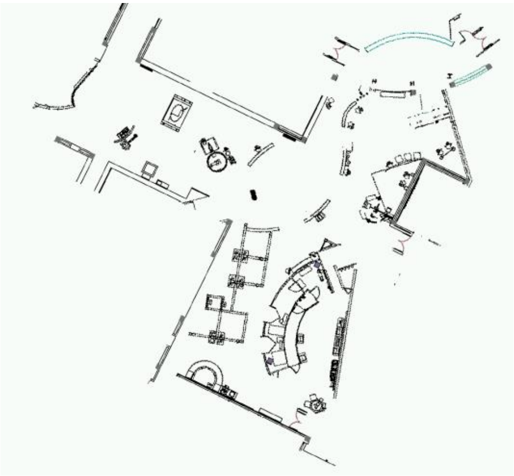
Surfel Splats

Images: Weise et al., 2011; Wurm et al., 2010; Newcombe et al., 2011

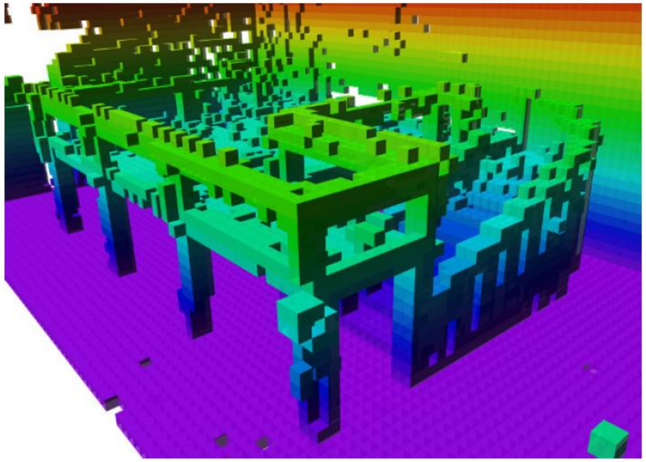
Occupancy Grid Maps

- Idea: Discretize space into grid and represent „occupancy“ of each cell

2D



3D



Images: Thrun et al., 2005; Wurm et al., 2010

Probabilistic Estimation of Occupancy

- Map $M = \{m_1, \ldots, m_S\}$ is a grid of cells
- Each cell state is modelled as a binary random variable $m_i \in \{\mathrm{occ}, \mathrm{empty}\}$ which can take on the values occupied or empty
- We obtain (stochastic) measurements $y_1, \ldots, y_t$ of the cell states
- We assume the probability of each cell state to be stochastically independent from the state of all other cells given the measurements

$$p(M \mid y_1, \ldots, y_t) = \prod_{i=1}^{S} p(m_i \mid y_1, \ldots, y_t)$$

- This means, we can estimate the occupancy probability in each cell individually

Recursive Bayesian Filtering of Occupancy

- Occupancy probability can be estimated recursively

$$p(m \mid y_1, \ldots, y_t) = \frac{p(y_t \mid m)\, p(m \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}$$
$$= \frac{p(m \mid y_t)\, p(y_t)\, p(m \mid y_{1:t-1})}{p(m)\, p(y_t \mid y_{1:t-1})}$$

- Note the use of the inverse sensor model $p(m \mid y_t)$
- Log-odds simplifies calculations and improves numeric stability

$$l(m = \mathrm{occ} \mid y_{1:t}) = \log\left(\frac{p(m = \mathrm{occ} \mid y_{1:t})}{1 - p(m = \mathrm{occ} \mid y_{1:t})}\right)$$
$$= \log\left(\frac{p(m = \mathrm{occ} \mid y_t)}{1 - p(m = \mathrm{occ} \mid y_t)}\right) - \log\left(\frac{p(m = \mathrm{occ})}{1 - p(m = \mathrm{occ})}\right) + l(m = \mathrm{occ} \mid y_{1:t-1})$$

Inverse Sensor Model

- Typical inverse sensor model for range sensors

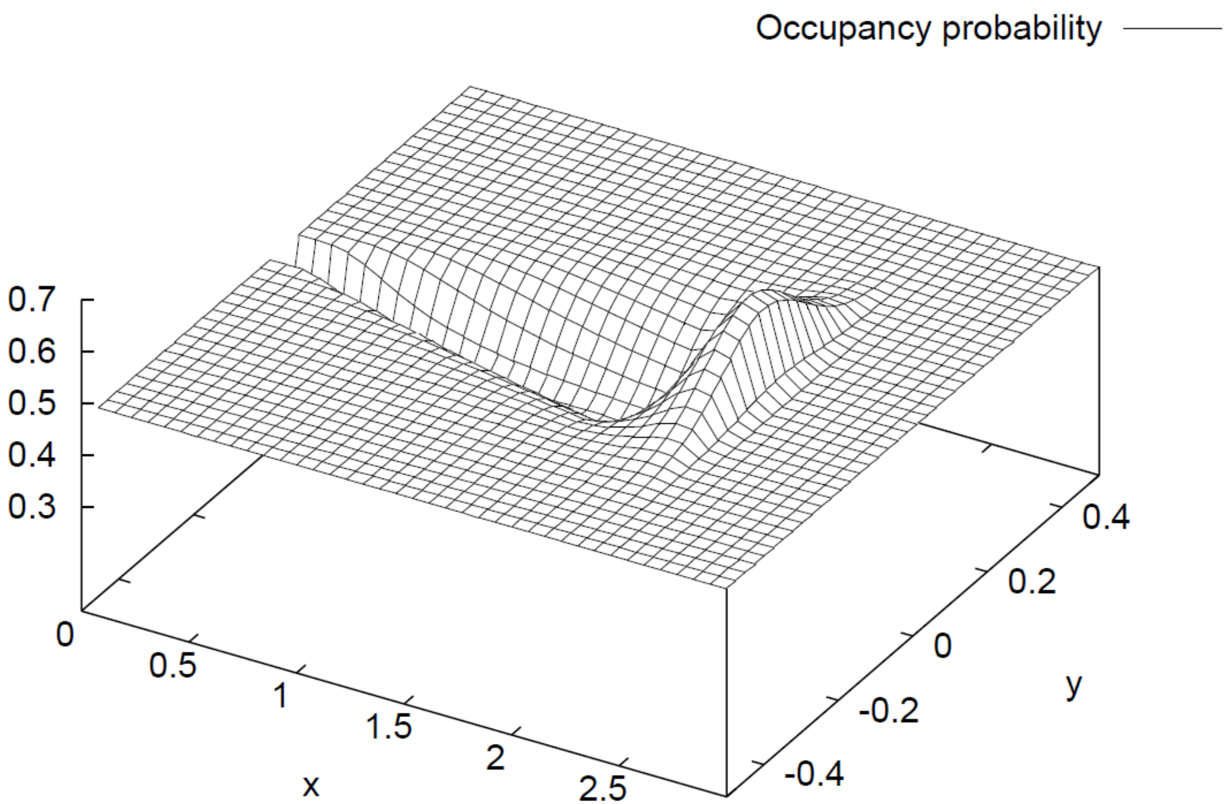


Image: C. Stachniss, 2006

Inverse Sensor Model

- Typical inverse sensor model for range sensors

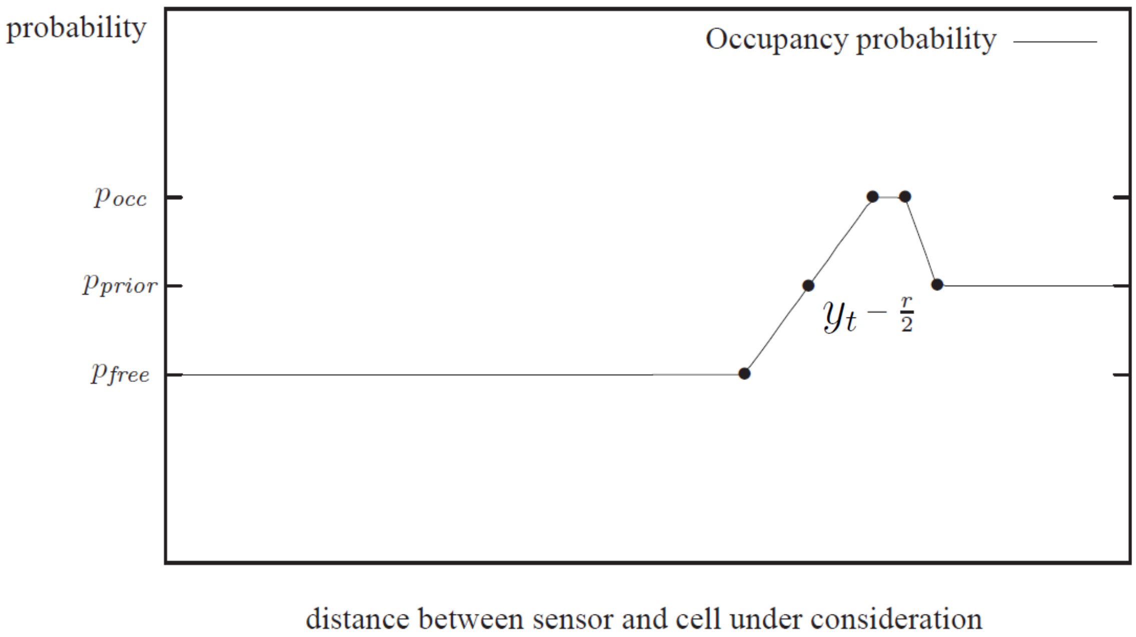


Image: C. Stachniss, 2006

Example: 2D Mapping with Sonar Sensors

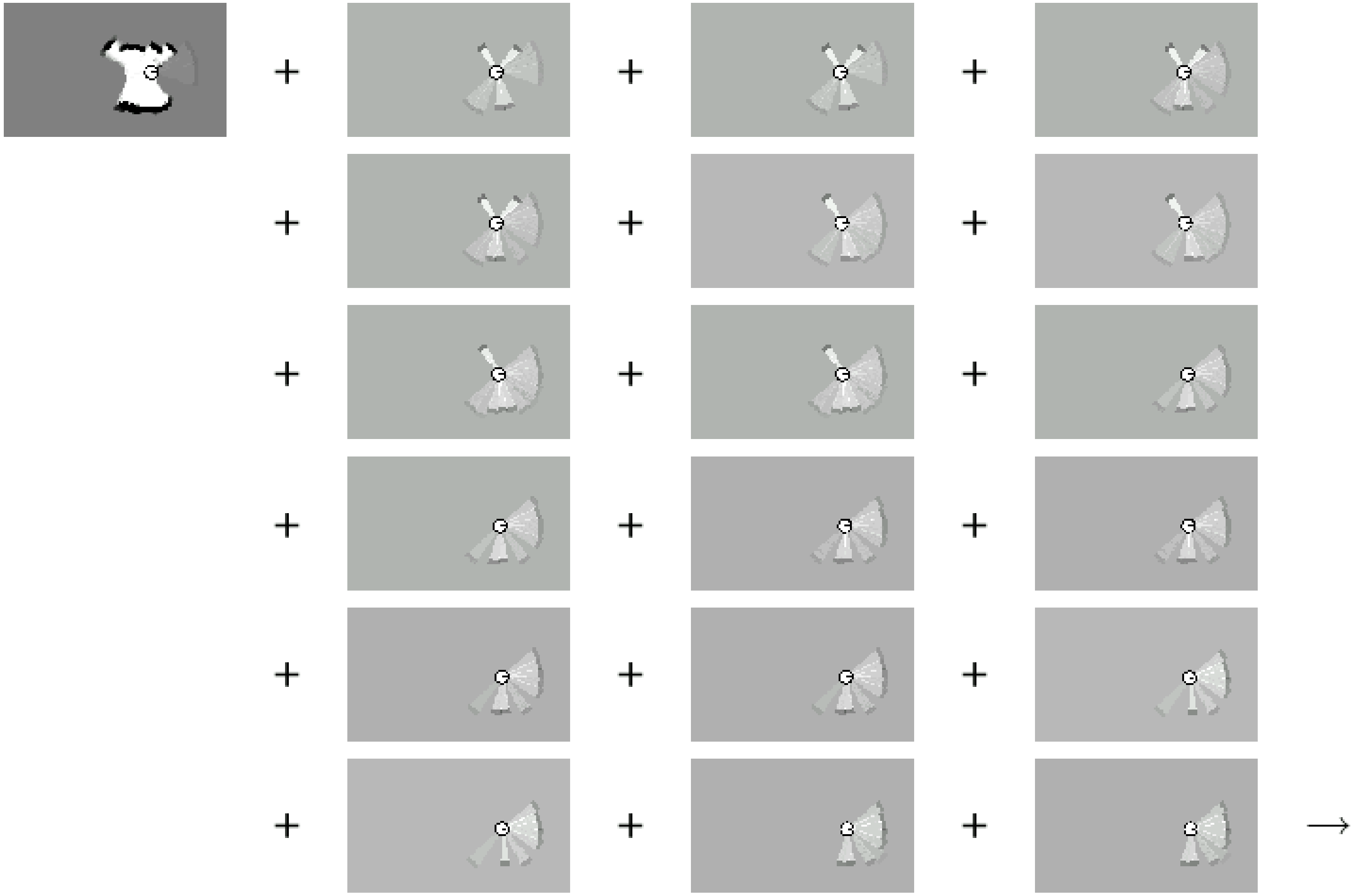


Image: Thrun et al., 2005

Example: 2D Mapping with Sonar Sensors

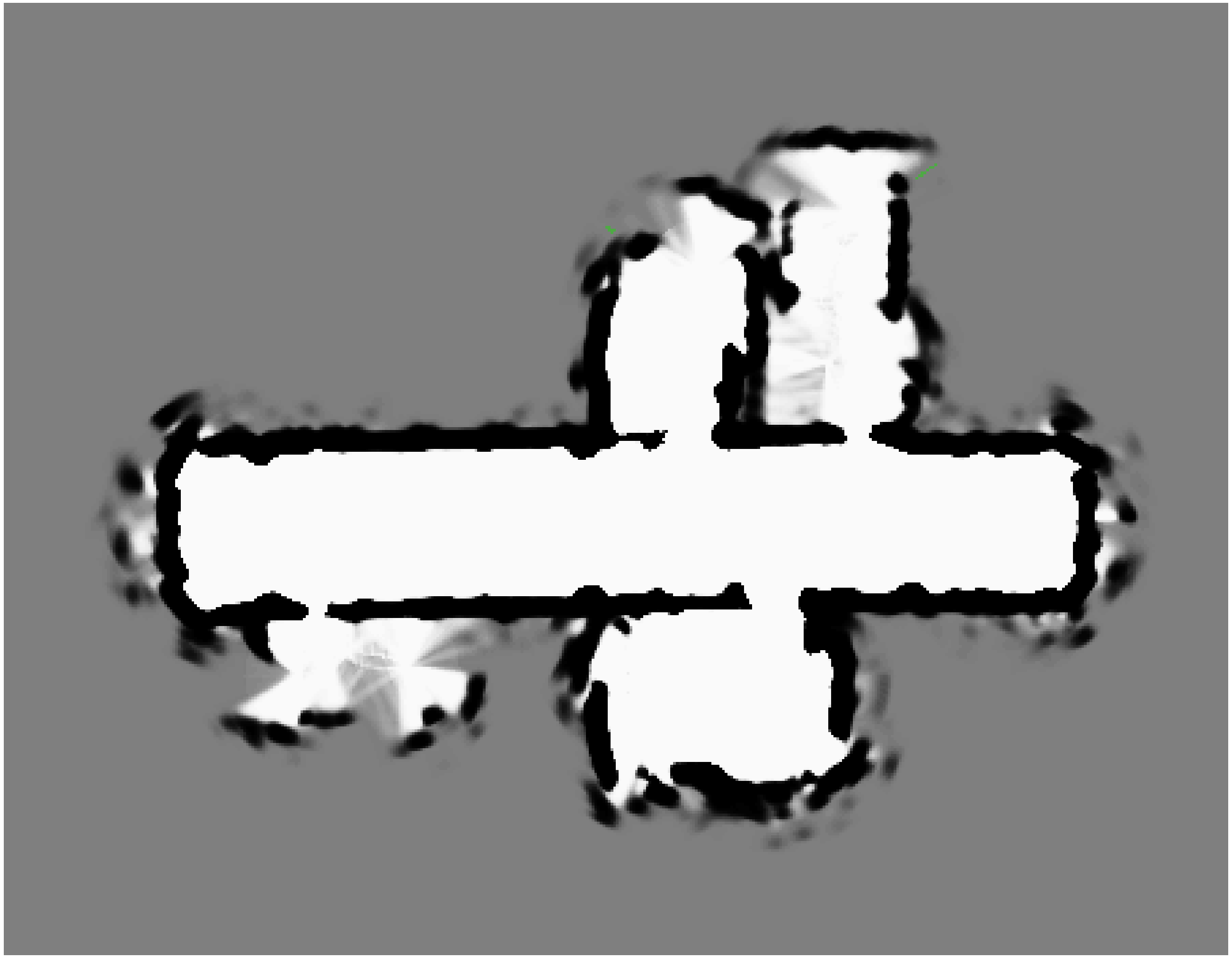
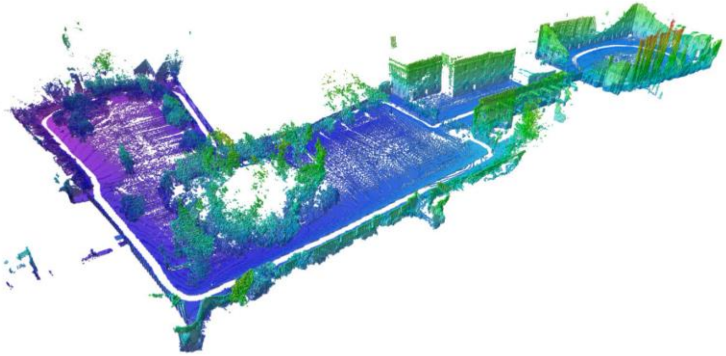


Image: Thrun et al., 2005

Memory Consumption

- 2D floor map of a 40m x 40m building at 0.05m resolution allocates $\frac{40^2}{0.05^2} = 640000$ cells (5.12 MB at double precision)

- 3D volumetric map with size 40x40x40m at 0.05m resolution needs $\frac{40^3}{0.05^3} = 512,000,000$ cells (4.096 GB at double precision)

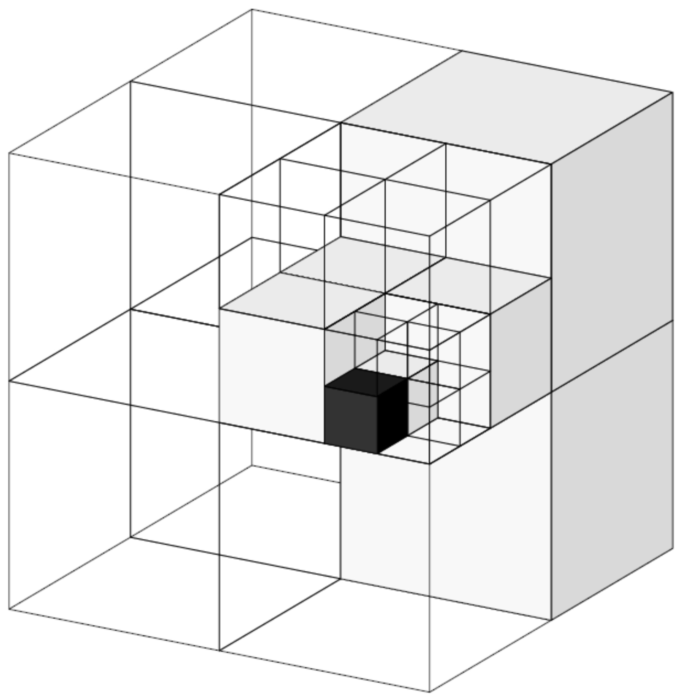


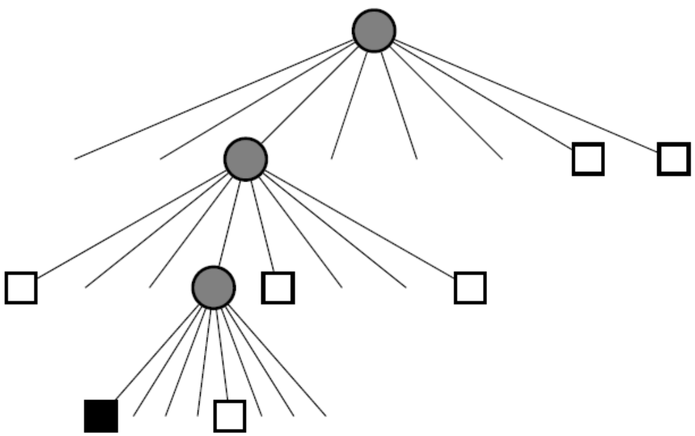
- Memory consumption quickly gets huge!
- Likely large volumes will be empty!
- What can we do?

Images: Thrun et al., 2005; Wurm et al., 2010

3D Occupancy Maps in Octrees

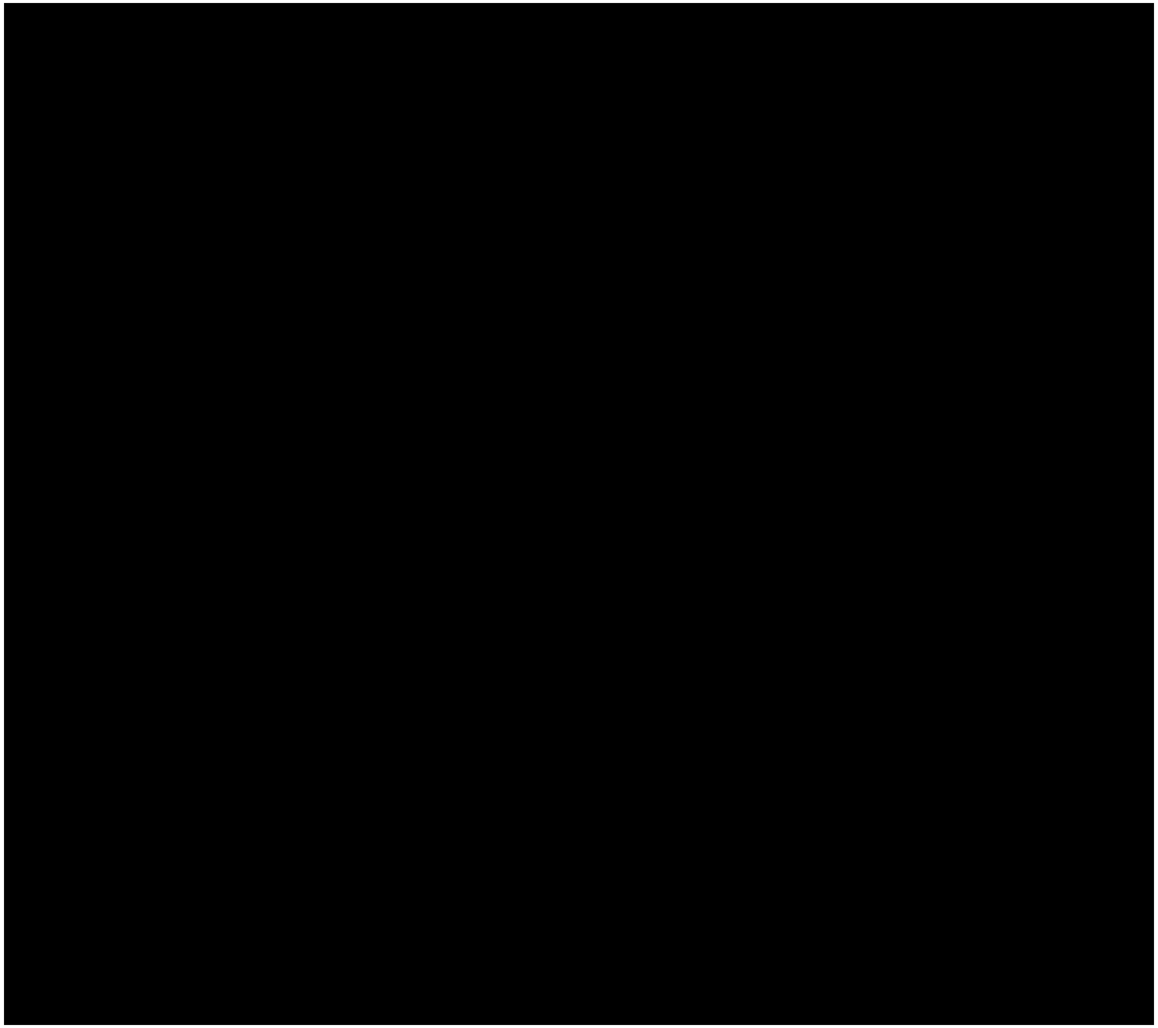
- Only allocate observed voxels
- Recursively subdivide map volume: multi-resolution





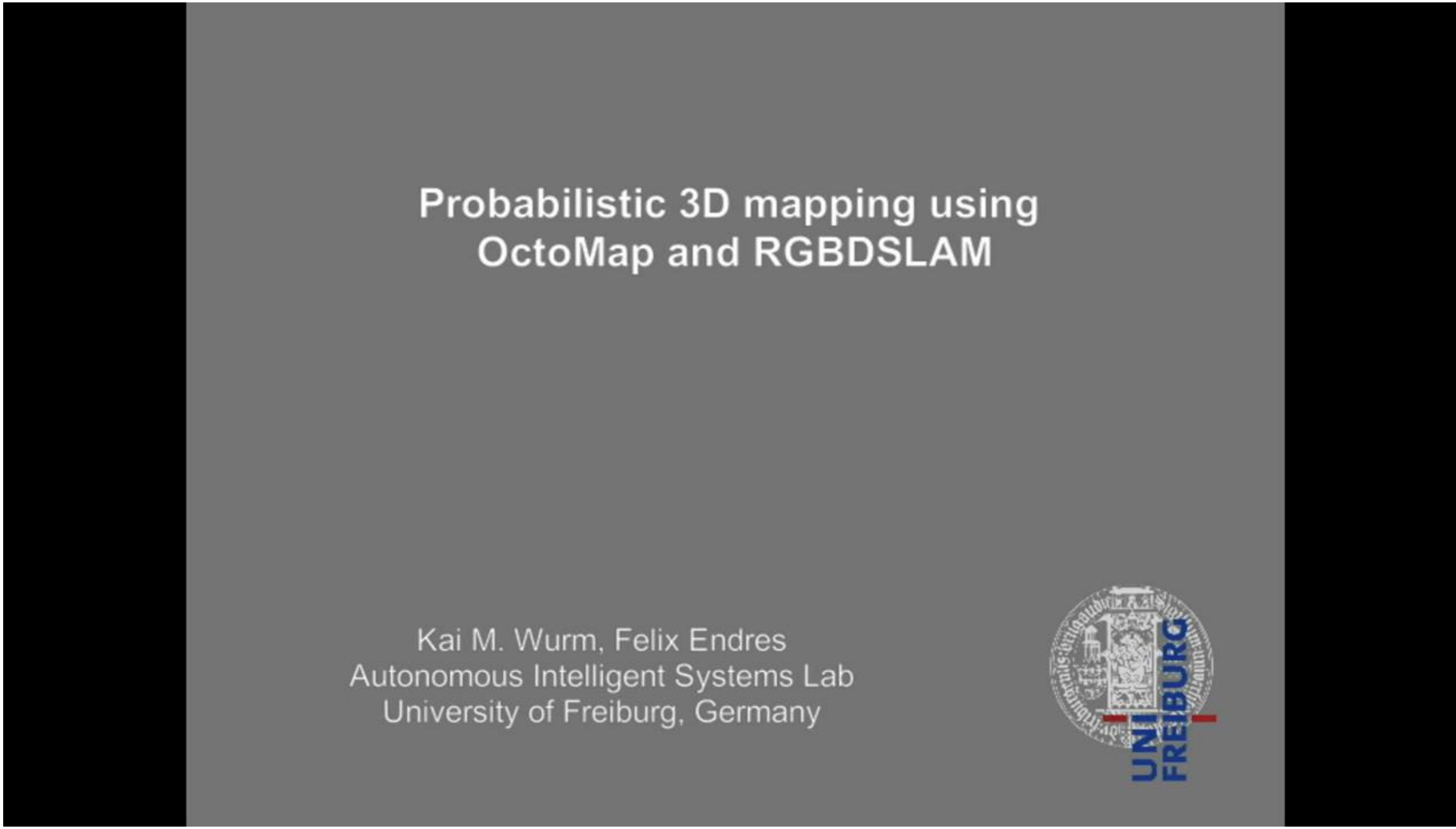
Images: Wurm et al., 2010

Example: OctoMap & RGB-D SLAM



Endres et al., 3D Mapping with RGB-D Cameras, TRO, 2014
Hornung et al., OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, 2013

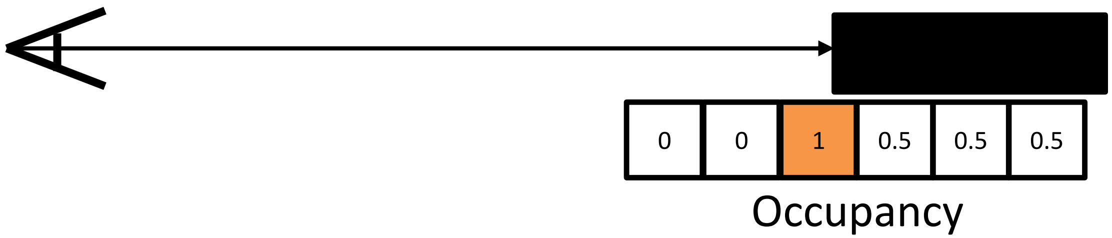
Example: OctoMap & RGB-D SLAM



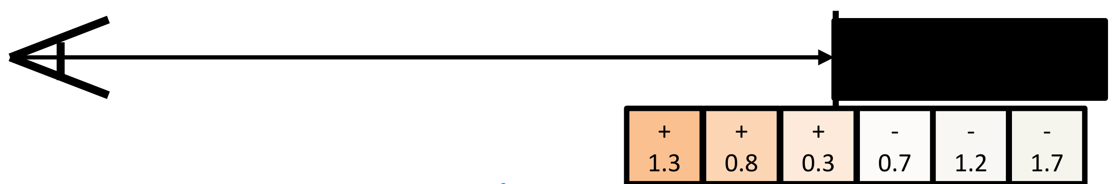
Endres et al., 3D Mapping with RGB-D Cameras, TRO, 2014
Hornung et al., OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, 2013

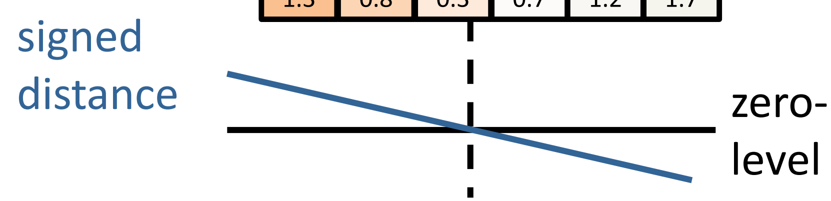
Signed Distance Function (SDF)

- Occupancy grid maps estimate occupancy of voxels
  - Surface only coarsely approximated



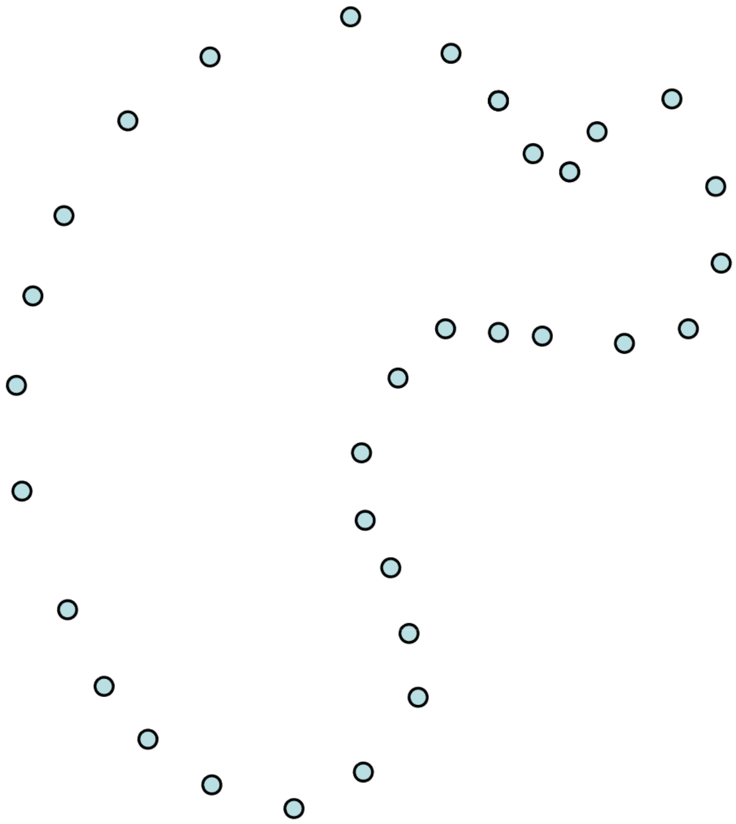
- Idea:
  - Instead of occupancy, store the distance from the surface in the grid cells
  - Represent inside/outside the object using the sign





  - We can find the zero-level through interpolation!

SDF Approach

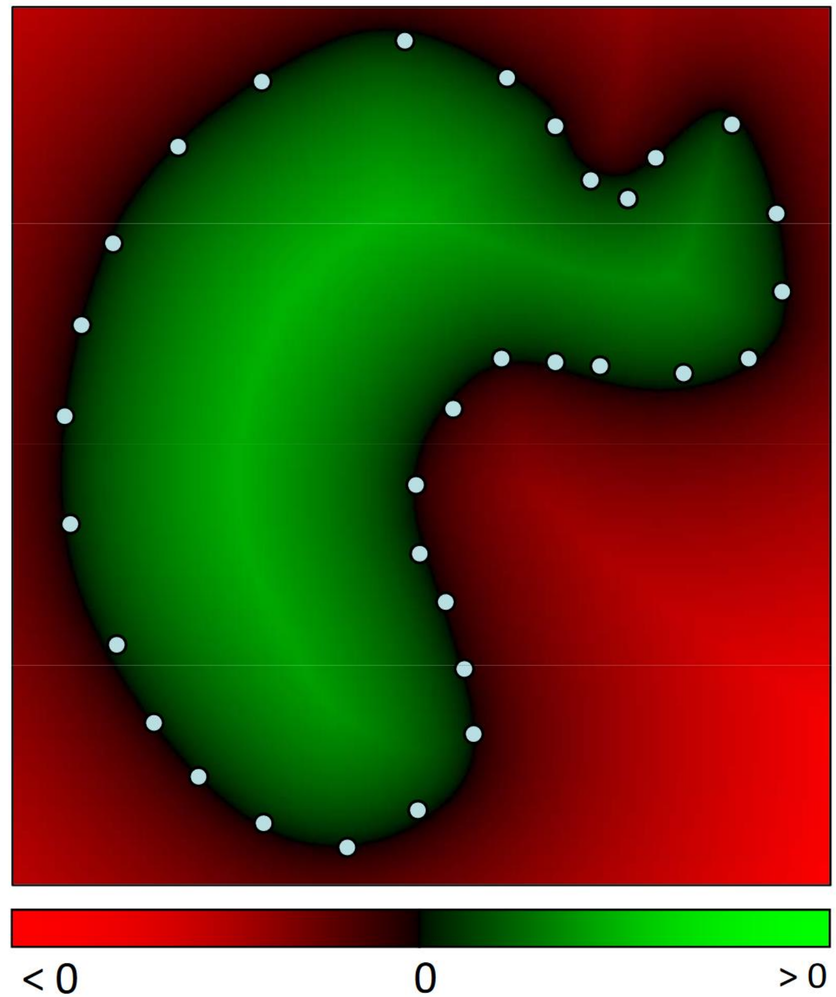


SDF Approach

- Define a function

$$f : R^3 \rightarrow R$$

with value < 0 outside and value > 0 inside object



SDF Approach

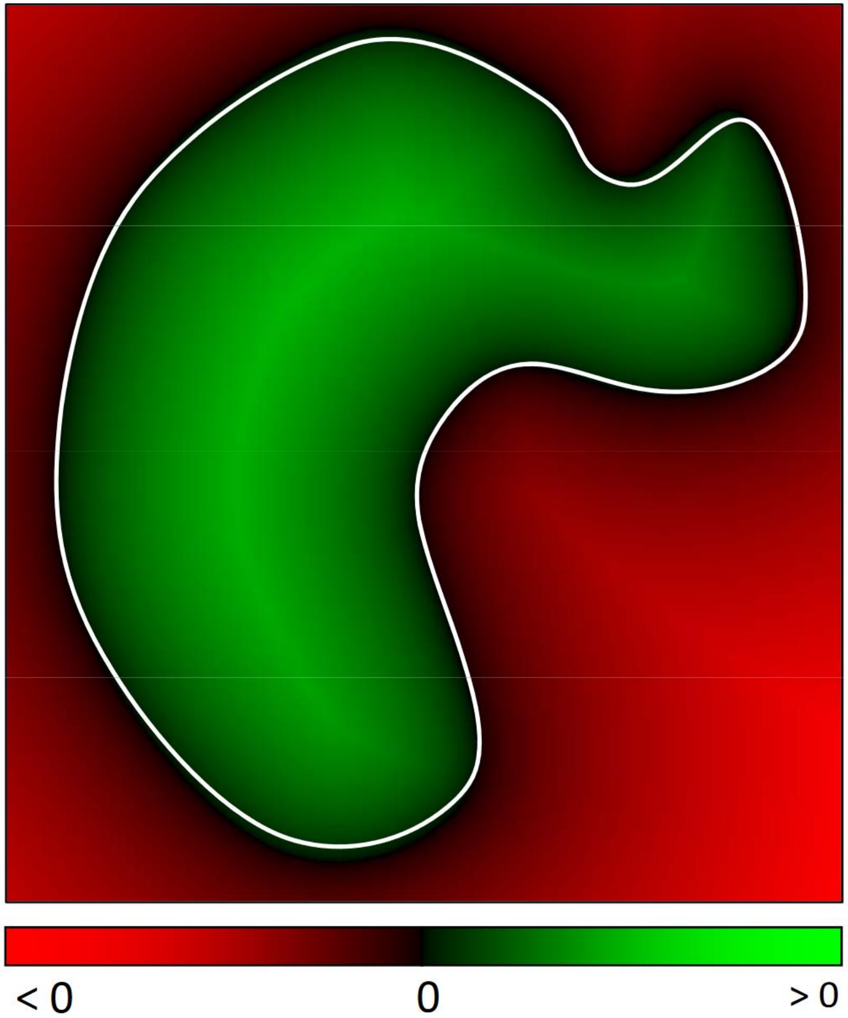
- Define a function

  $$f : R^3 \rightarrow R$$

  with value < 0 outside and value > 0 inside object

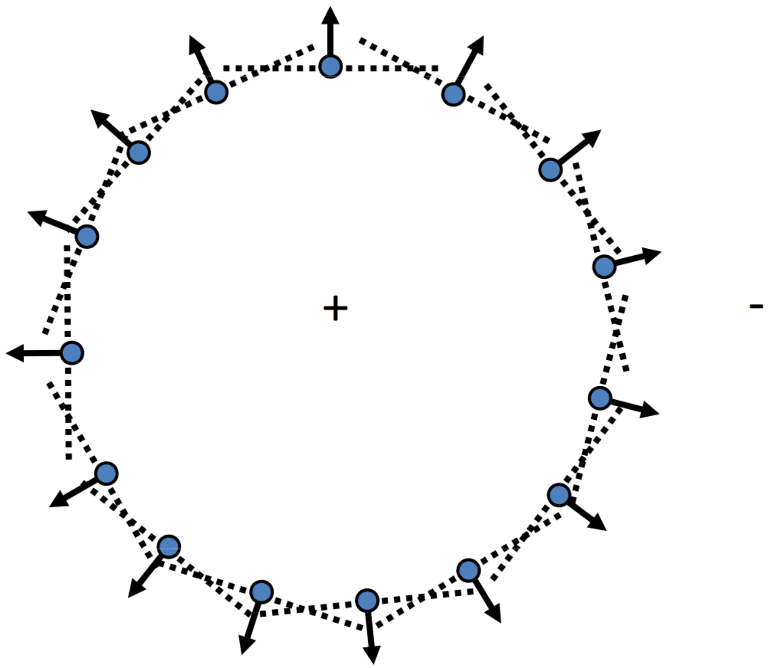
- Extract zero-level set

  $$\{x : f(x) = 0\}$$



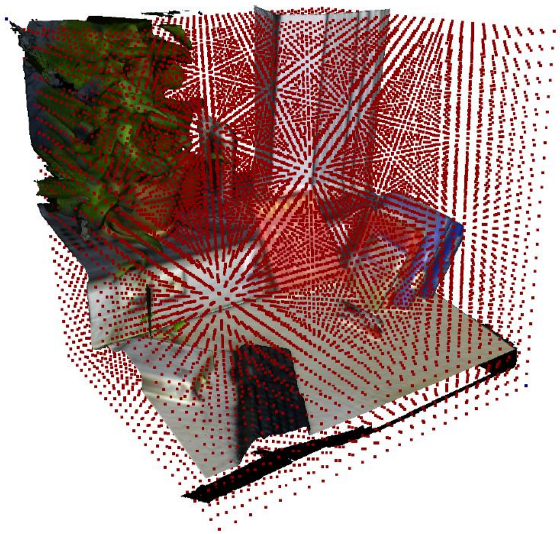
SDF from Point Sets

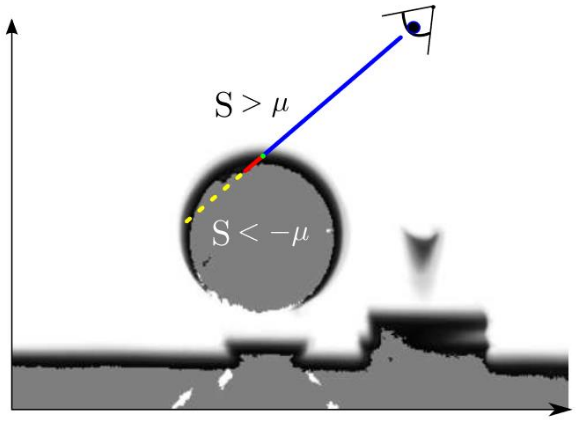
- Distance to points not sufficient
- Approximate surface locally linear: point and normal
- Determine closest distance to points along normals
- Inside/outside from normal direction
- Smooth approximation using radial basis function (RBF) kernels

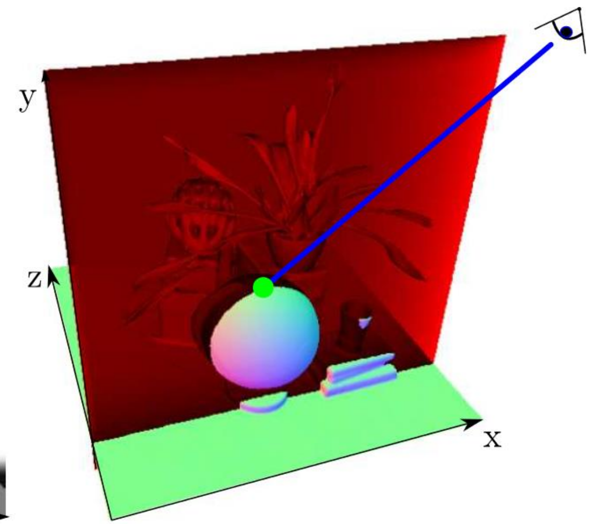


$$dist(\mathbf{x}) = \sum_i w_i \varphi(\|\mathbf{x} - \mathbf{c}_i\|)$$

SDFs for 3D Map Representation



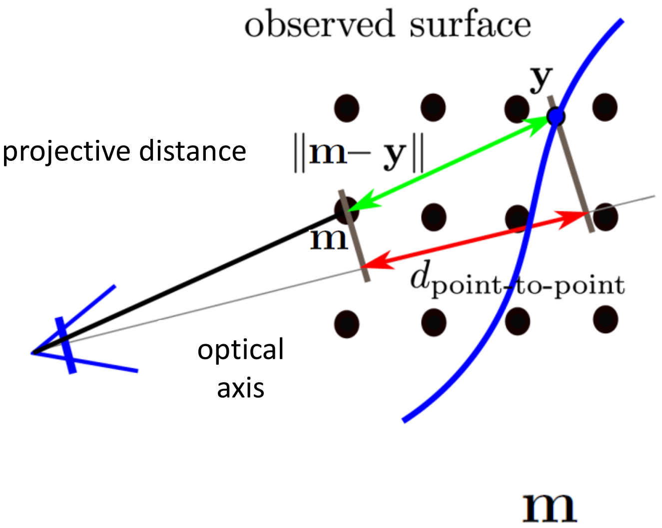




Images: Bylow et al., 2013; Newcombe et al., 2011

Projective SDFs from Depth Images

- Given: Depth images, camera intrinsics, camera poses
- The depth images observe distance of camera view point to surface
  - Approximate closest distance from surface with projective distance
  - Further approximation: use distance along optical axis, i.e. depth



- Estimate weighted average of observed distances to each voxel

$$\psi_t(\mathbf{m}) = \frac{D_t(\mathbf{m})}{W_t(\mathbf{m})}$$

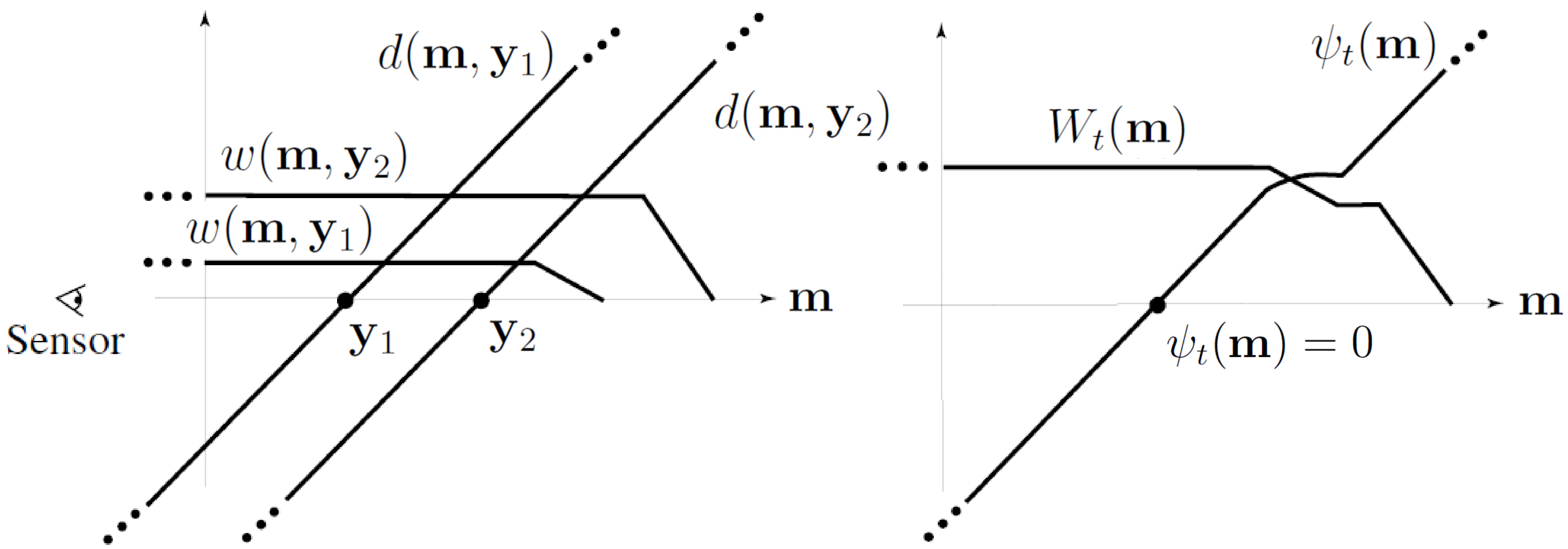
SDF

$$D_t(\mathbf{m}) = D_{t-1}(\mathbf{m}) + w(\mathbf{m}, \mathbf{y}_t)\, d(\mathbf{m}, \mathbf{y}_t)$$
$$W_t(\mathbf{m}) = W_{t-1}(\mathbf{m}) + w(\mathbf{m}, \mathbf{y}_t)$$

Images: Bylow et al., 2013; Izadi et al., 2011

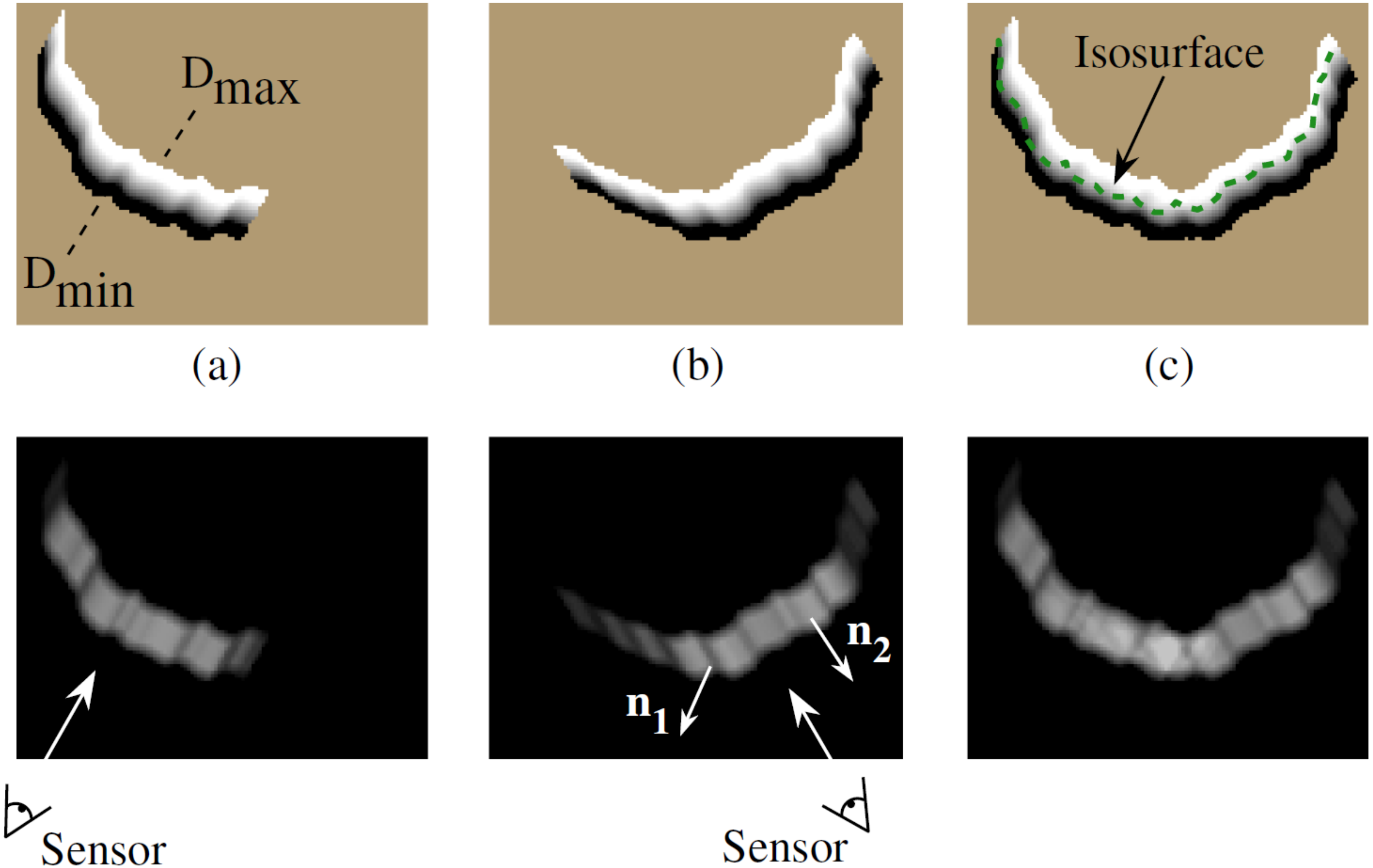
Weighting Functions

- Weighting function represents „confidence“ in the distance measurement



Images: Curless and Levoy, 1996

SDF: 2D Example



Images: Curless and Levoy, 1996

Further Insights

- Curless and Levoy, 1996, showed that using orthographic projection, the zero crossing of the integrated signed distance function is the least squares surface fit to the distances

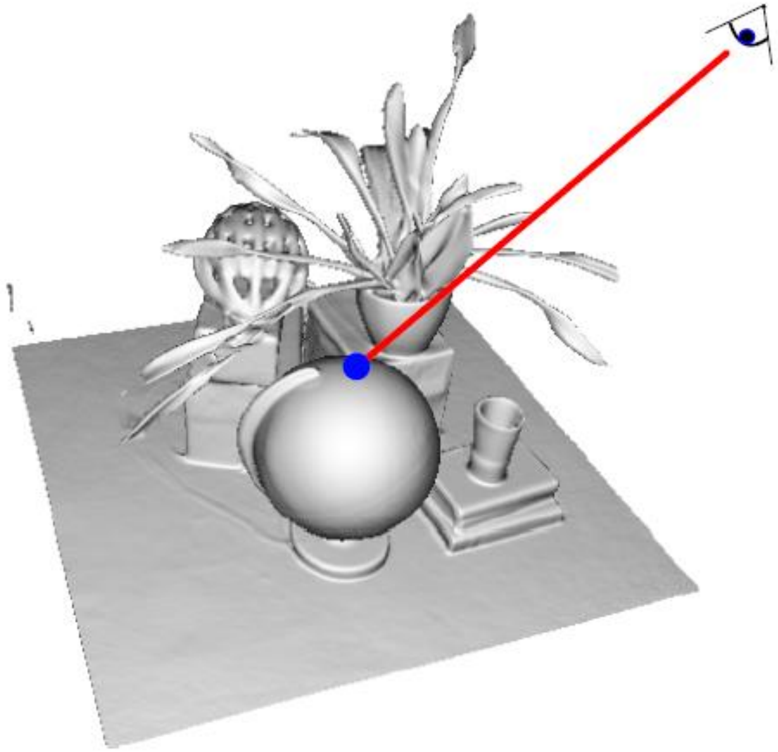
- Typically, noise cancels out over multiple measurements

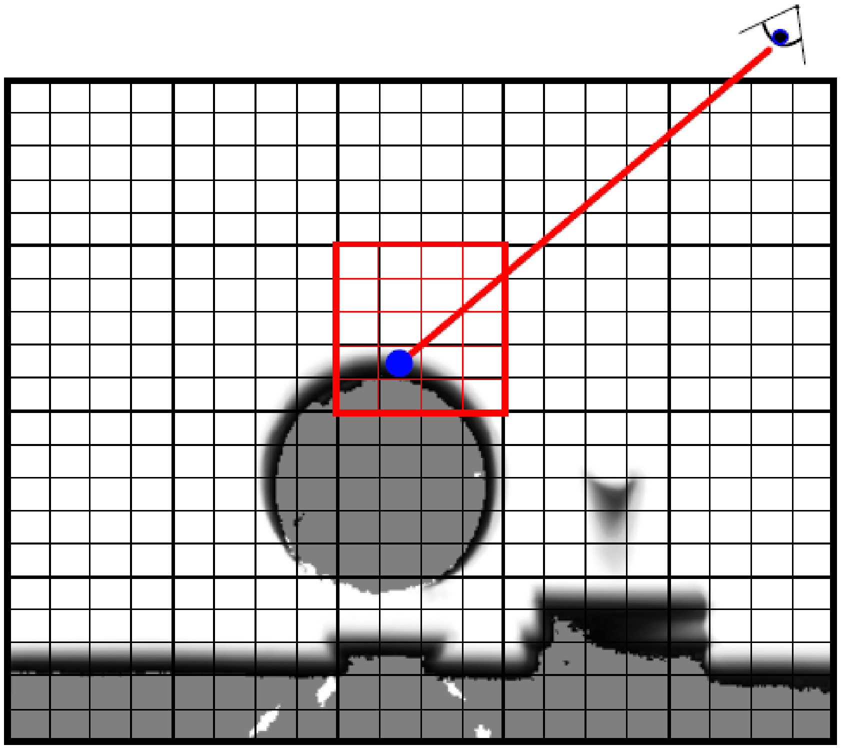
- Often, one limits the integration range to a narrow band around the zero level-set to increase efficiency and allow for thin objects. The signed distance function is then called truncated SDF (TSDF).

- The surface corresponds to the zero-level set
  - To generate a depth image from a novel view, it can be efficiently extracted using raycasting
  - A triangular mesh can be extracted using the Marching Cubes algorithm

Raycasting

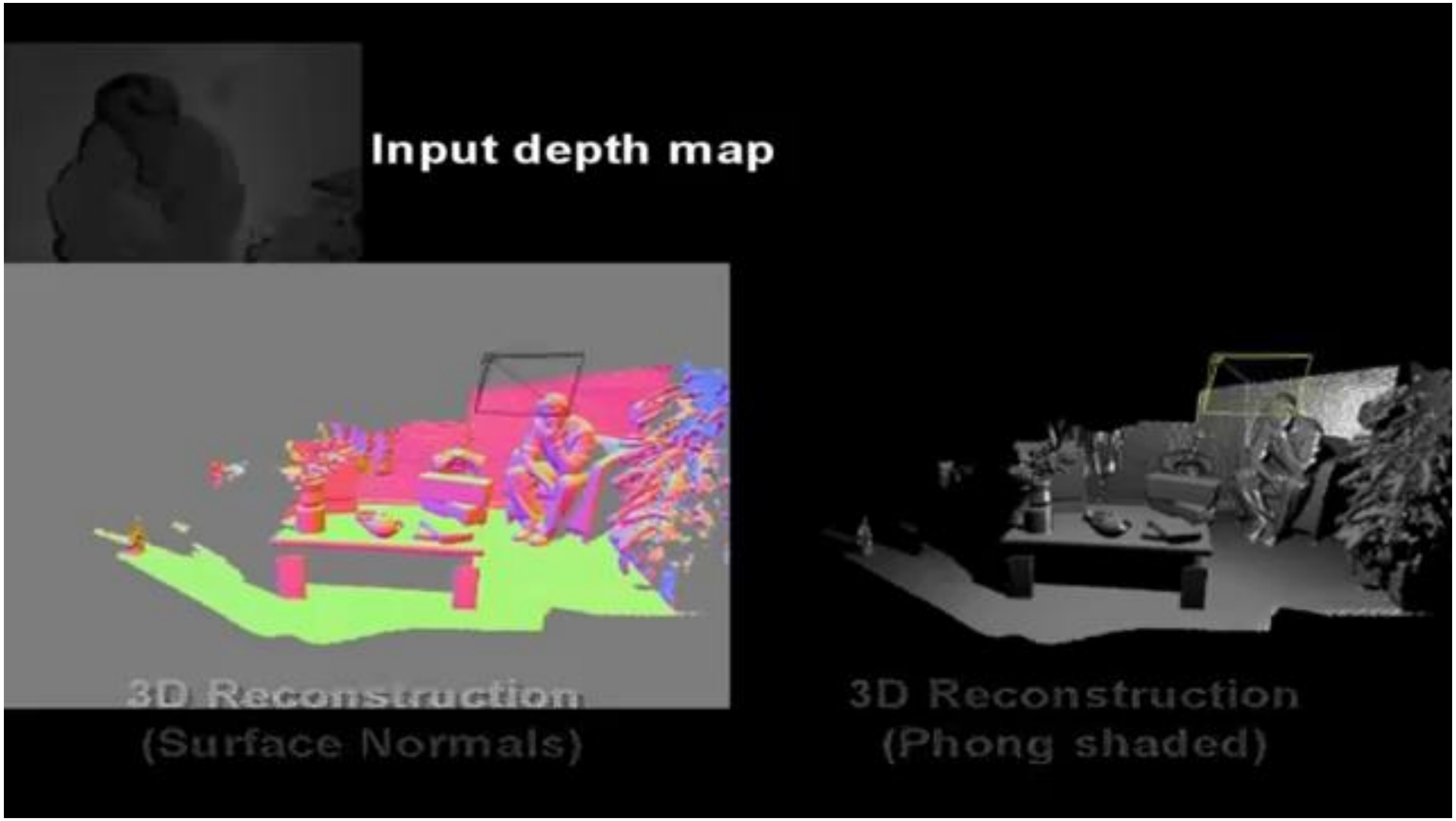
- For each pixel in the novel view, cast a ray to find the first zero-crossing





Example: KinectFusion

- Tracking: Render depth image from current pose, align image
- Mapping: TSDF integration of current image from tracked pose



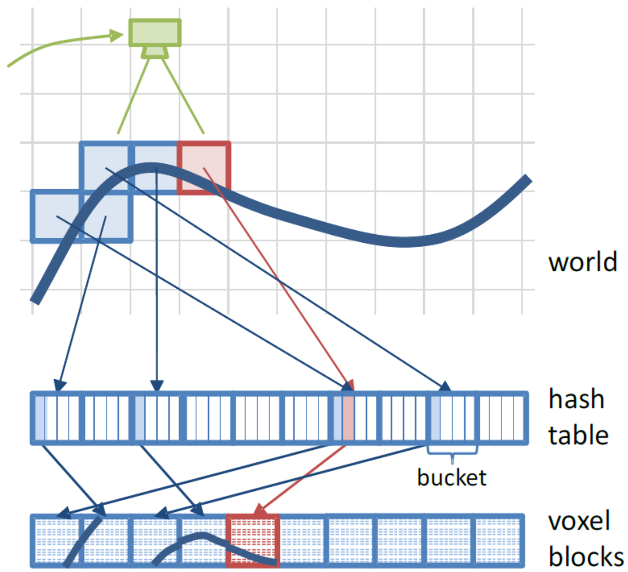
Newcombe et al., KinectFusion, ISMAR 2011

Voxel Hashing and Octrees for TSDFs

- Memory consumption of fully allocated volumetric grid representations of TSDFs also is cubic in environment size and inverse cell size
- How to scale TSDF maps to larger environments at higher resolution?

- Idea 1:
  - Only allocate voxels close to the updated narrow band along the surface
  - Index voxels through hashing

- Idea 2:
  - Use octree to represent TSDF
  - Also incorporate voxel hashing (idea 1)
  - Nice feature: multi-resolution TSDF



Hash function in voxel position (x,y,z):

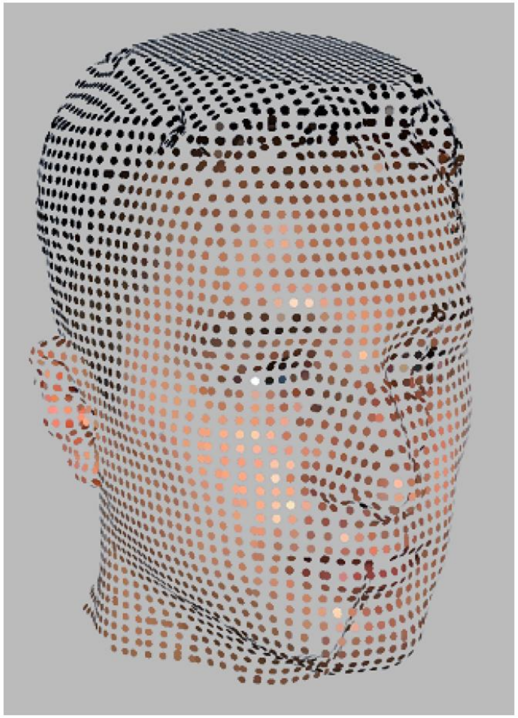
$$H(x,y,z) = (x \cdot p_1 \oplus y \cdot p_2 \oplus z \cdot p_3)\, mod\, n$$

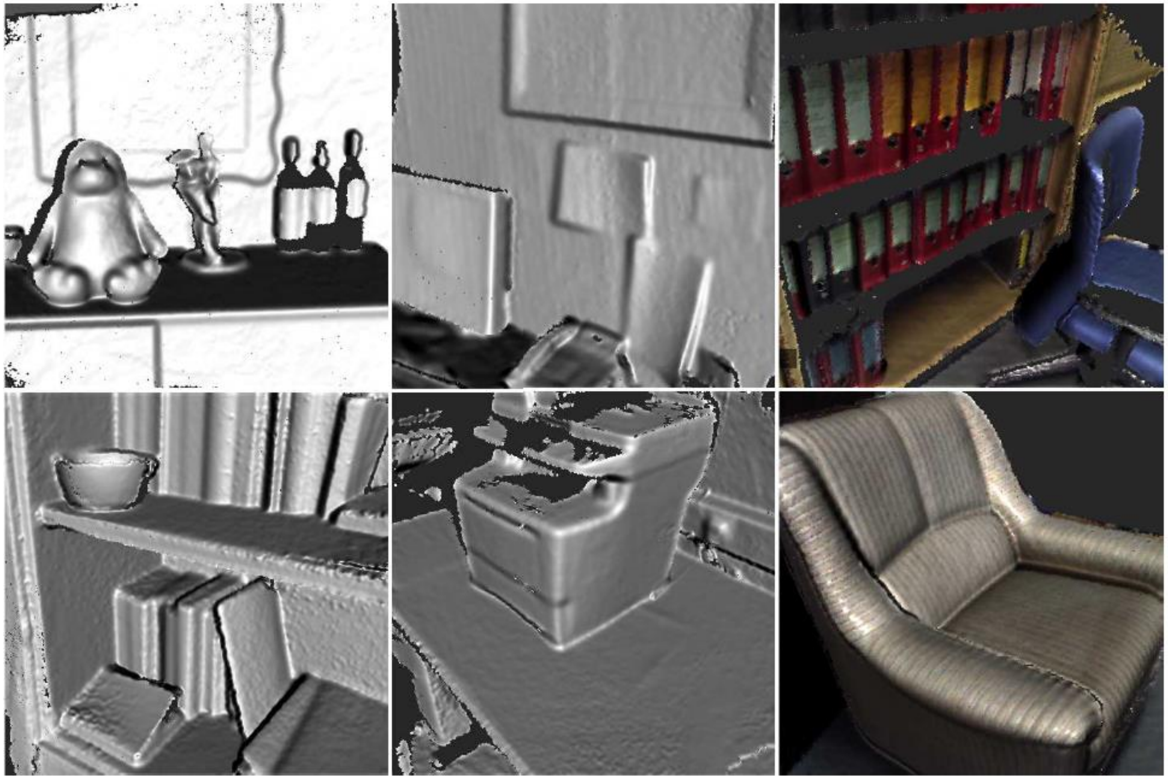
Example: TSDF Voxel Hashing

Nießner et al., Real-time 3D Reconstruction at Scale using Voxel Hashing, SIGGRAPH Asia, 2013

Surfel Map Representation

- Represent map as a set of surfel splats
- Surfel splat: point+normal+radius





Images: M. Zwicker, Keller et al., 2013

Surfel Map Representation

- Represent map $M = \{m_1, \ldots, m_S\}$ set of surfel splats
- Surfel splat $m_i = \{\mathbf{x}_i, \mathbf{n}_i, r_i, c_i, t_i\}$ consists of 3D position $\mathbf{x}_i$, normal $\mathbf{n}_i$, radius r_i , confidence c_i, and time of last observation t_i
- Surfel splats are associated with pixels in depth image through raycasting
- Fusion of point/normal measurement $\mathbf{x}_{t,y}$, $\mathbf{n}_{t,y}$ with associated surfel splat

$$\mathbf{x}_{t,i} = \frac{c_{t-1,i}\mathbf{x}_{t-1,i} + \alpha \mathbf{x}_{t,y}}{c_{t-1,i} + \alpha} \qquad \mathbf{n}_{t,i} = \frac{c_{t-1,i}\mathbf{n}_{t-1,i} + \alpha \mathbf{n}_{t,y}}{c_{t-1,i} + \alpha}$$

$$c_{t,i} = c_{t-1,i} + \alpha$$

$$t_i = t$$

$$\alpha = \exp\left(-\frac{\gamma^2}{2\sigma^2}\right)$$

radial distance from optical center

noise parameter

- Unassociated pixels initiate new surfel splats, radius set in proportion to depth

Example: Point-Based Fusion

Real-time 3D Reconstruction in Dynamic Scenes using Point-based Fusion

Maik Keller
pmdtechnologies

Damien Lefloch
University of Siegen

Martin Lambers
University of Siegen

Shahram Izadi
Microsoft Research

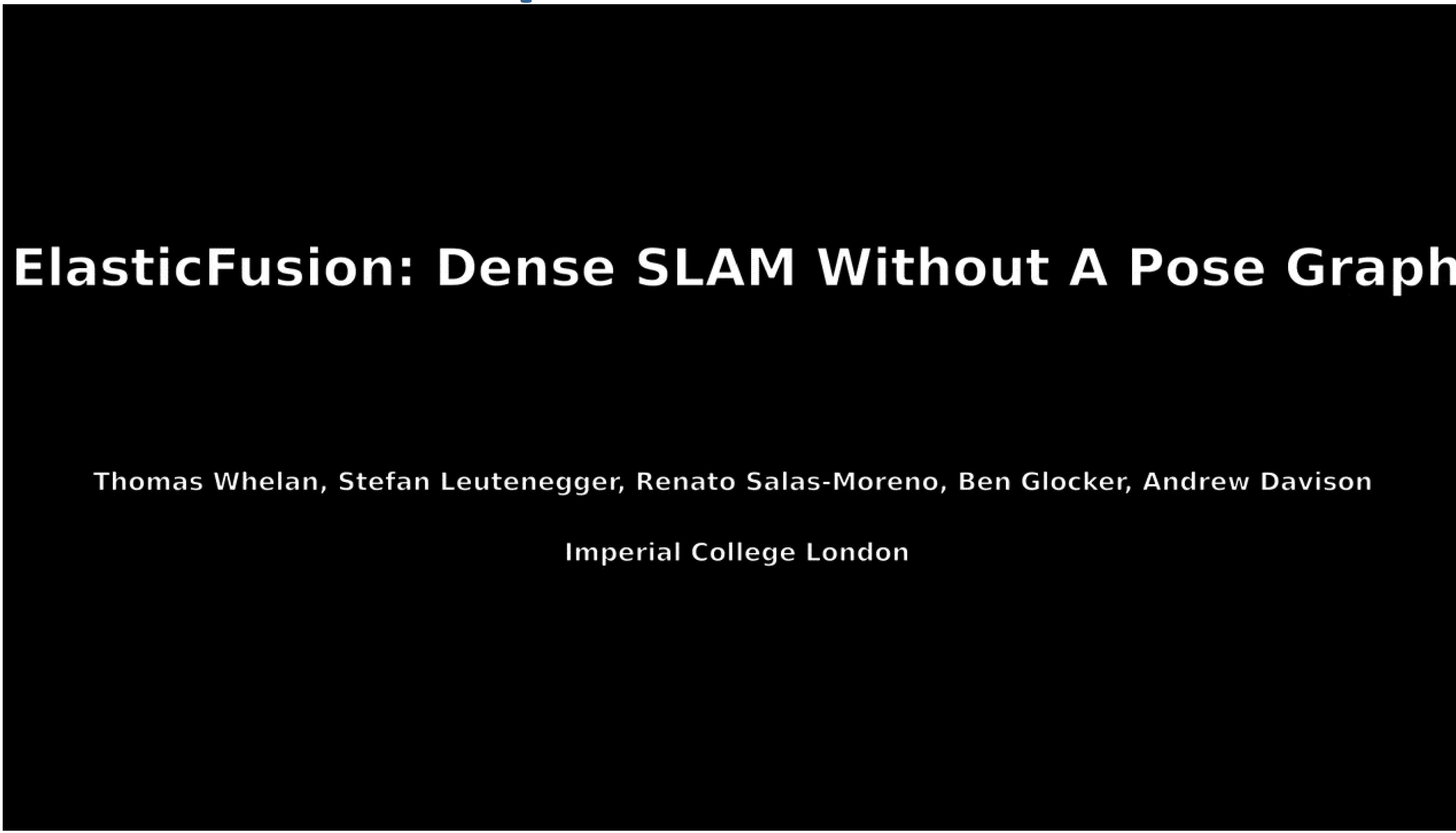
Tim Weyrich
University College London

Andreas Kolb
University of Siegen

3DV 2013

Keller et al., Real-time 3D Reconstruction in Dynamic Scenes using Point-Based Fusion, 3DV 2013

Example: ElasticFusion



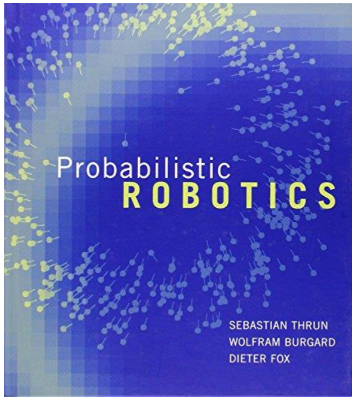
Whelan et al., ElasticFusion: Dense SLAM Without A Pose Graph, RSS 2015

Lessons Learned Today

- Dense 3D map representations useful for augmented / virtual reality and robot navigation and exploration
- 3D occupancy grid maps
  - Implicit volumetric surface representation: occupancy probability in grid cells
  - Recursive Bayesian estimation using log-odds filter and inverse sensor model
- 3D truncated signed distance functions (TSDFs)
  - Implicit volumetric surface representation: distance to surface in grid cells
  - Recursive weighted average of distance measurements to surface
- Improve memory efficiency of volumetric representations through octrees and voxel hashing
- 3D surfel representation (explicit)

Further Reading

- Probabilistic Robotics textbook



Probabilistic Robotics, S. Thrun, W. Burgard, D. Fox, MIT Press, 2005

- Publications:
  - Curless and Levoy, A Volumetric Method for Building Complex Models from Range Images, Proc. of Annual Conf. on Computer Graphics and Interactive Techniques, 1996
  - Newcombe et al., KinectFusion: Real-Time Dense Surface Mapping and Tracking, ISMAR 2011
  - Hornung et al., OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, 2013
  - Nießner et al., Real-time 3D Reconstruction at Scale using Voxel Hashing, SIGGRAPH Asia, 2013
  - Keller et al., Real-time 3D Reconstruction in Dynamic Scenes using Point-Based Fusion, 3DV 2013
  - Whelan et al., ElasticFusion: Dense SLAM Without A Pose Graph, RSS 2015

Thanks for your attention!