Robotic 3D Vision

Lecture 19: Map Representations

WS 2020/21
Dr. Niclas Zeller
Artisense GmbH

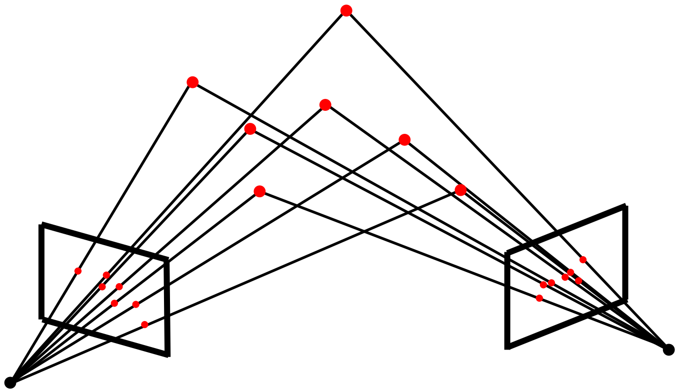
What We Will Cover Today

- Depth Sensors (leftover from last lecture)
  - Structured light
  - Time-of-flight

- Dense map representations
  - Occupancy maps
  - Signed distance functions

Recap: Dense Depth from Two Views

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



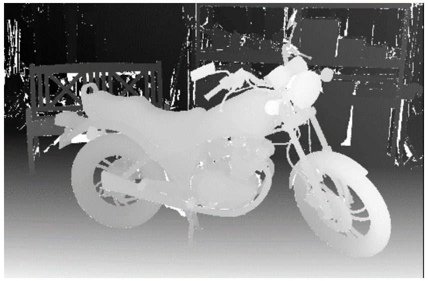
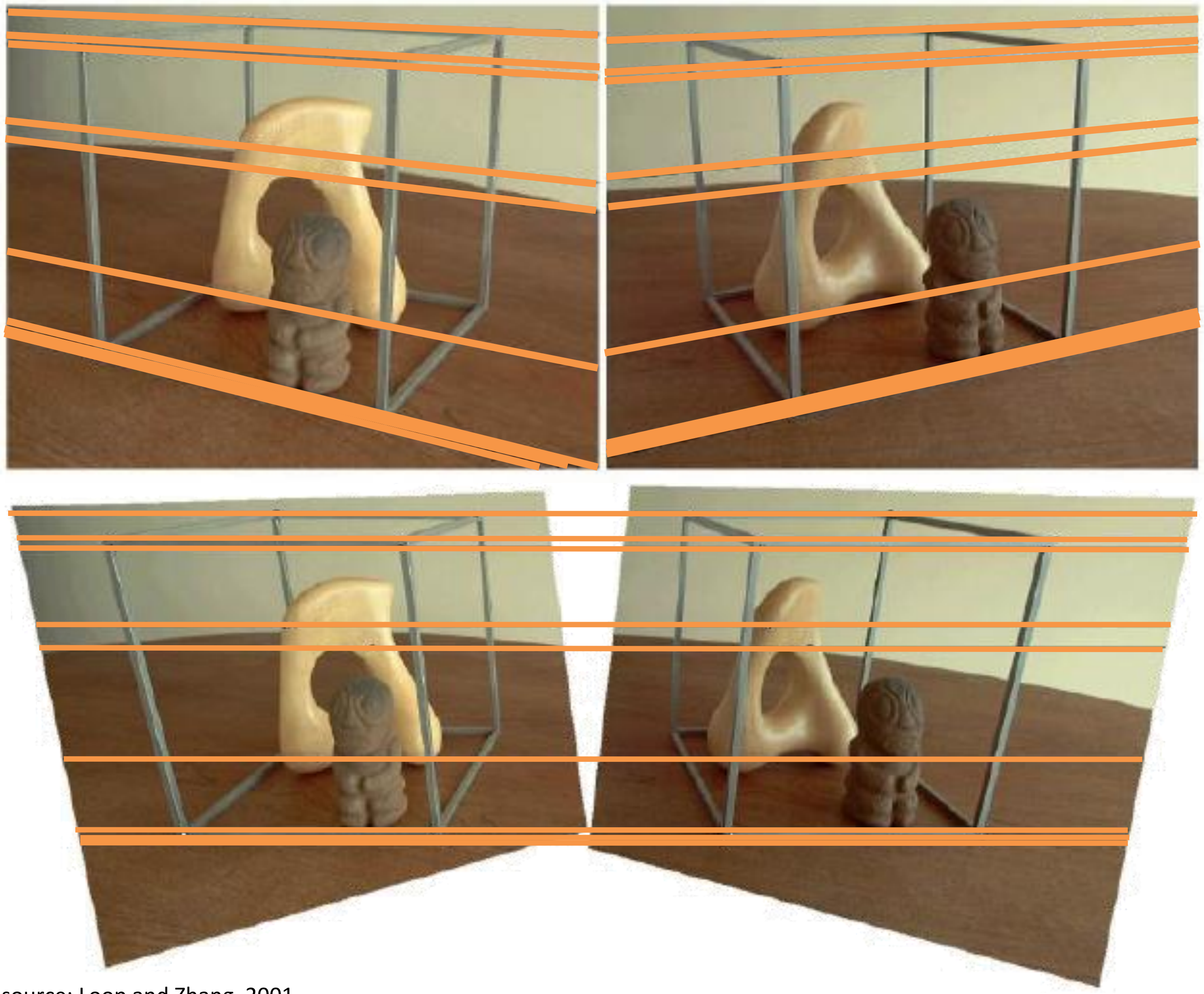


Image source: Scharstein et al., Middlebury stereo benchmark

Recap: Stereo Rectification Example



rectified

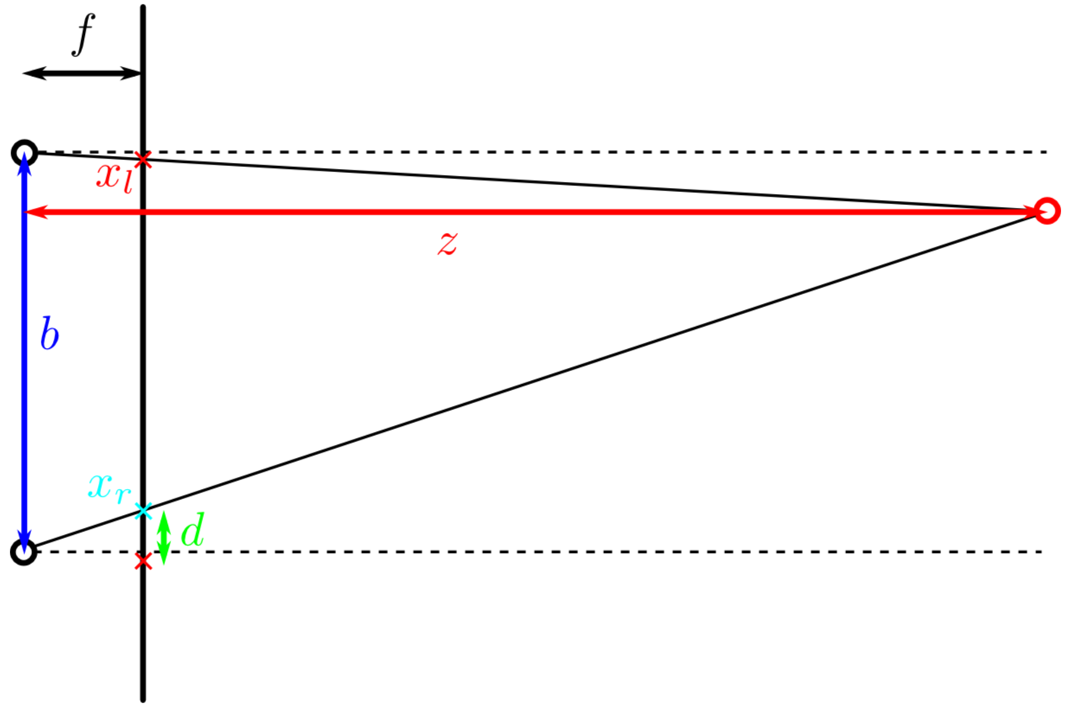
Image source: Loop and Zhang, 2001

Recap: Relation of Disparity and Depth

Similar triangles:

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

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



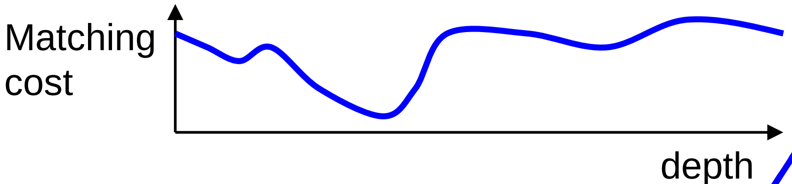
- Disparity is inverse proportional to depth:
  - The larger the depth, the smaller the disparity
- Disparity is proportional to the baseline:
  - The larger the baseline, the larger the disparity
  - Larger baseline means also higher depth accuracy

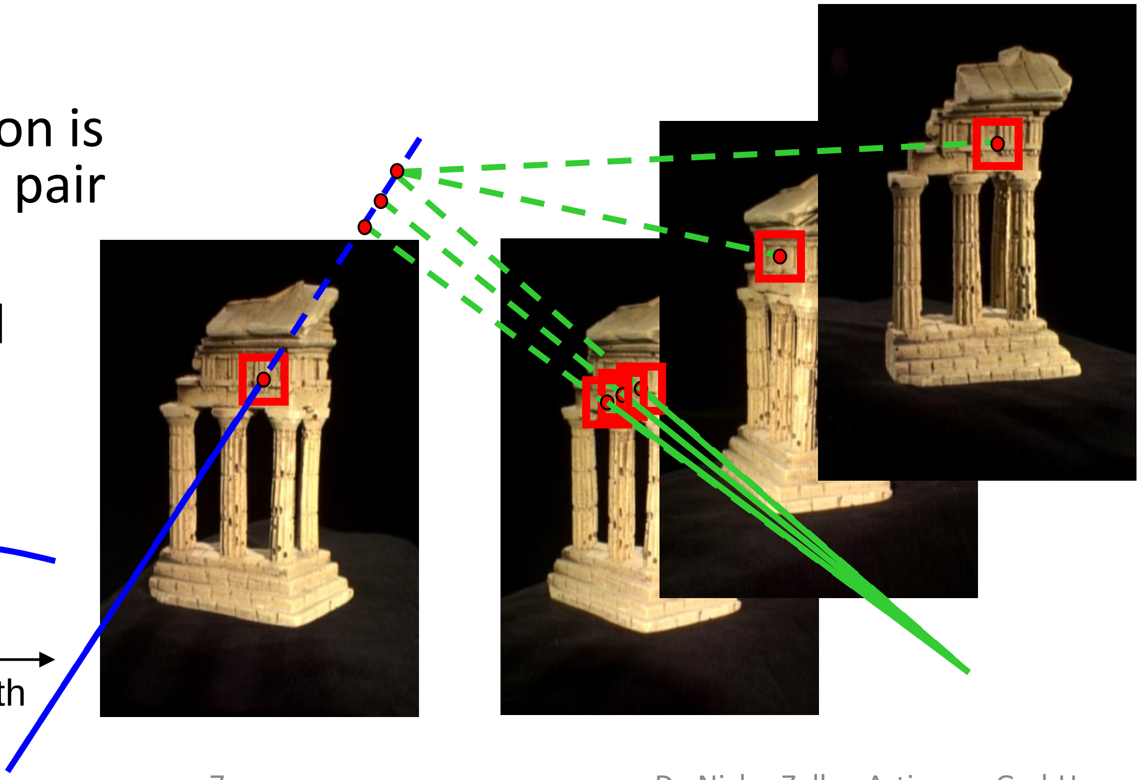
Recap: Block Matching Algorithm

- Input: Two images, intrinsics camera calibration, relative pose
- Output: Disparity image
- Algorithm:
  - Rectify images
  - For each pixel in left image:
    - Compute matching cost along epipolar line using patch comparison
    - Determine minimum in matching cost
      - with sub-pixel accuracy, e.g. using linear interpolation
  - Filter outliers

Recap: Dense Depth from Multiple Views

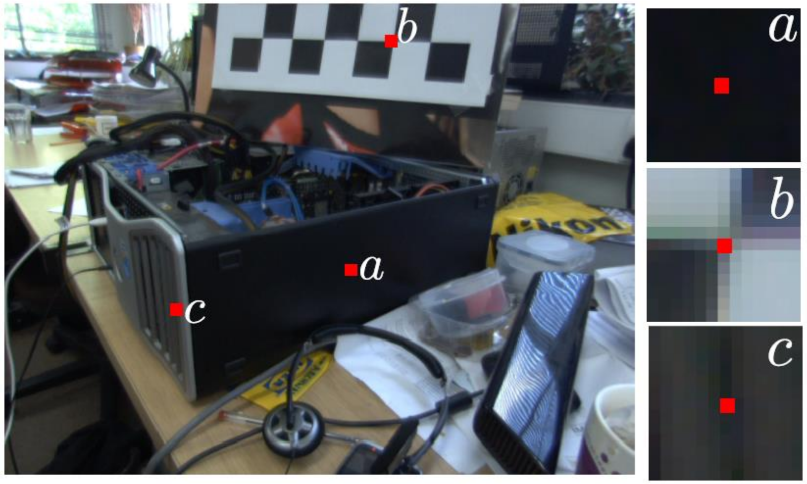
- Straightforward approach: extend two-view matching cost to sum over matching costs of an image towards multiple images
- In general for multiple views images cannot be rectified anymore
- Disparity to depth relation is different for each image pair
- Matching cost is defined as a function of depth (or inv. depth)

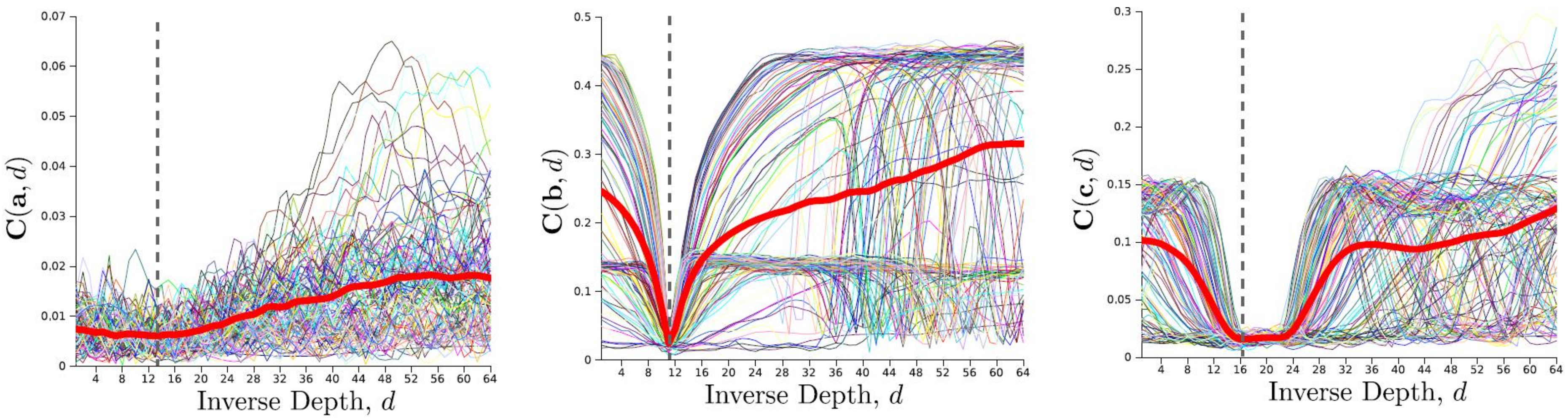




Slide adapted from R. Szeliski

Recap: Multi-View Correspondence Measure



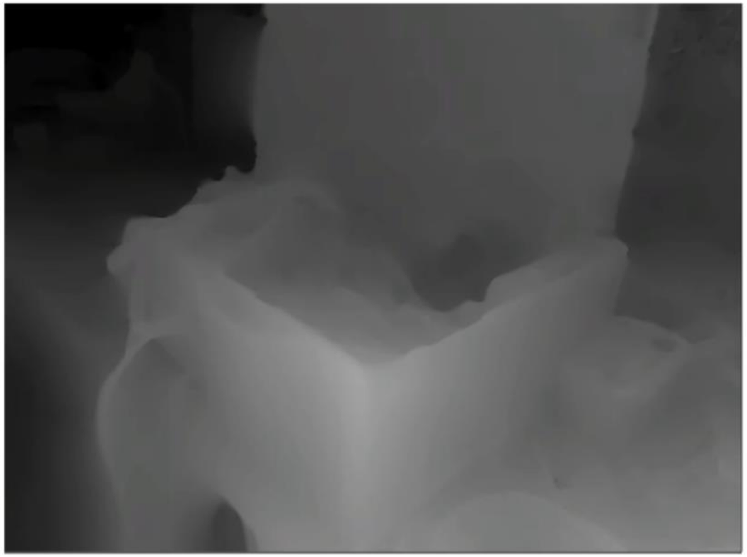


Images: R. Newcombe, 2014

Recap: Effect of Regularization

Data term: cost volume over L1-norm on photometric residuals

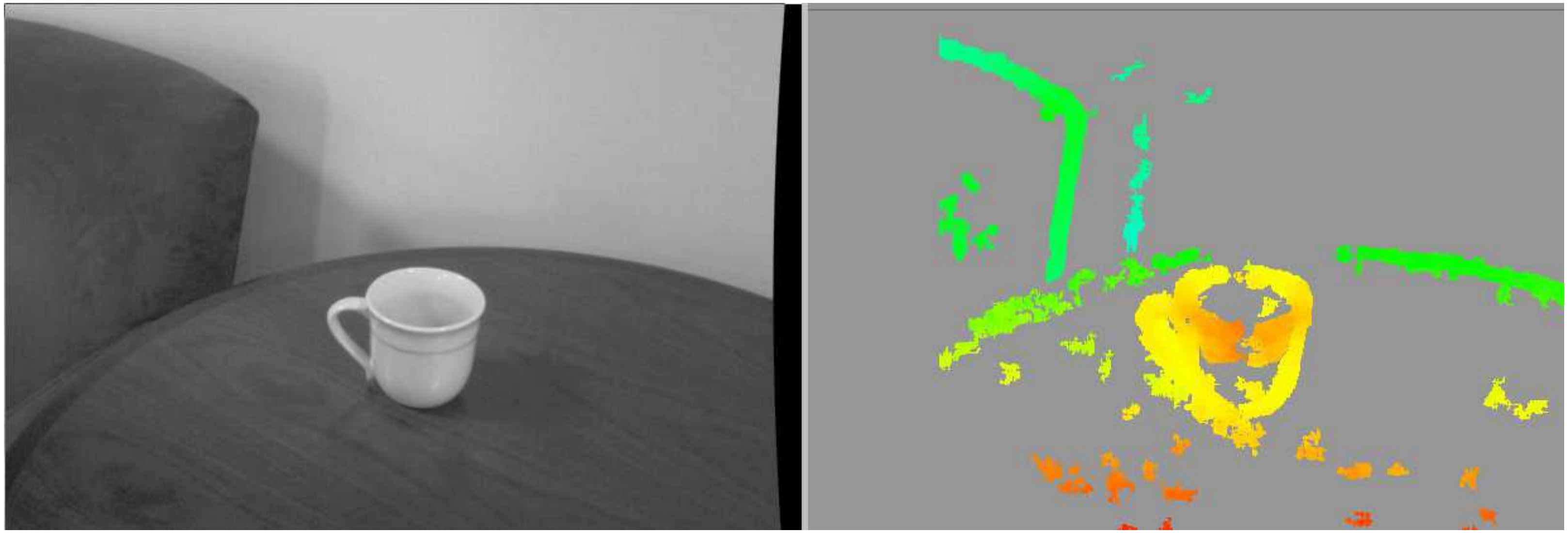
Regularizer: Huber-norm on inverse depth gradient

Images: R. Newcombe et al., 2011

Active Depth Sensing

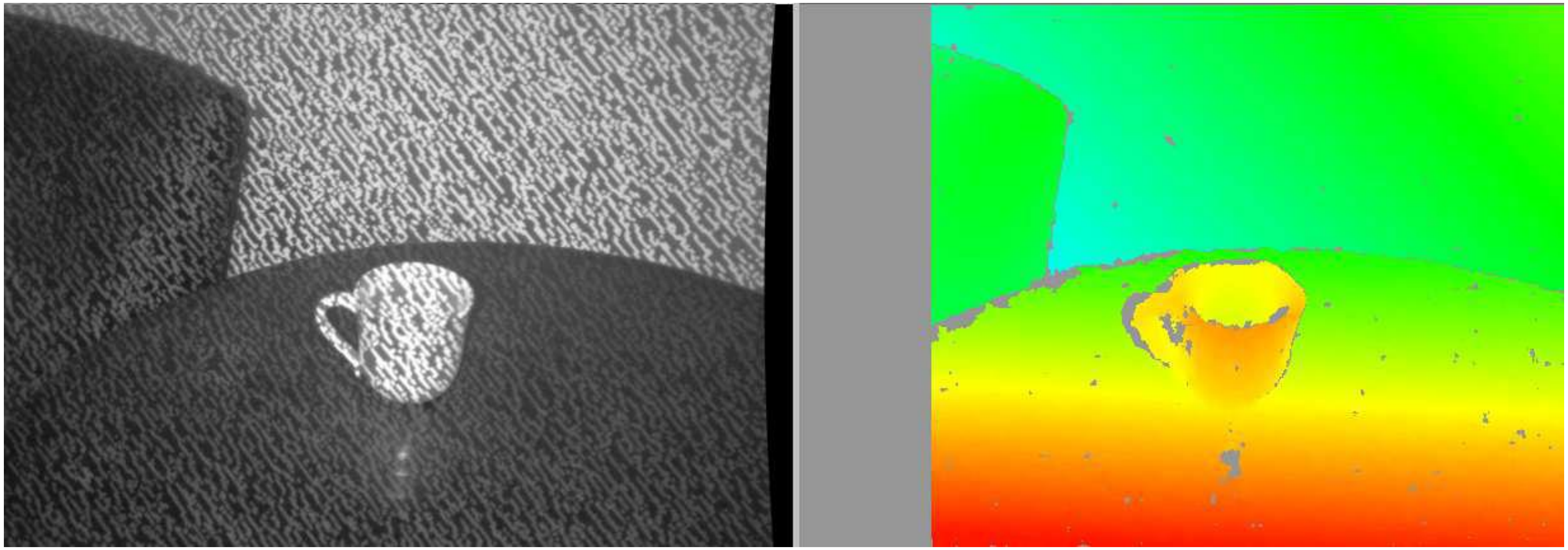
- What can we do about textureless scenes?



Images: J. Sturm

Active Depth Sensing

- Idea: Project light/texture



Images: J. Sturm

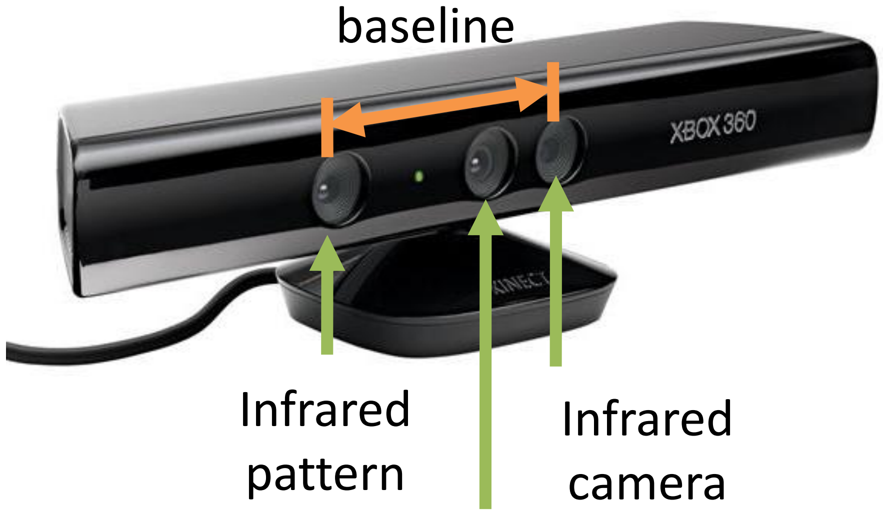
Depth Cameras

Time-of-Flight

Structured Light

Structured Light Measurement Principle

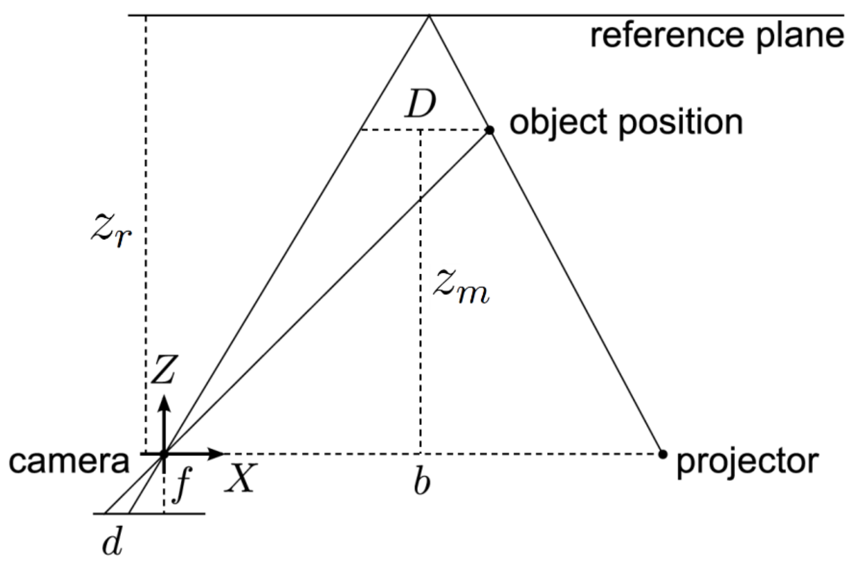
- Project speckle pattern using infrared laser and diffraction element
- Measure infrared speckles using infrared camera
- Measure corresponding RGB image using color camera



Slide adapted from J. Sturm

Structured Light Measurement Principle

- Use known baseline and reference pattern for depth measurement

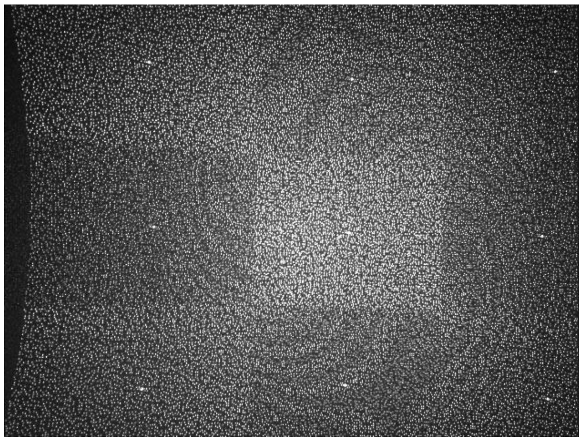


$$\frac{d}{f} = \frac{D}{z_m}$$

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

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

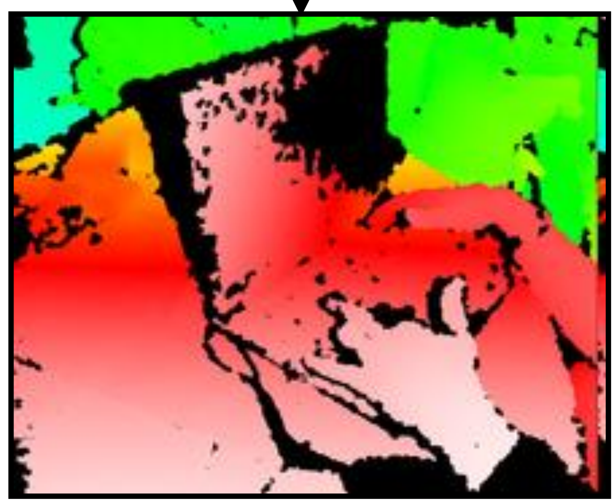
Structured Light Measurement Principle



IR reference pattern

Block
matching
(9x9)

IR pattern
in actual scene



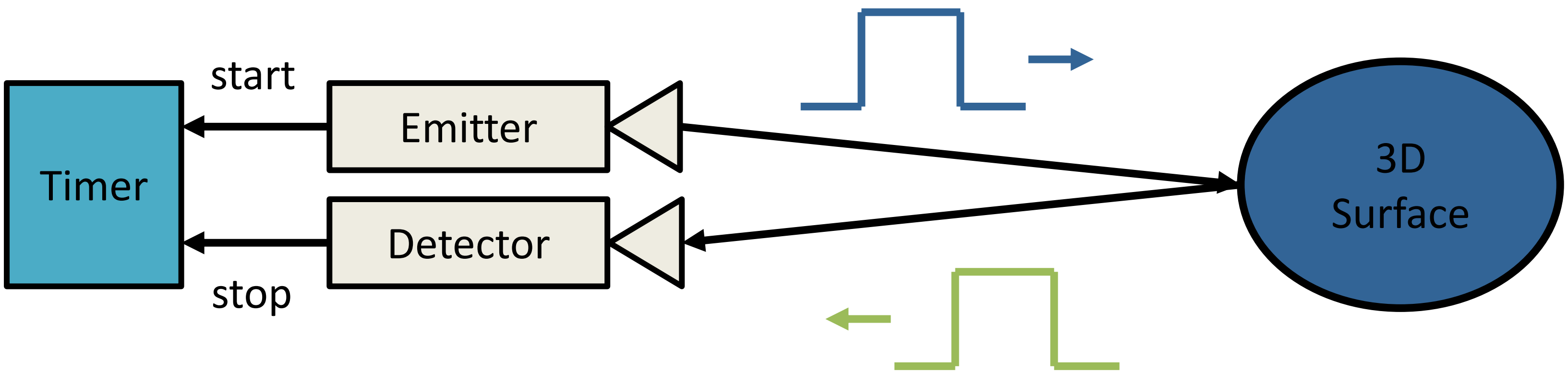
Depth image

Slide adapted from J. Sturm

Time-of-Flight Measurement Principle

- Idea: emit timed IR pulse and measure its time of return

- Difficult to create pulses and measure time precisely

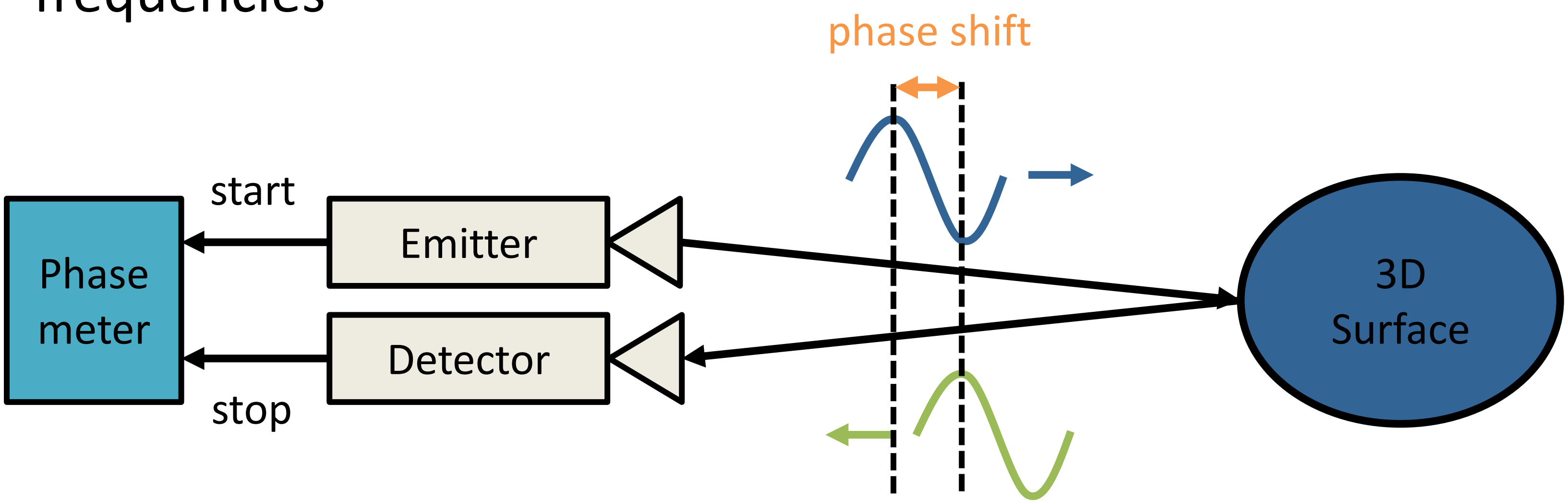


Slide adapted from N. Navab

Time-of-Flight Measurement Principle

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

- Signal periodicity creates phase ambiguities: use multiple frequencies



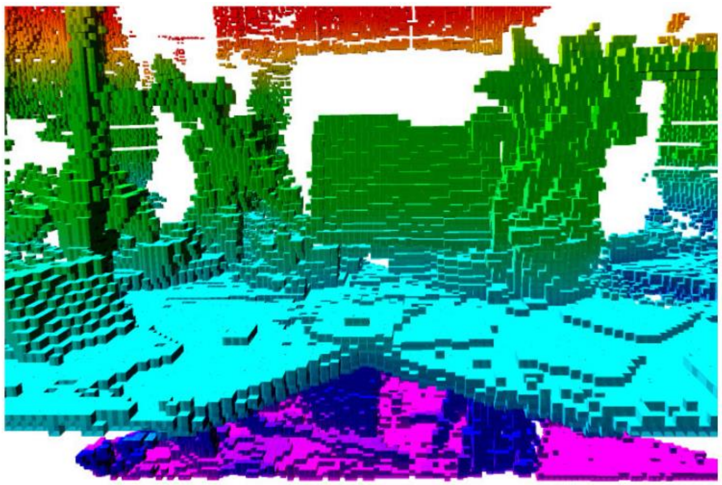
Slide adapted from N. Navab

Active vs. Passive Sensors

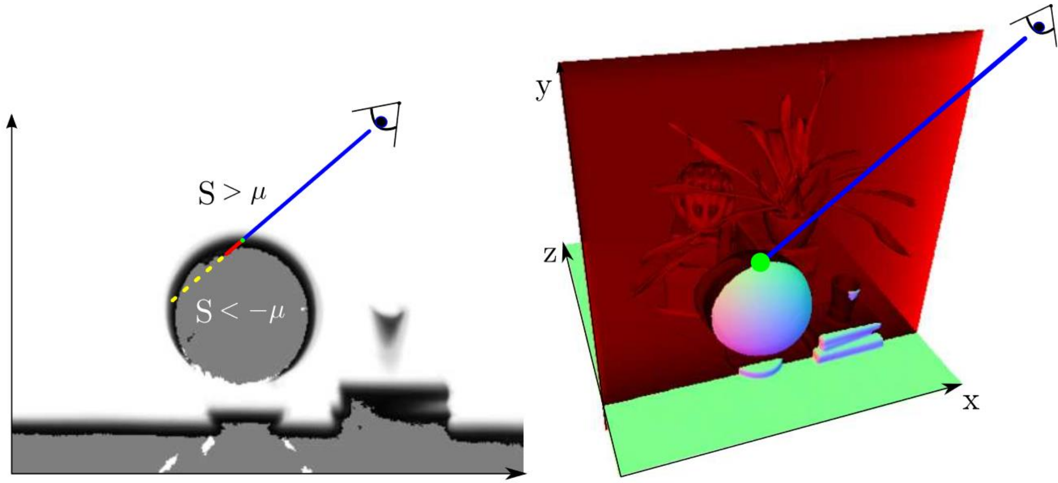
- Active Sensors
  - Surfaces do not need to be textured
  - Bring their own light, also work in low-light scenarios
  - But: Diffuse IR sunlight typically overrides emitted light
  - Difficulties for IR-absorbing or reflective materials

- Passive Sensors (e.g RGB-only)
  - Do not rely on measuring emitted light
  - Are not limited by the resolution of the projection pattern or ToF measurement principle
    - Distance
    - Multi-path noise (ToF)

Dense 3D Map Representations



Volumetric Occupancy Maps



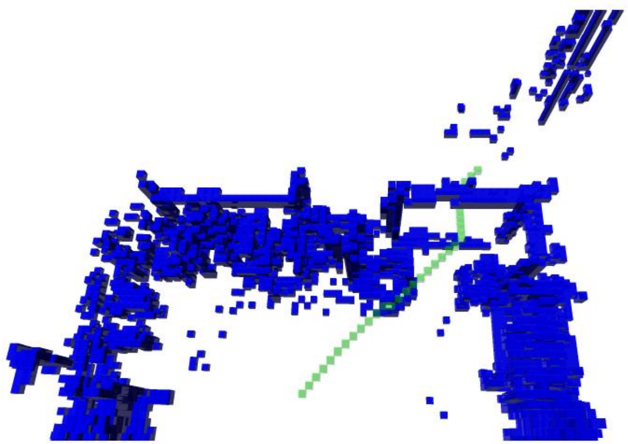
Volumetric Signed Distance Functions

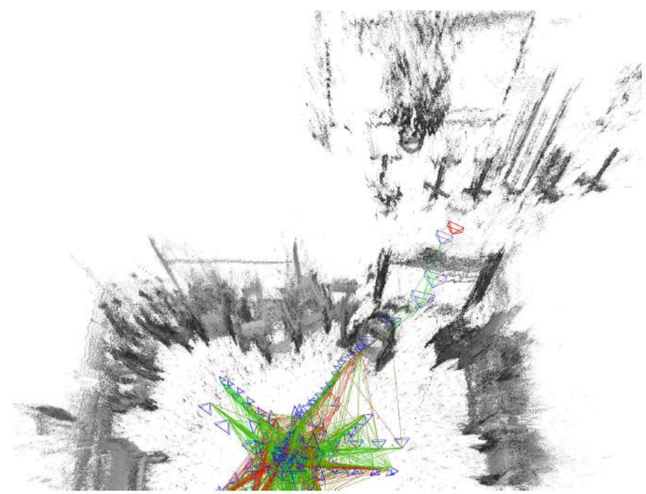
Images: Wurm et al., 2010; Newcombe et al., 2011

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 based on optimized camera poses
  - Offline integration after sequence recording
  - Online integration requires map modification when poses change
    - Still an open research topic

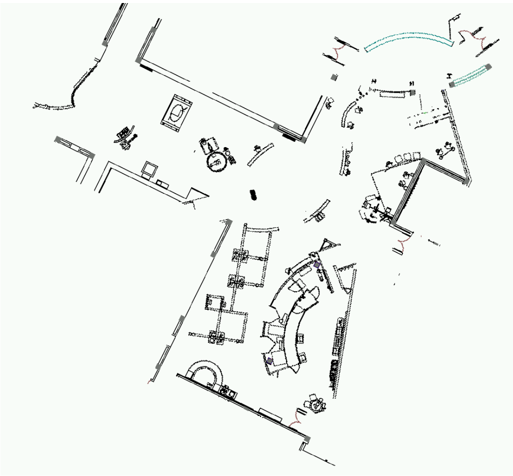
- Most common in robotics are implicit, grid based representations
  - E.g. occupancy grid, signed distance functions (SDF)

- Alternatives can be explicit surface representation
  - e.g. meshes

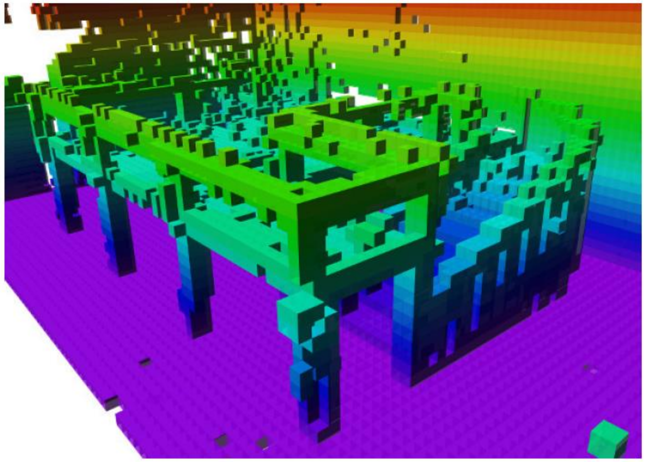
Occupancy Grid Maps

- 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 $m_i \in \{\text{occ}, \text{empty}\}$ 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 ratio simplifies calculations and improves numeric stability

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

Recursive Bayesian Filtering of Occupancy

- Ratio of probabilities

$$\frac{p(m = \text{occ}|y_{1:t})}{p(m = \text{free}|y_{1:t})} = \frac{p(m = \text{occ}|y_{1:t})}{1 - p(m = \text{occ}|y_{1:t})}$$

$$= \frac{\frac{p(m = \text{occ}|y_t)p(y_t)p(m = \text{occ}|y_{1:t-1})}{p(m = \text{occ})p(y_t|y_{1:t-1})}}{\frac{p(m = \text{free}|y_t)p(y_t)p(m = \text{free}|y_{1:t-1})}{p(m = \text{free})p(y_t|y_{1:t-1})}}$$

$$= \frac{p(m = \text{occ}|y_t)}{p(m = \text{free}|y_t)} \cdot \frac{p(m = \text{free})}{p(m = \text{occ})} \cdot \frac{p(m = \text{occ}|y_{1:t-1})}{p(m = \text{free}|y_{1:t-1})}$$

$$= \frac{p(m = \text{occ}|y_t)}{1 - p(m = \text{occ}|y_t)} \cdot \frac{1 - p(m = \text{occ})}{p(m = \text{occ})} \cdot \frac{p(m = \text{occ}|y_{1:t-1})}{1 - p(m = \text{occ}|y_{1:t-1})}$$

Recursive Bayesian Filtering of Occupancy

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

$$= \underbrace{\log\left(\frac{p(m = \text{occ} \mid y_t)}{1 - p(m = \text{occ} \mid y_t)}\right)}_{\text{term involving current measurement}} - \underbrace{\log\left(\frac{p(m = \text{occ})}{1 - p(m = \text{occ})}\right)}_{\text{prior}} + \underbrace{l(m = \text{occ} \mid y_{1:t-1})}_{\text{previous estimate}}$$

Inverse Sensor Model

- Typical inverse sensor model for range sensors

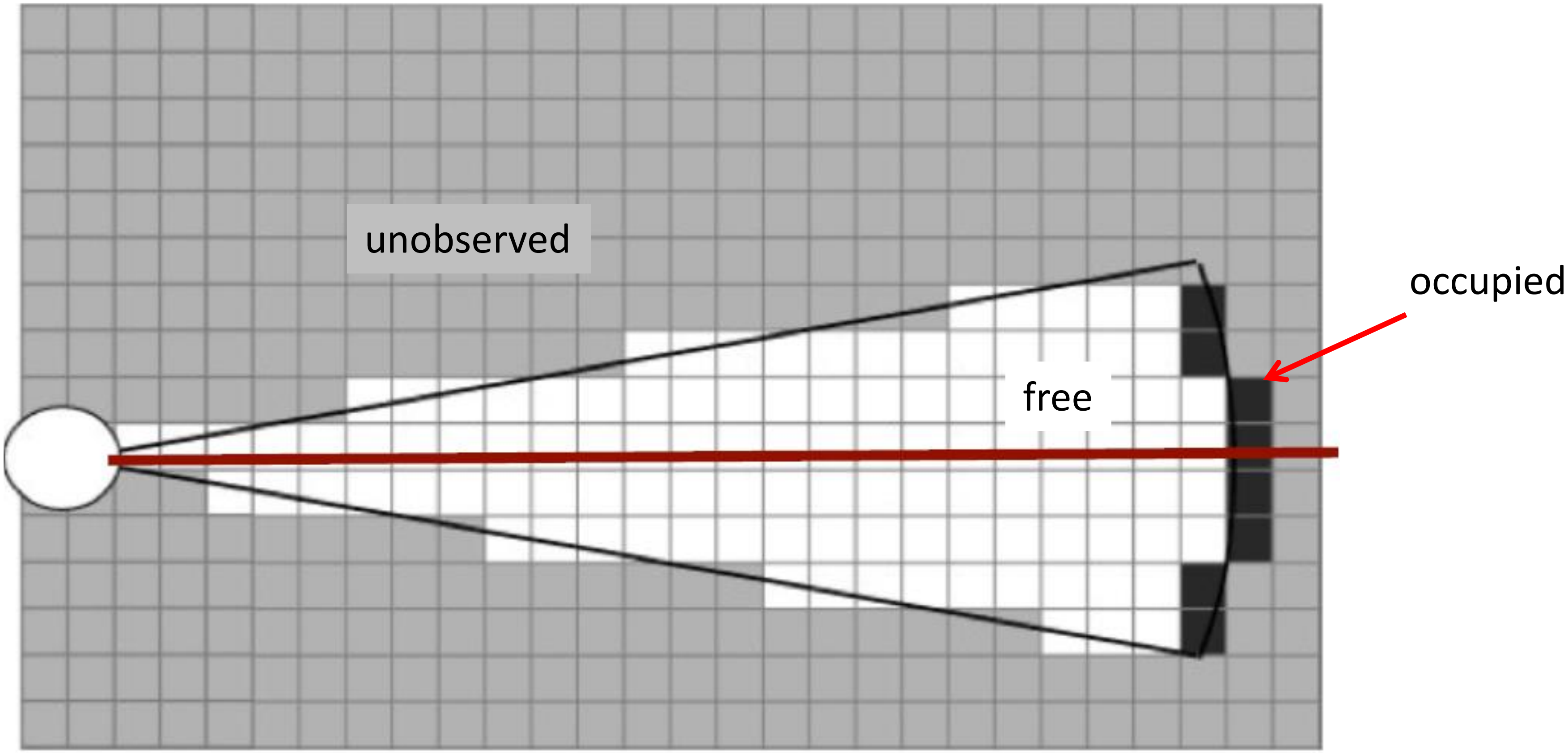


Image: C. Stachniss

Inverse Sensor Model

- Typical inverse sensor model for range sensors

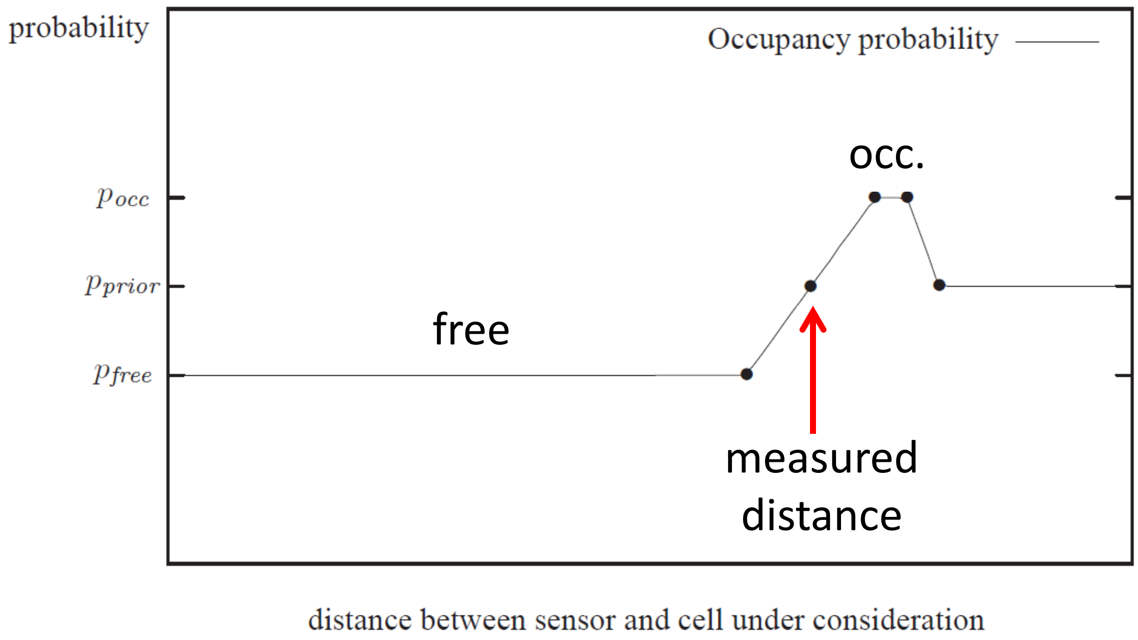


Image: C. Stachniss, 2006

Example: 2D Mapping with Sonar Sensors

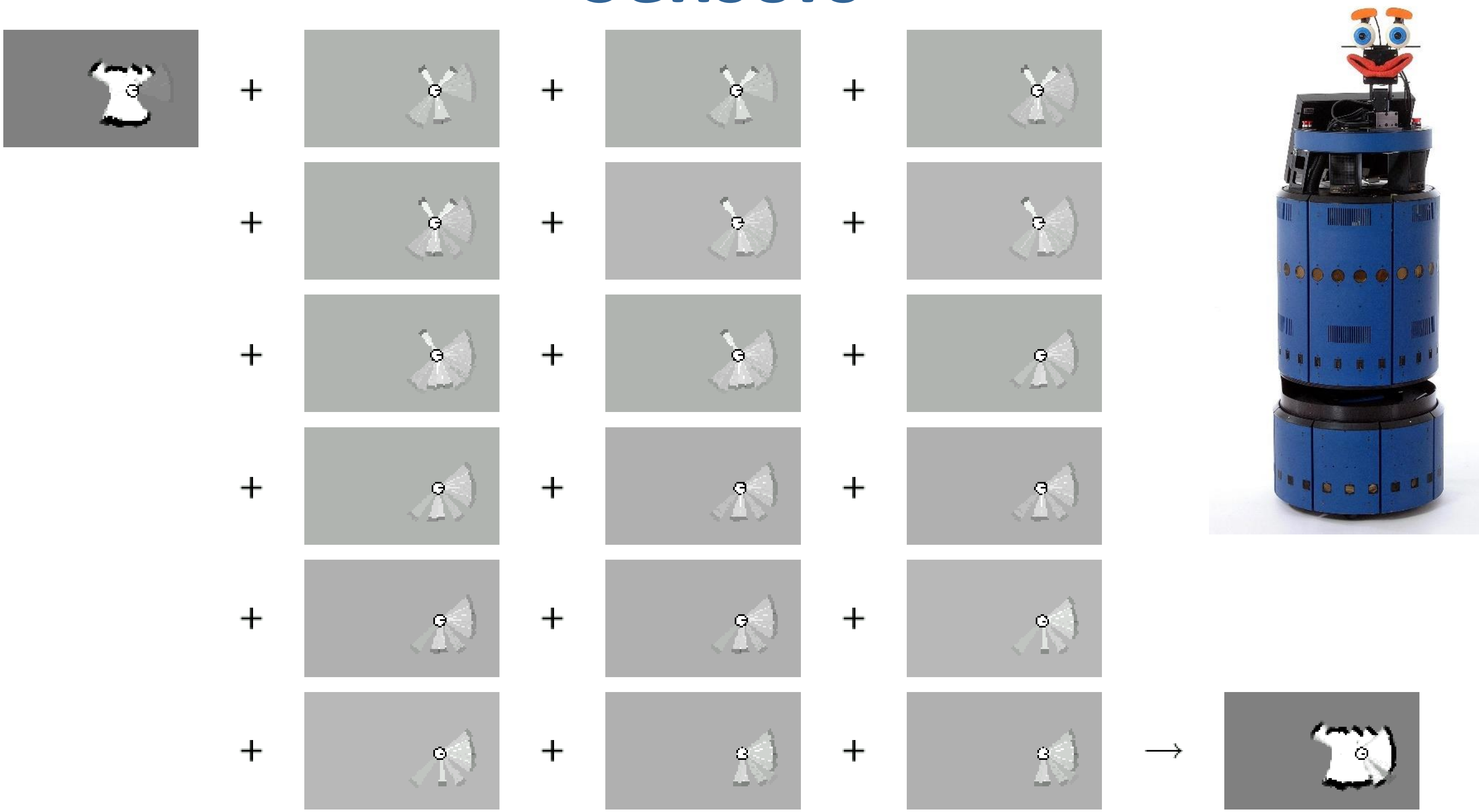


Image: Thrun et al., 2005

Example: 2D Mapping with Sonar Sensors

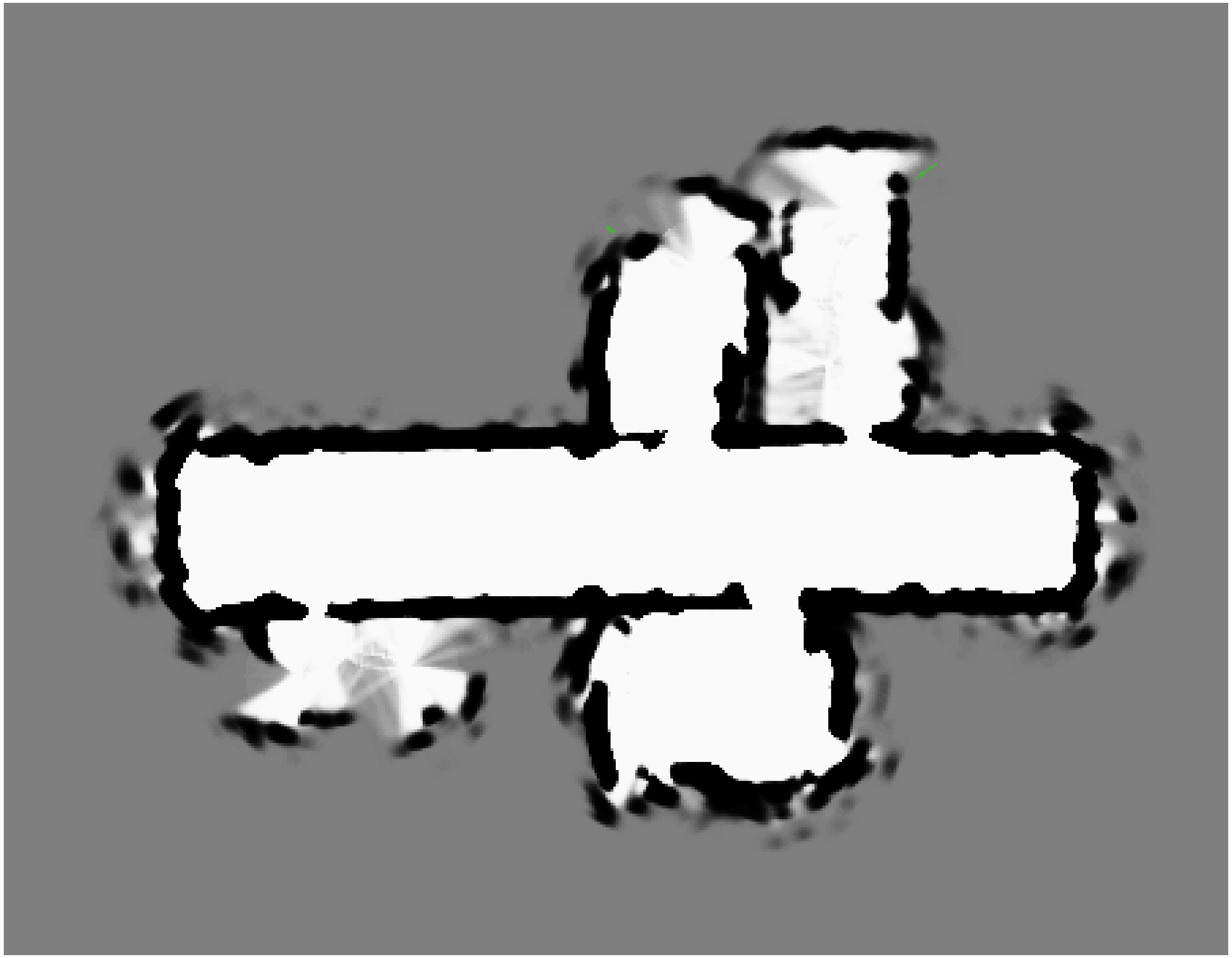
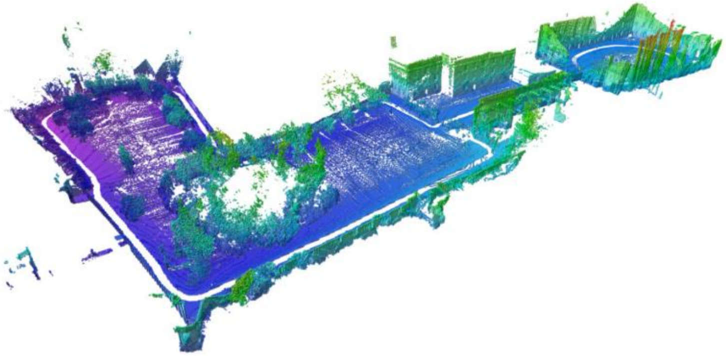


Image: Thrun et al., 2005

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)

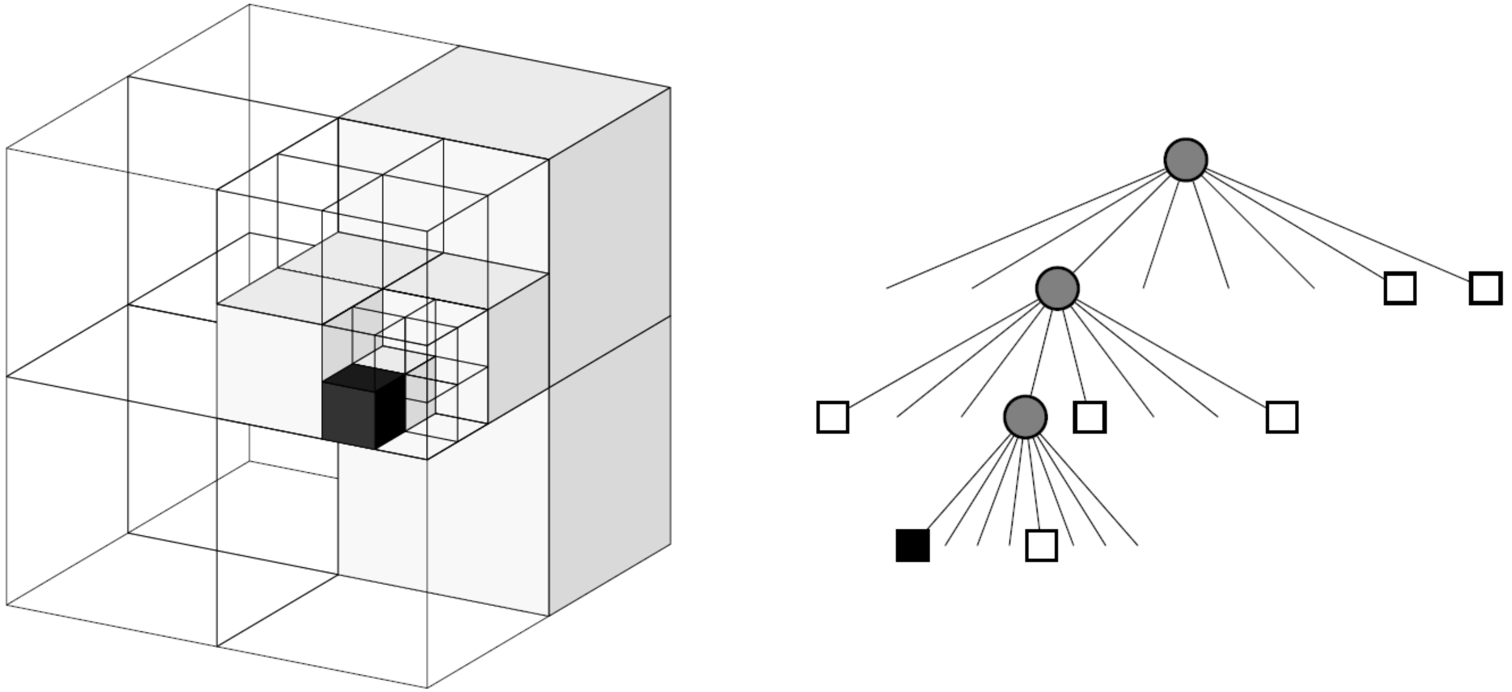


- Memory consumption quickly gets huge!
- Likely large volumes will be empty! (unobserved)
- What can we do?

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

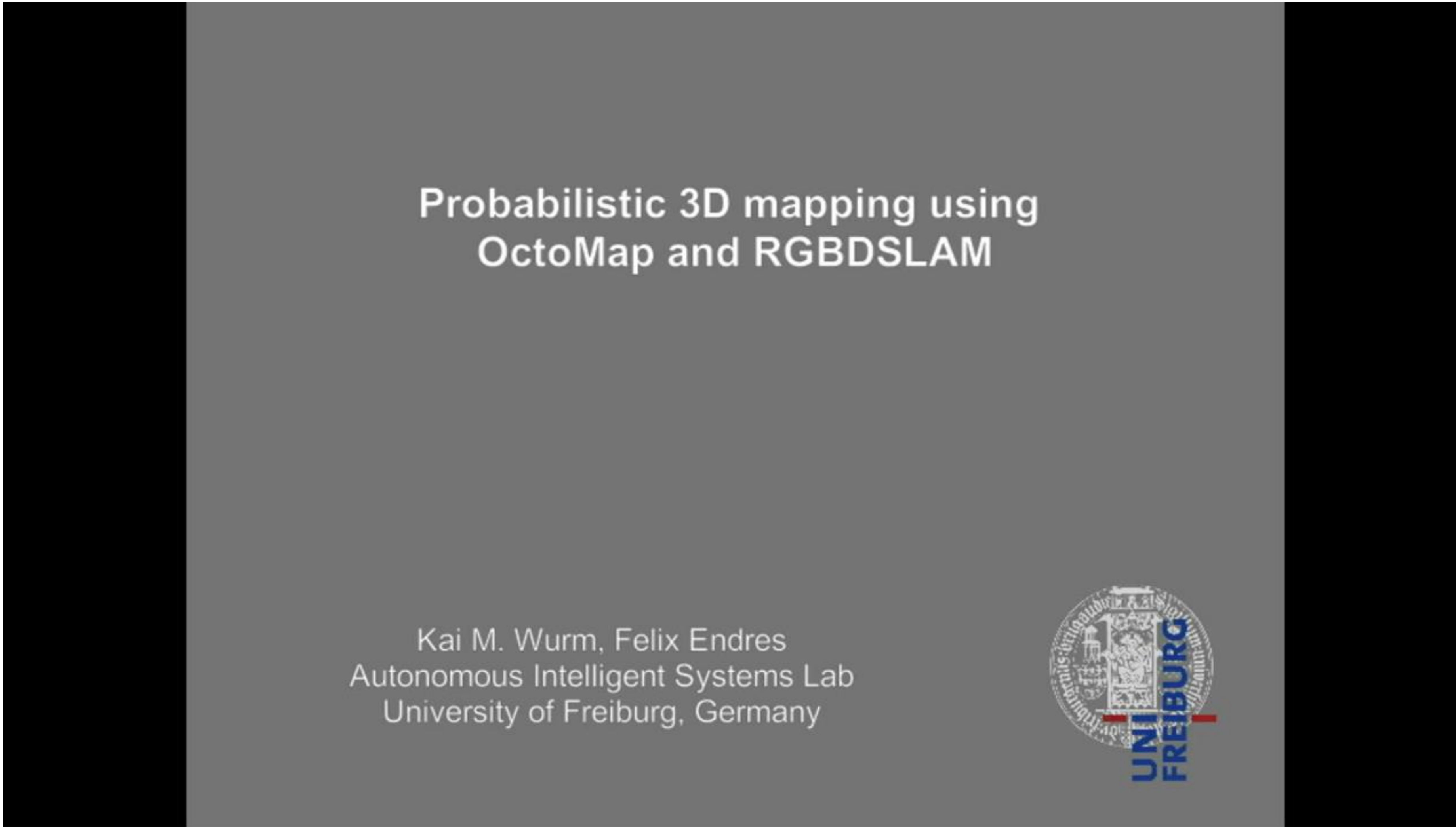
3D Occupancy Maps in Octrees

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



Images: Wurm et al., 2010

Example: OctoMap & RGB-D SLAM

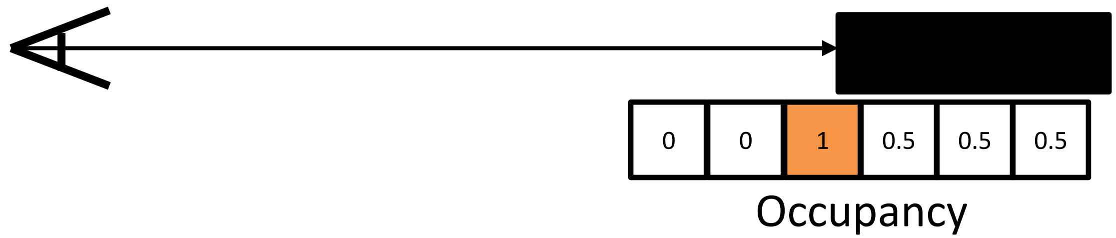


Endres et al., 3D Mapping with RGB-D Cameras, TRO, 2014
Hornung et al., OctoMap, Autonomous Robots, 2013

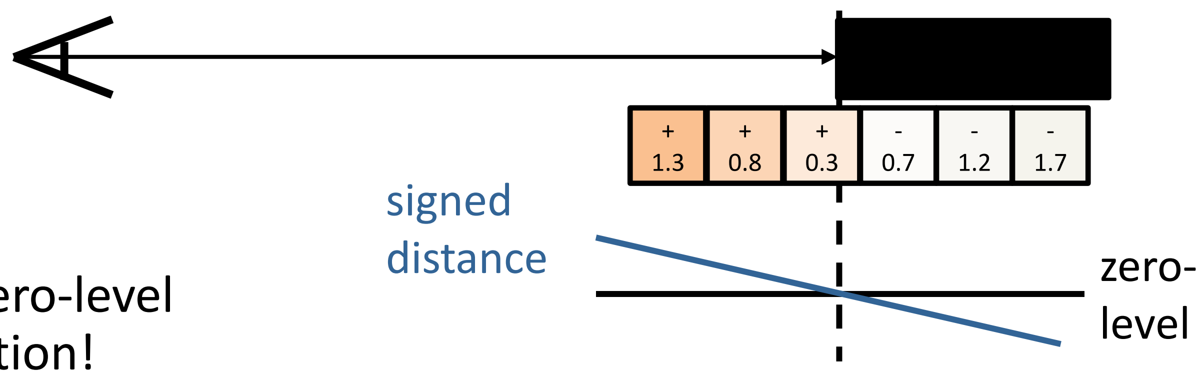
https://www.youtube.com/watch?v=9f32FmbtHCs

Signed Distance Function (SDF)

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

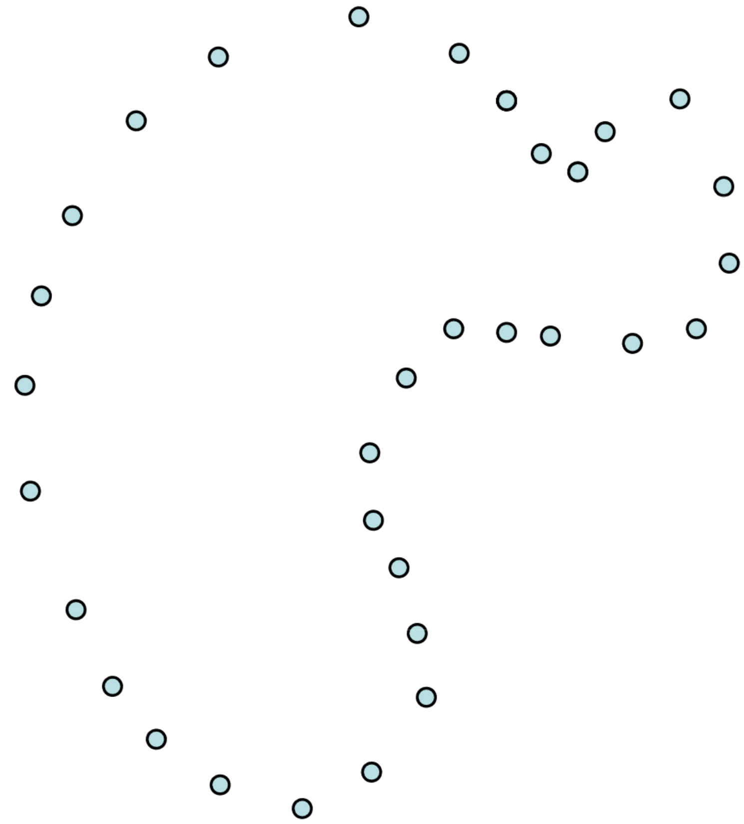


- Idea:
  - Instead of occupancy, store the distance from the surface in the grid cells
  - Represent inside/outside the object using the sign



  - We can find the zero-level through interpolation!

SDF Approach

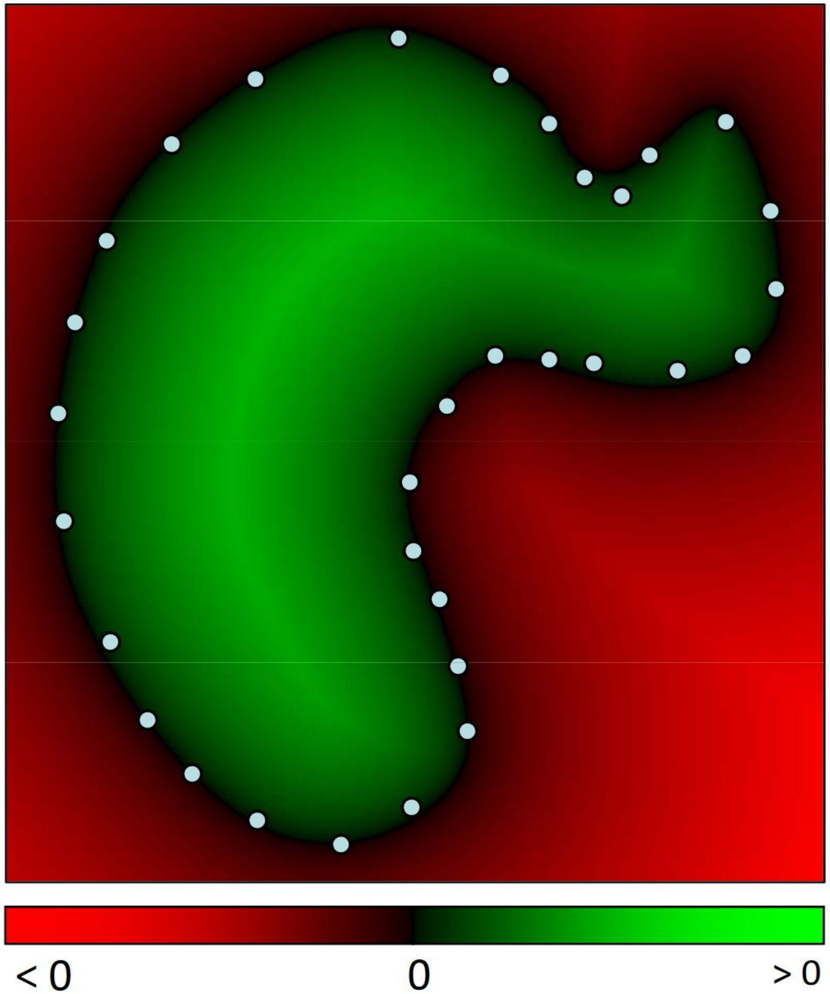


SDF Approach

- Define a function

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

  with value < 0 outside and value > 0 inside object



SDF Approach

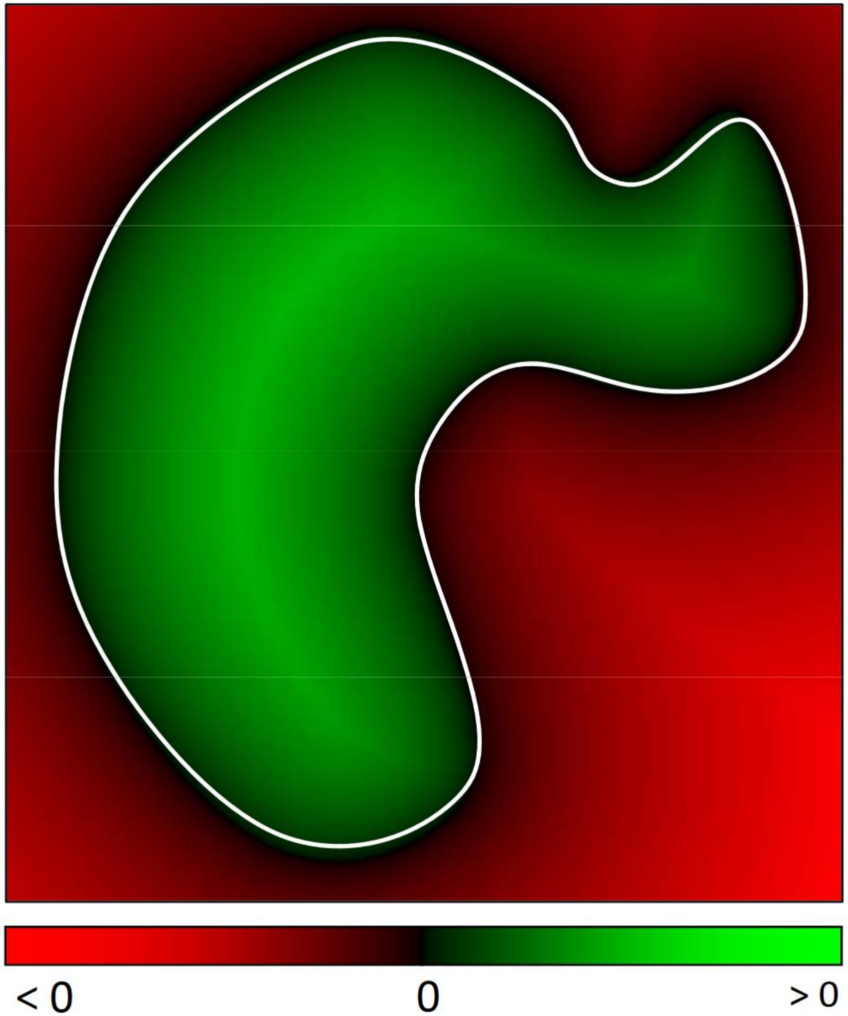
- Define a function

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

with value < 0 outside and value > 0 inside object

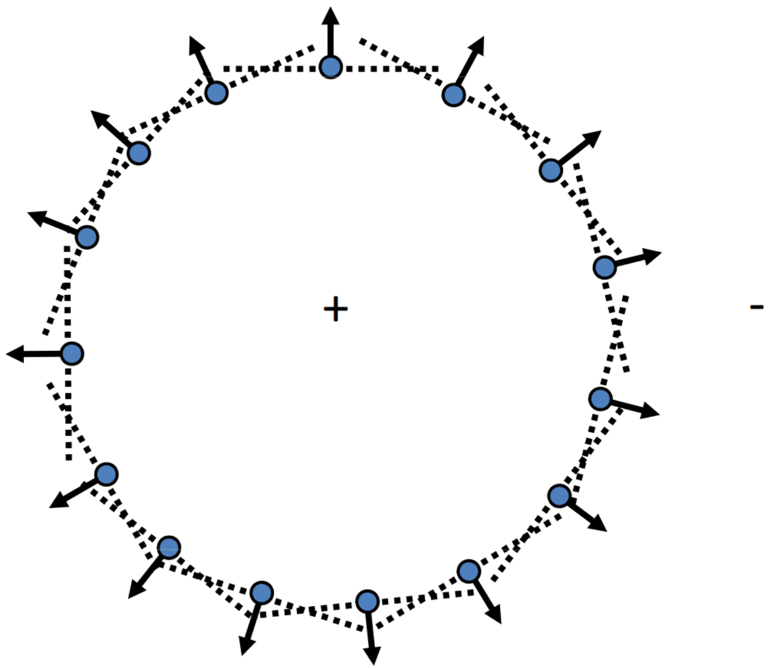
- Extract zero-level set

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

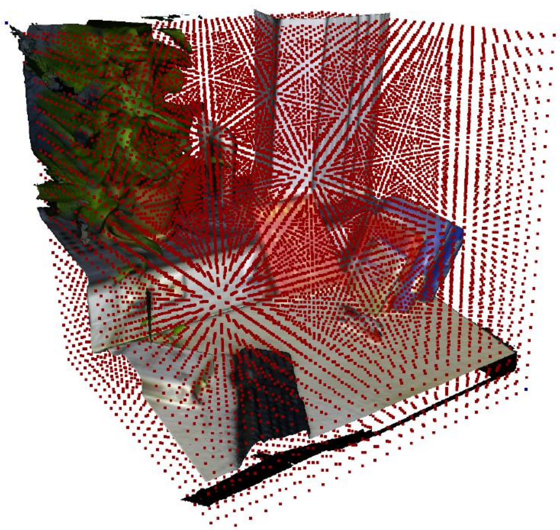
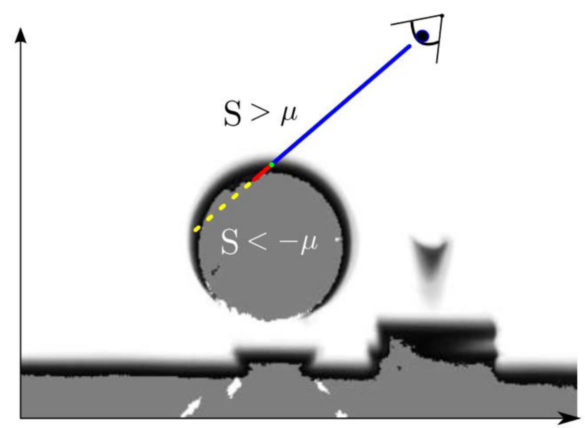


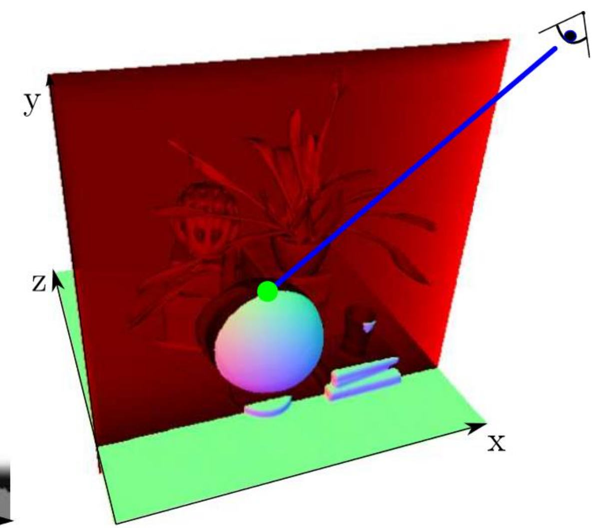
SDF from Point Sets

- 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



SDFs for 3D Map Representation

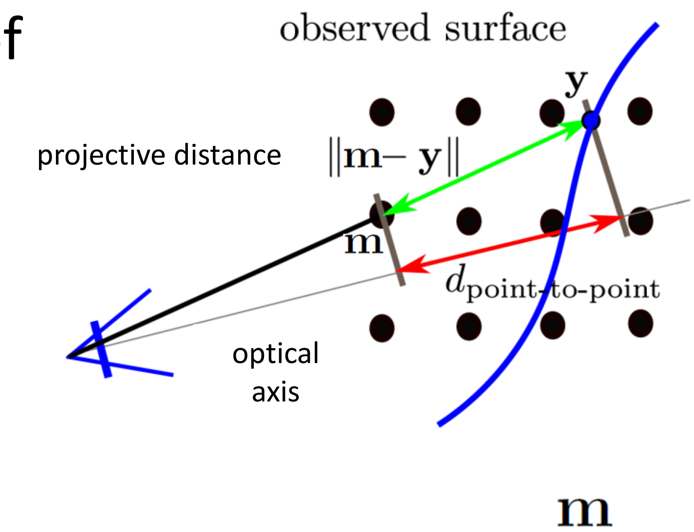





Images: Bylow et al., 2013; Newcombe et al., 2011

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

Further Insights

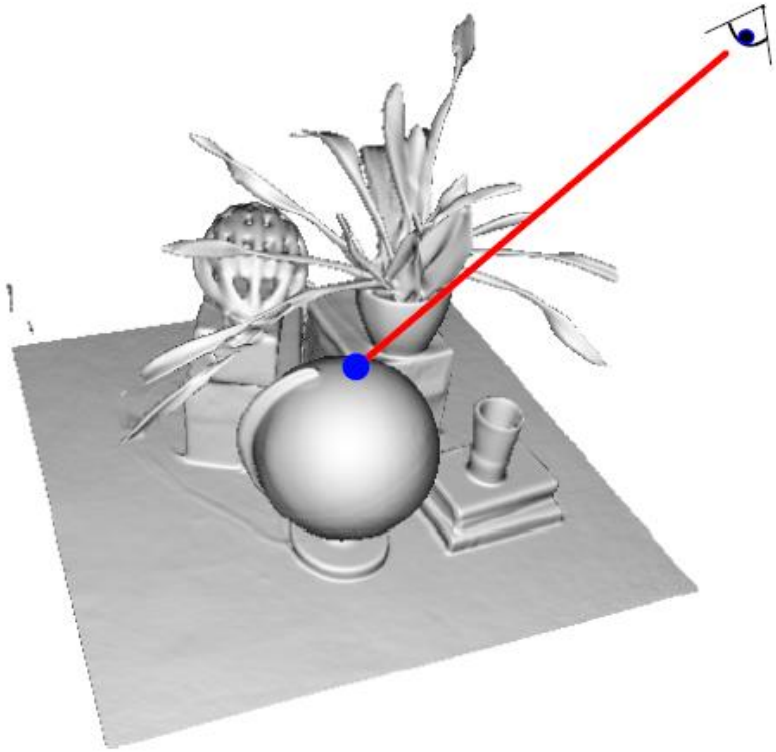
- Typically, noise cancels out over multiple measurements

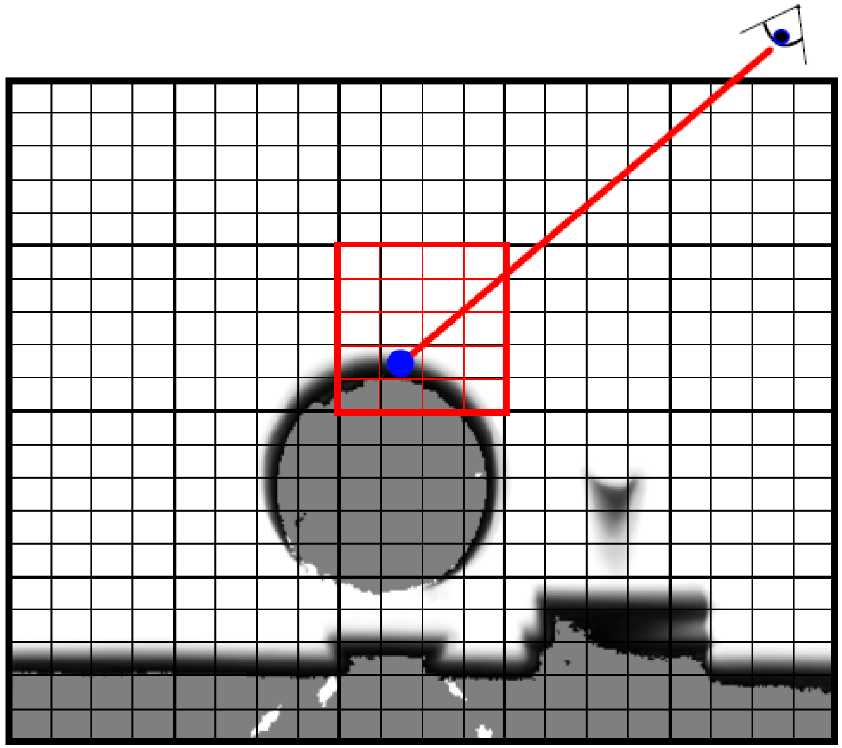
- Truncated signed distance functions (TSDF)
  - In practice , one often limits the integration range to a narrow band around the zero level-set to increase efficiency and allow for thin objects. The signed distance function is then called truncated SDF (TSDF).

- 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

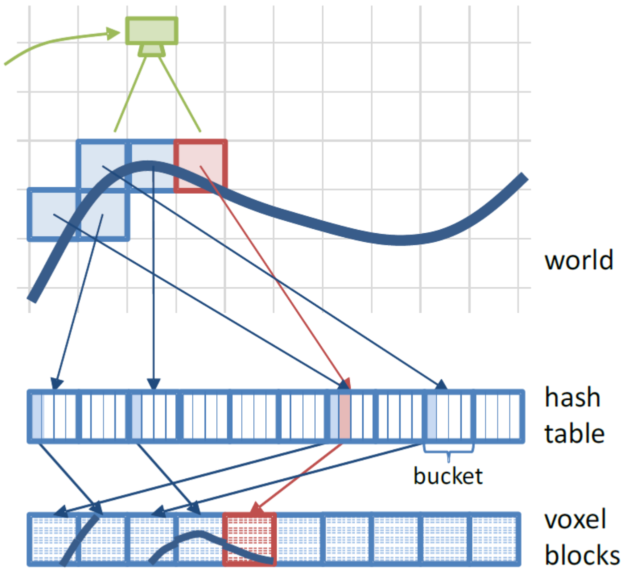
- For each pixel in the novel view, cast a ray to find the first zero-crossing





Voxel Hashing 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 high resolution?
- Only allocate voxels close to the updated narrow band along the surface
- Index voxels through hashing



Images: Niessner et al., 2013

Lessons Learned Today

- Dense 3D map representations useful for augmented / virtual reality and robot navigation and exploration

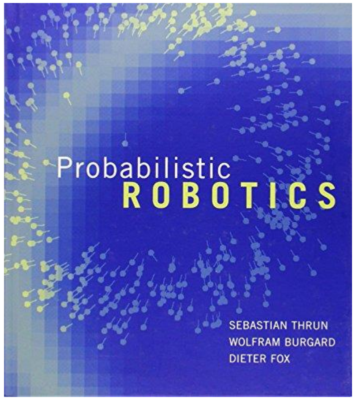
- 3D occupancy grid maps
  - Implicit volumetric surface representation: occupancy probability in grid cells
  - Recursive Bayesian estimation using log-odds filter and inverse sensor model

- 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

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!

Slides Information

- These slides have been initially created by Jörg Stückler as part of the lecture "Robotic 3D Vision" in winter term 2017/18 at Technical University of Munich.
- The slides have been revised by myself (Niclas Zeller) for the same lecture held in winter term 2020/21
- Acknowledgement of all people that contributed images or video material has been tried (please kindly inform me if such an acknowledgement is missing so it can be added).