Microsoft Kinect

May 30, 2012

Young Min Kim

Geometric Computing Group

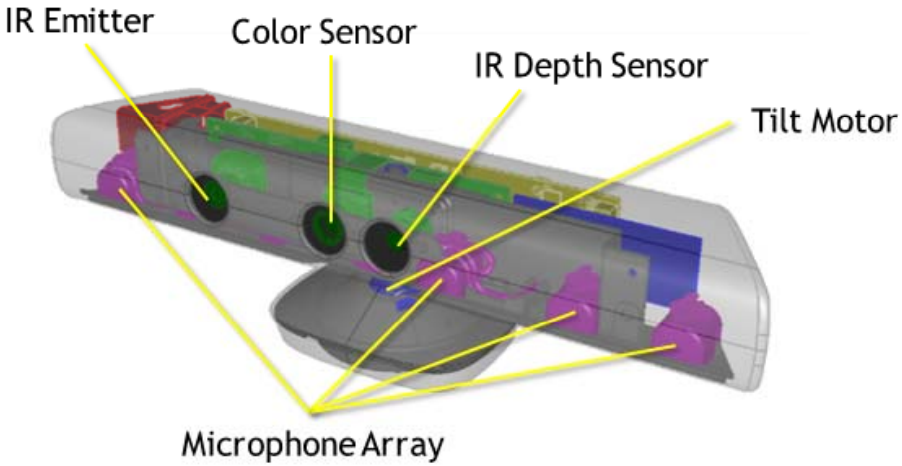
Kinect Effect

- Kinect Effect
- Everybody can have an access to 3-D data
- Real-time processing

Technology

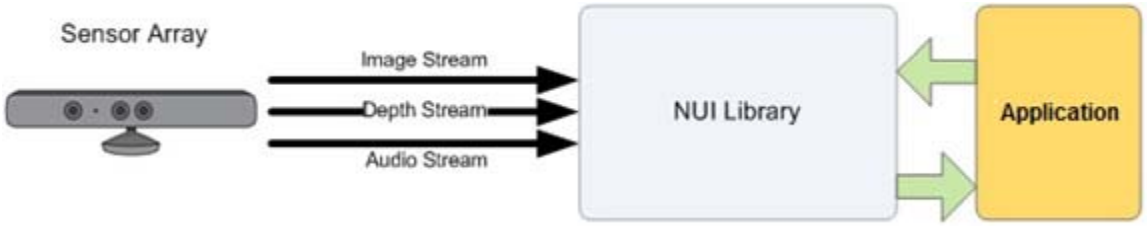
- Motion sensor
- Skeleton tracking
- Facial recognition
- Voice recognition

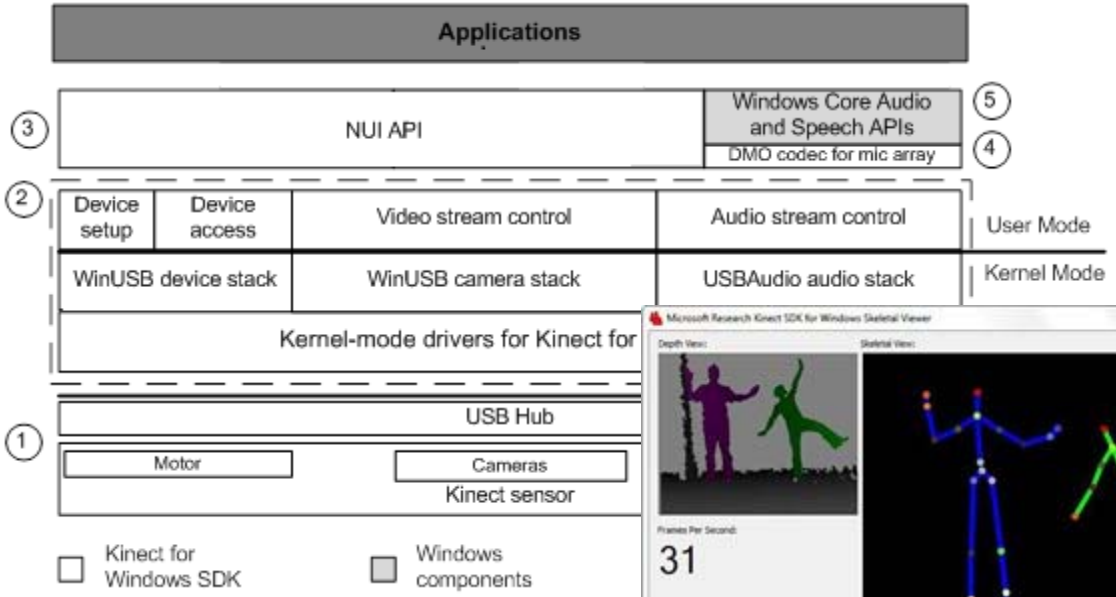
Sensors



Viewing angle	43° vertical by 57° horizontal field of view
Vertical tilt range	±27°
Frame rate (depth and color stream)	30 frames per second (FPS)
Audio format	16-kHz, 24-bit mono pulse code modulation (PCM)
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression

API

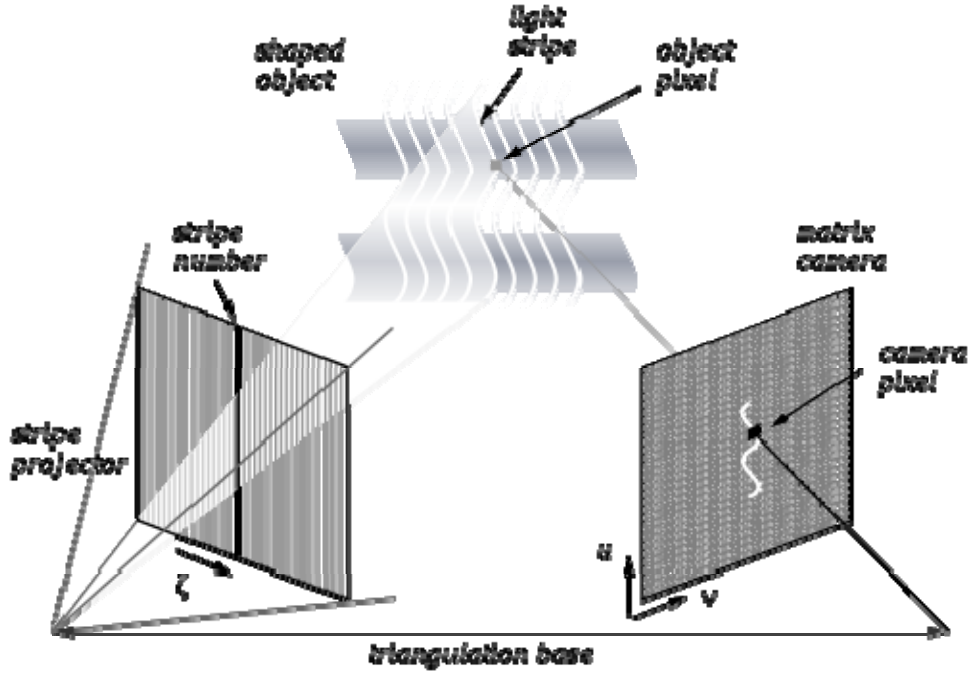

Sensor Array
Image Stream
Depth Stream
Audio Stream
NUI Library
Application


Applications
3
NUI API
Windows Core Audio and Speech APIs
5
DMO codec for mic array
4
2
Device setup
Device access
Video stream control
Audio stream control
User Mode
WinUSB device stack
WinUSB camera stack
USBAudio audio stack
Kernel Mode
Kernel-mode drivers for Kinect for
USB Hub
1
Motor
Cameras
Kinect sensor
Kinect for Windows SDK
Windows components

31

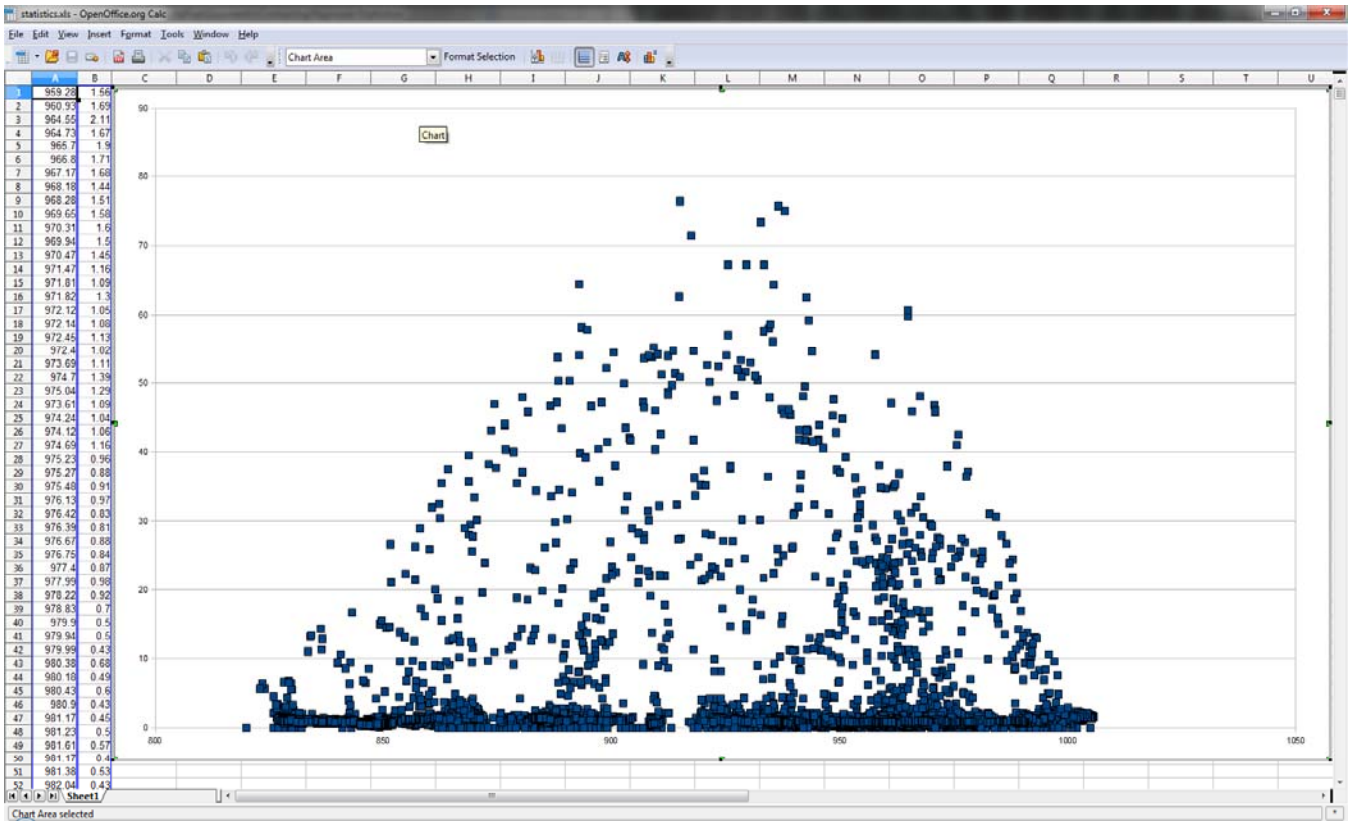
How it works

- How kinect works



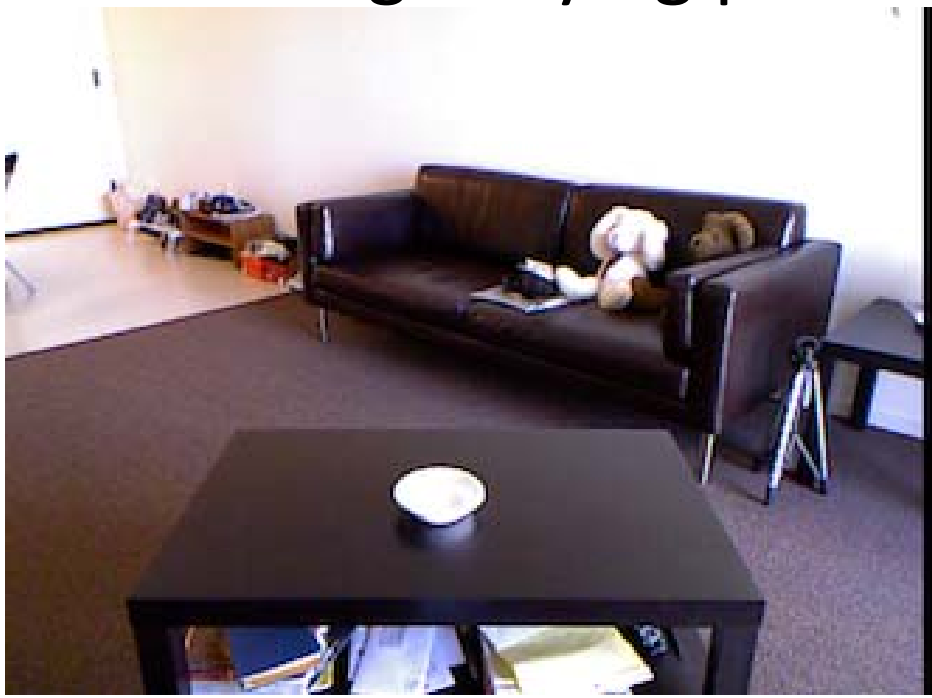
Noise characteristics

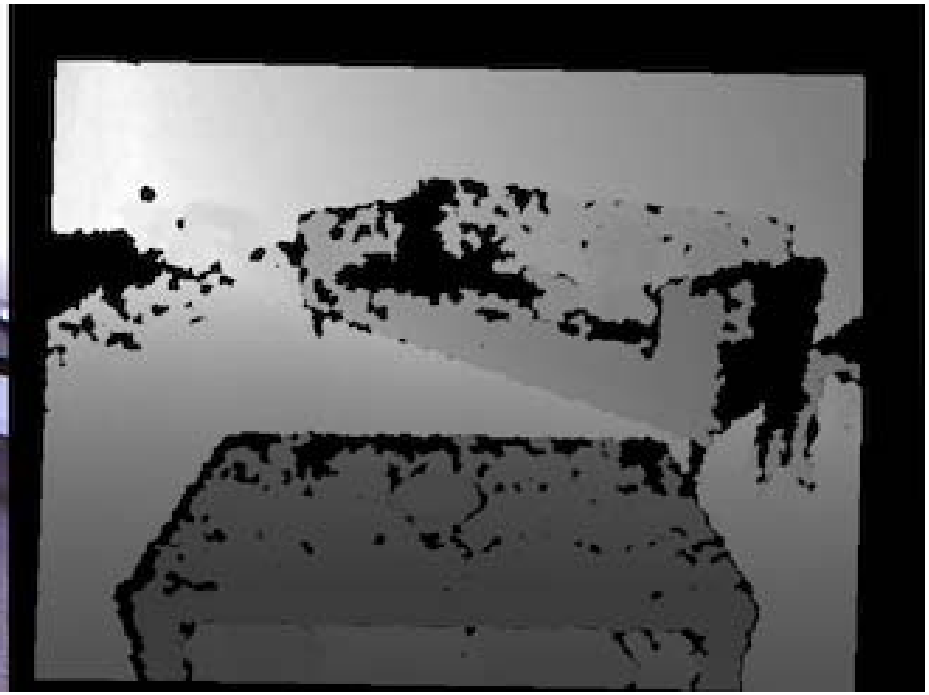
- Distance vs. noise



Noise characteristics

- Distance vs. noise
- Material property
- Missing & flying pixels





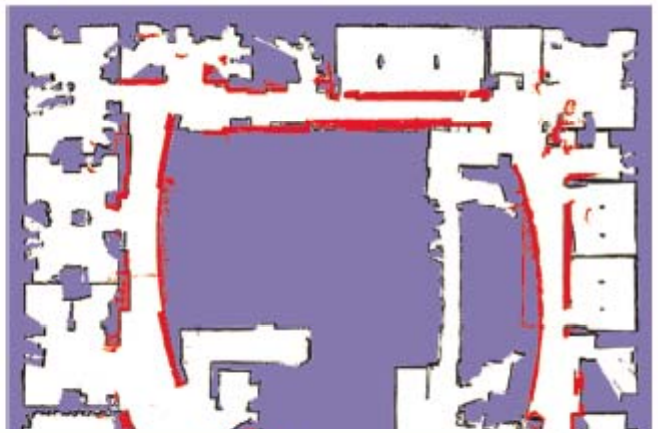
Noise characteristics

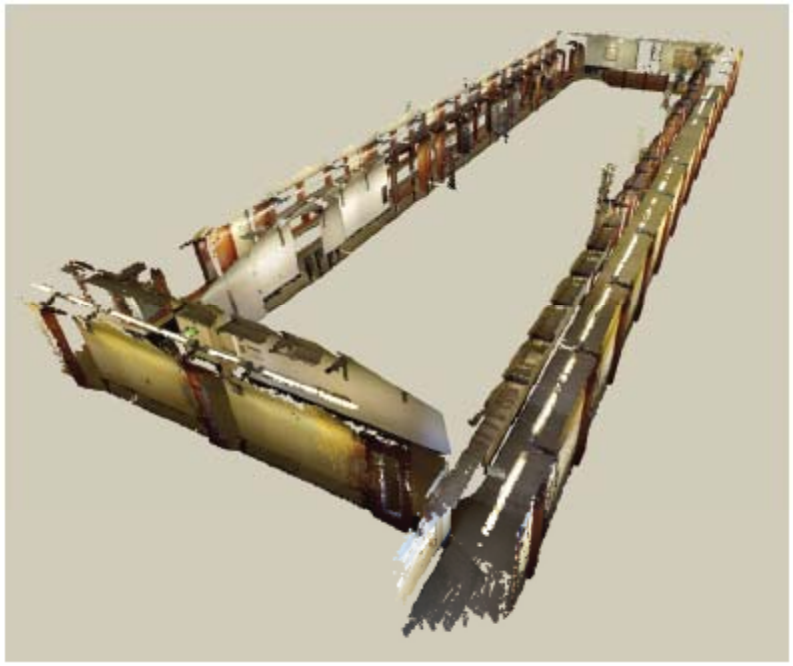
- Distance vs. noise
- Material property
- Missing & flying pixels
- Quantization noise

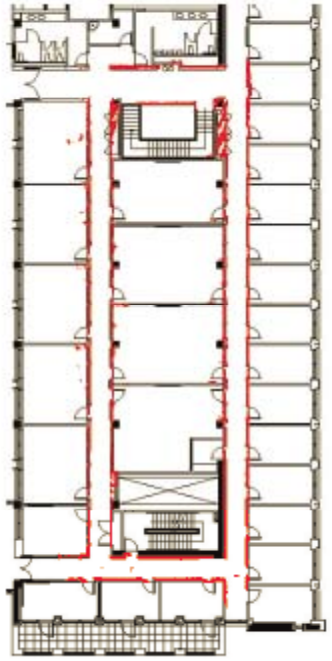
Data structure: 2.5 D

- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid

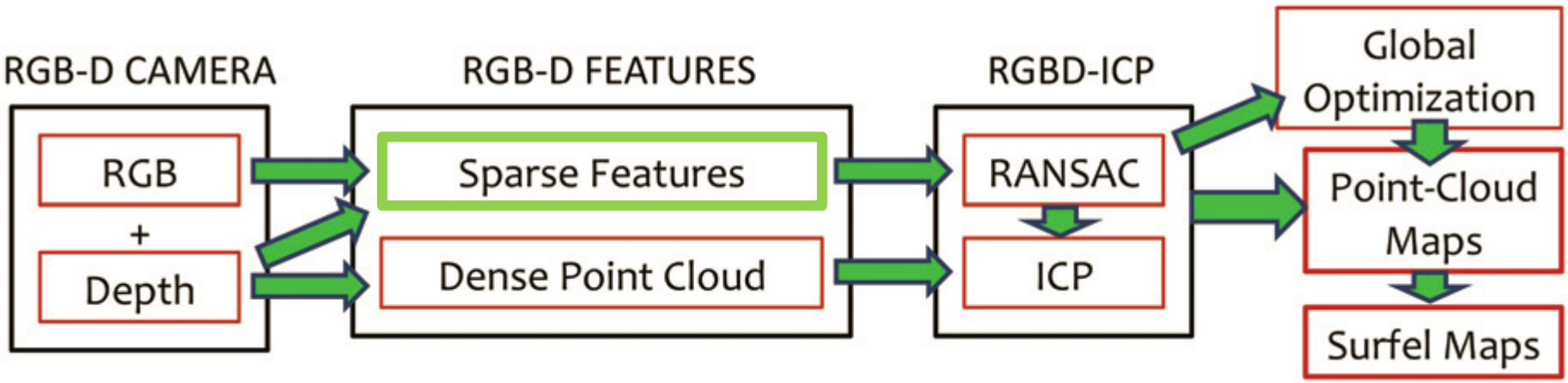
RGB-D mapping [Henry et al. 2010]

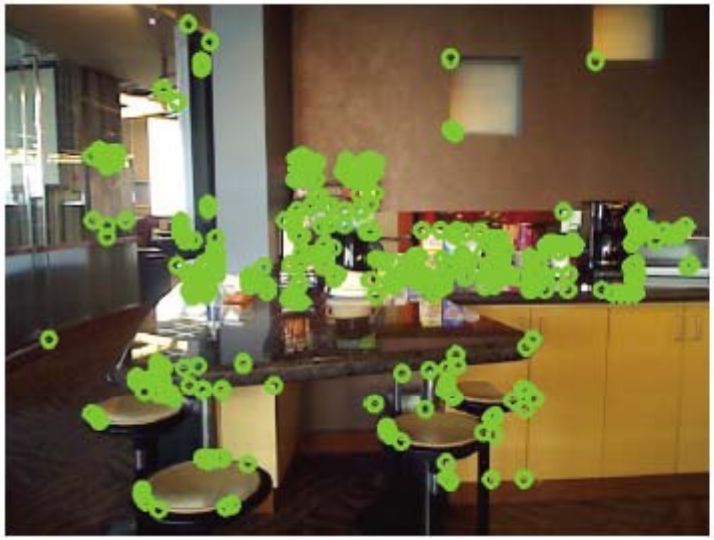




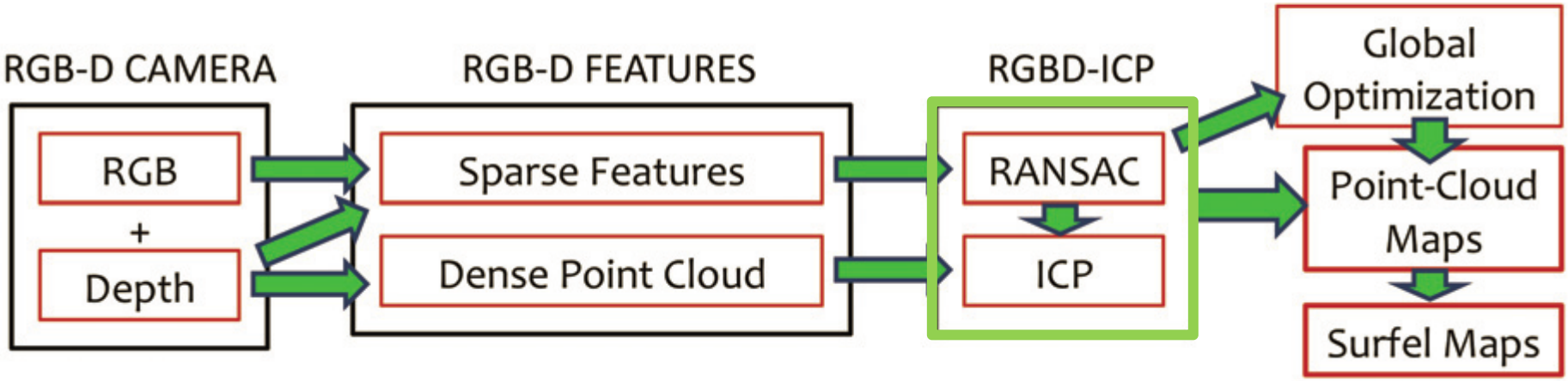


RGB-D mapping [Henry et al. 2010]





RGB-D mapping [Henry et al. 2010]



ICP (Iterative Closest Point) [Besl et al., 1992]

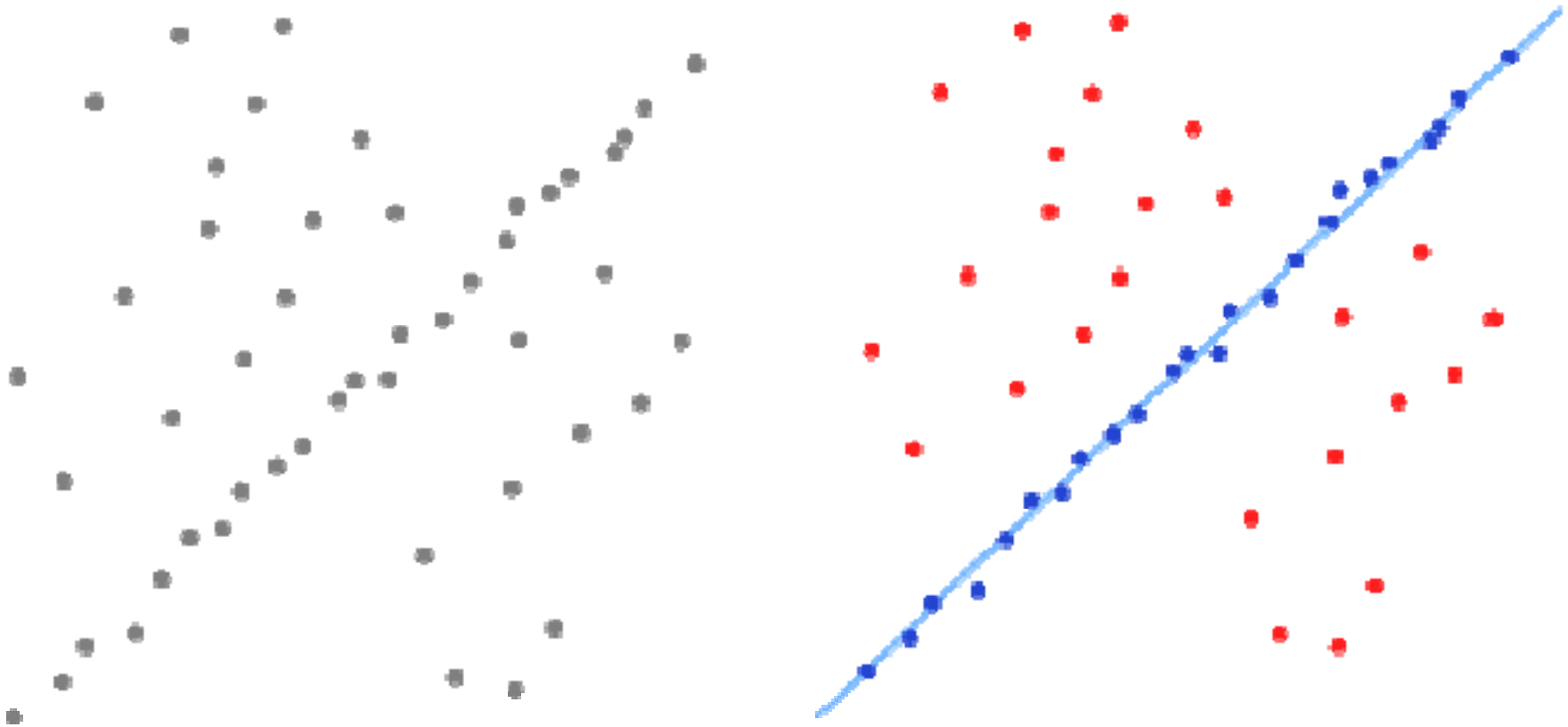
- Given two scans $\mathbf{P}$ and $\mathbf{Q}$.
- Iterate:
  - Find some pairs of closest points (p_i, q_i)
  - Find rotation $\mathbf{R}$ and translation $\mathbf{t}$ to minimize

$$\min_{\mathbf{R},\mathbf{t}} \sum_i \left\| p_i - \mathbf{R}q_i - \mathbf{t} \right\|^2$$

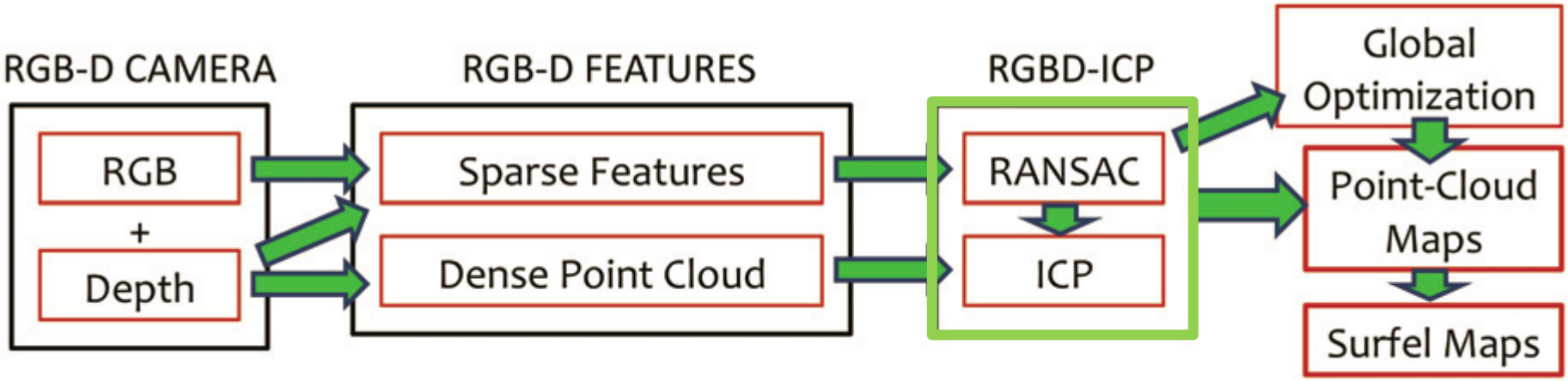
RANSAC [Fischler et al., 1981]

- RANdom Sample Consensus
- Parameter estimation, robust to outliers
- Algorithm
  - Input
    - Data
    - k: minimum number of samples needed for parameter assumption
    - e : error threshold
    - t: minimum number of inliers
    - N: number of iterations
  - iter=1:N
    - Sample k points from data
    - Solve for parameters with sampled points
    - Count number of inliers (within e)
    - If the number of inliers are more than t, then exit

RANSAC



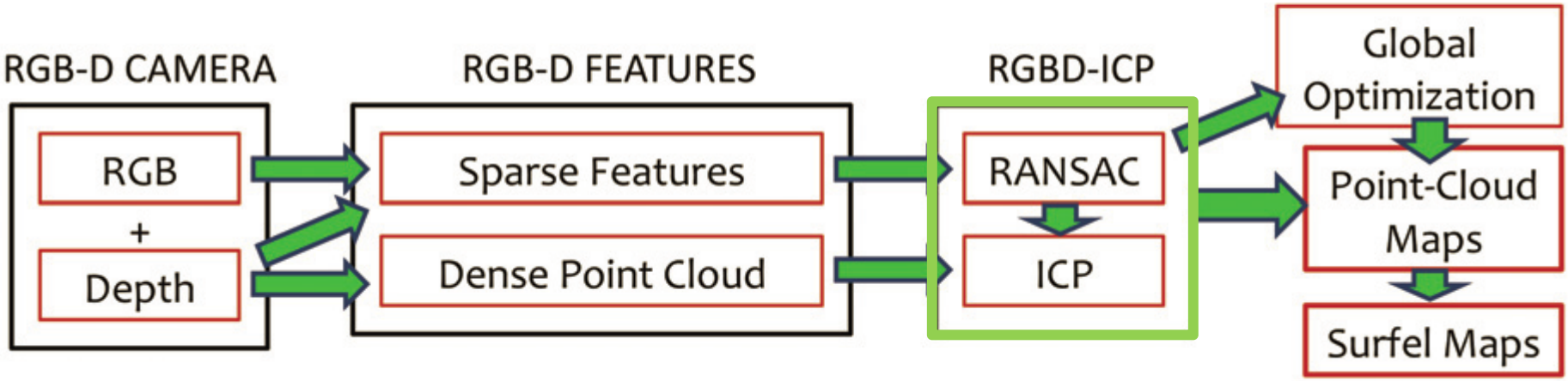
RGB-D mapping [Henry et al. 2010]



$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \left(\frac{1}{|A_f|} \sum_{i \in A_f} w_i \left| \mathbf{T}(f_s^i) - f_t^i \right|^2 \right)$$

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \left(\frac{1}{|A_f|} \sum_{i \in A_f} \left| Proj(\mathbf{T}(f_s^i)) - Proj(f_t^i) \right|^2 \right)$$

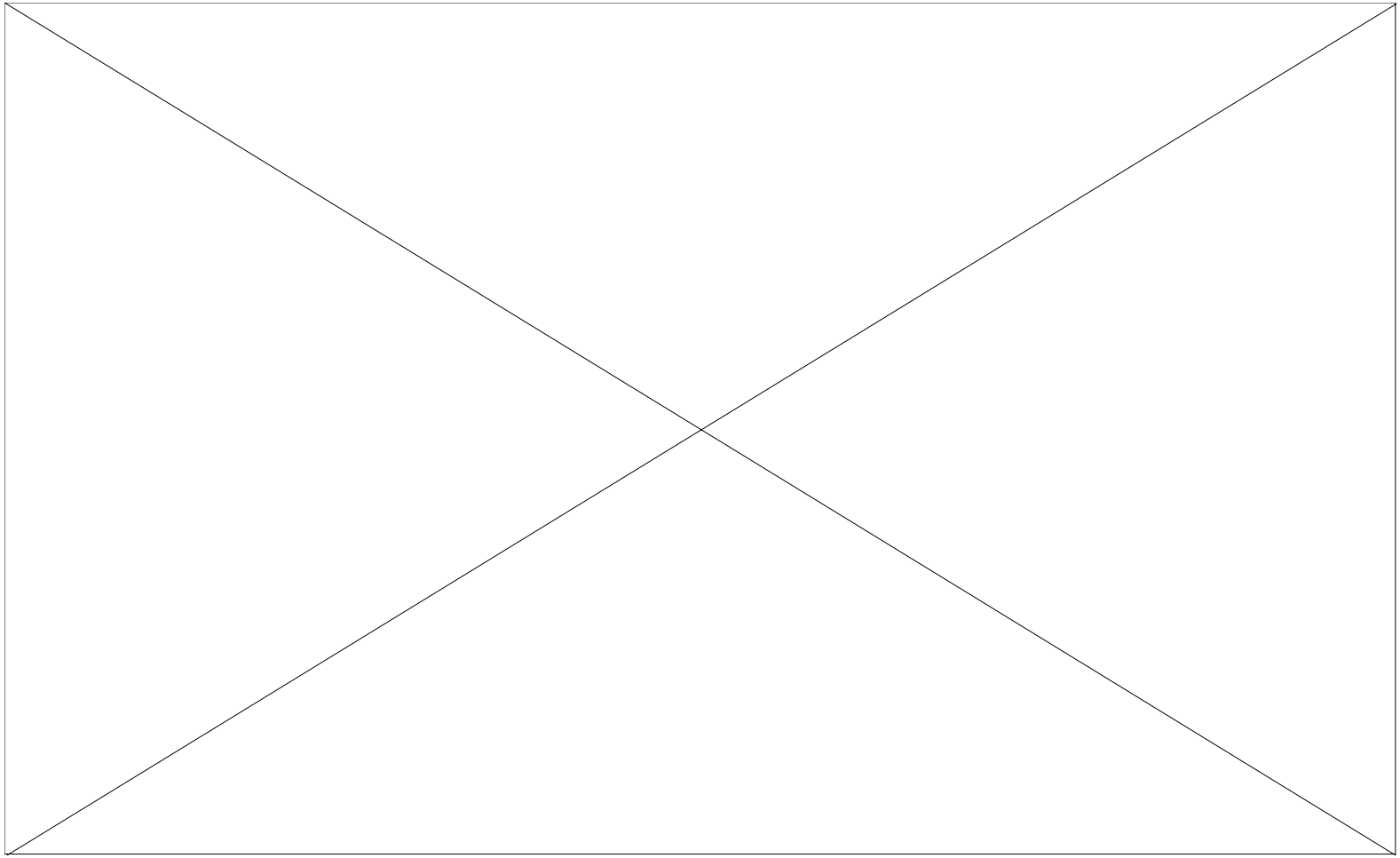
RGB-D mapping [Henry et al. 2010]



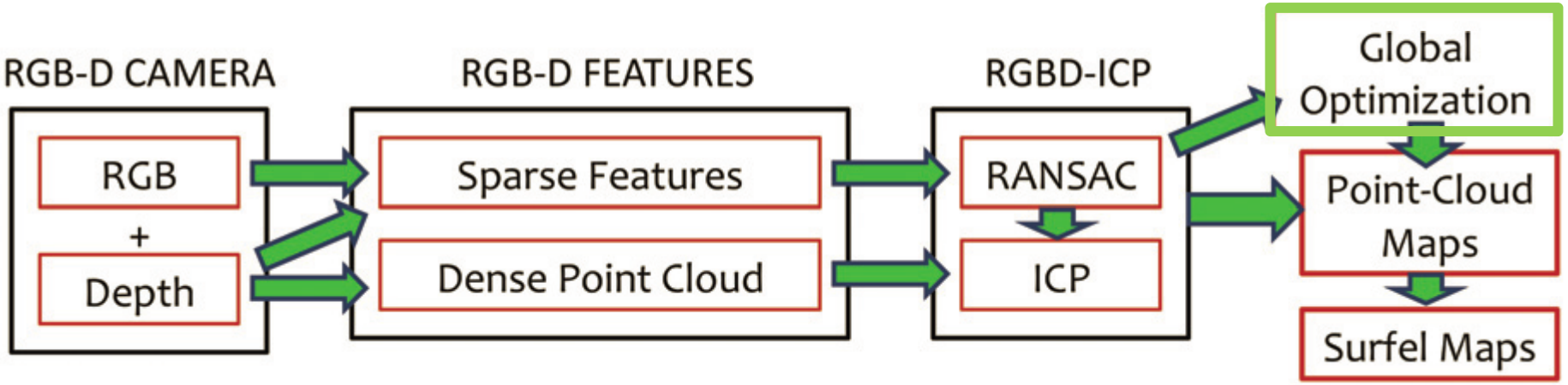
$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \left[\alpha \left(\frac{1}{|A_f|} \sum_{i \in A_f} w_i \left| \mathbf{T}(f_s^i) - f_t^i \right|^2 \right) + (1-\alpha) \left(\frac{1}{|A_d|} \sum_{j \in A_d} w_j \left| (\mathbf{T}(p_s^j) - p_t^j) \cdot n_t^j \right|^2 \right) \right]$$

$$\mathbf{T}^* = \underset{\mathbf{T}}{\operatorname{argmin}} \left[\left(\frac{1}{|A_f|} \sum_{i \in A_f} \left| Proj(\mathbf{T}(f_s^i)) - Proj(f_t^i) \right|^2 \right) + \beta \left(\frac{1}{|A_d|} \sum_{j \in A_d} w_j \left| (\mathbf{T}(p_s^j) - p_t^j) \cdot n_t^j \right|^2 \right) \right].$$

RGB-D mapping [Henry et al. 2010]



RGB-D mapping [Henry et al. 2010]



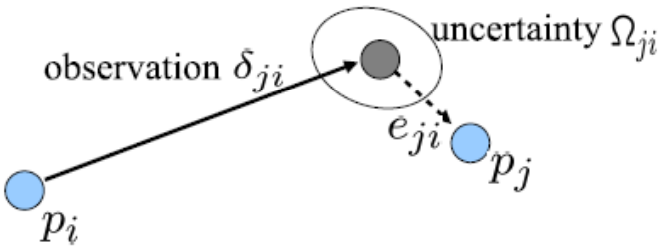
- Loop closure detection
  - Feature matching
- Global optimization
  - Pose graph optimization
  - Sparse bundle adjustment

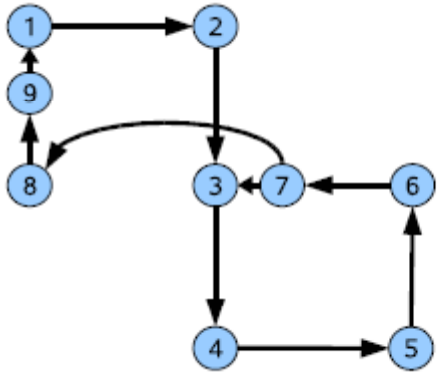
Loop closure detection

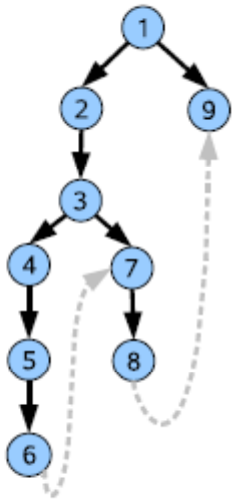
- Every frame to every other frame
- Key frames
  - Every n-th frame
  - Compute visual overlap
- Filter key frames
  - Estimated global pose
  - Place recognition algorithm

Pose graph optimization [Grisetti et al., 2009]

- Vertex: pose of camera
- Edge: constraint between a pair of vertices
- Uncertainty assigned for every edge
- Use tree structure for optimization

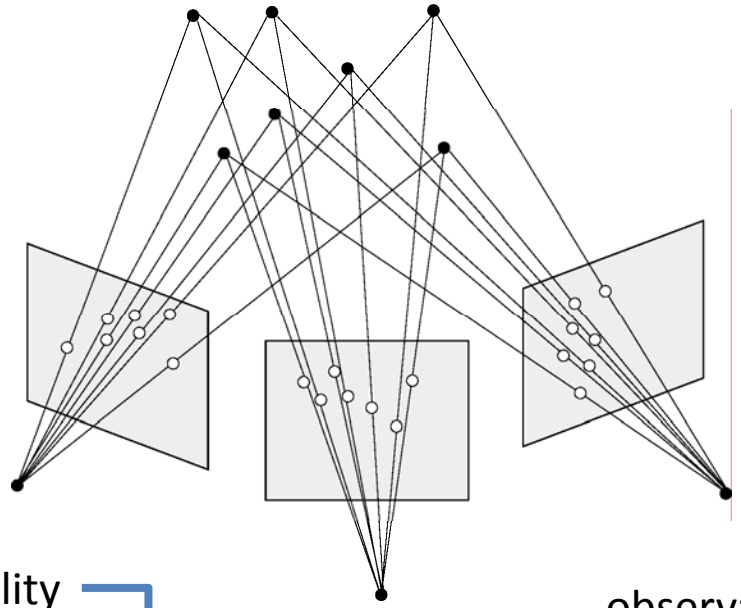






Sparse bundle adjustment [Lourakis et al., 2009]

- Minimize re-projection error of feature points



visibility

observation

$$\sum_{c_i \in C} \sum_{p_j \in P} v_{ij} \left| Proj(c_i(p_j)) - (\bar{u}, \bar{v}, \bar{d}) \right|^2$$

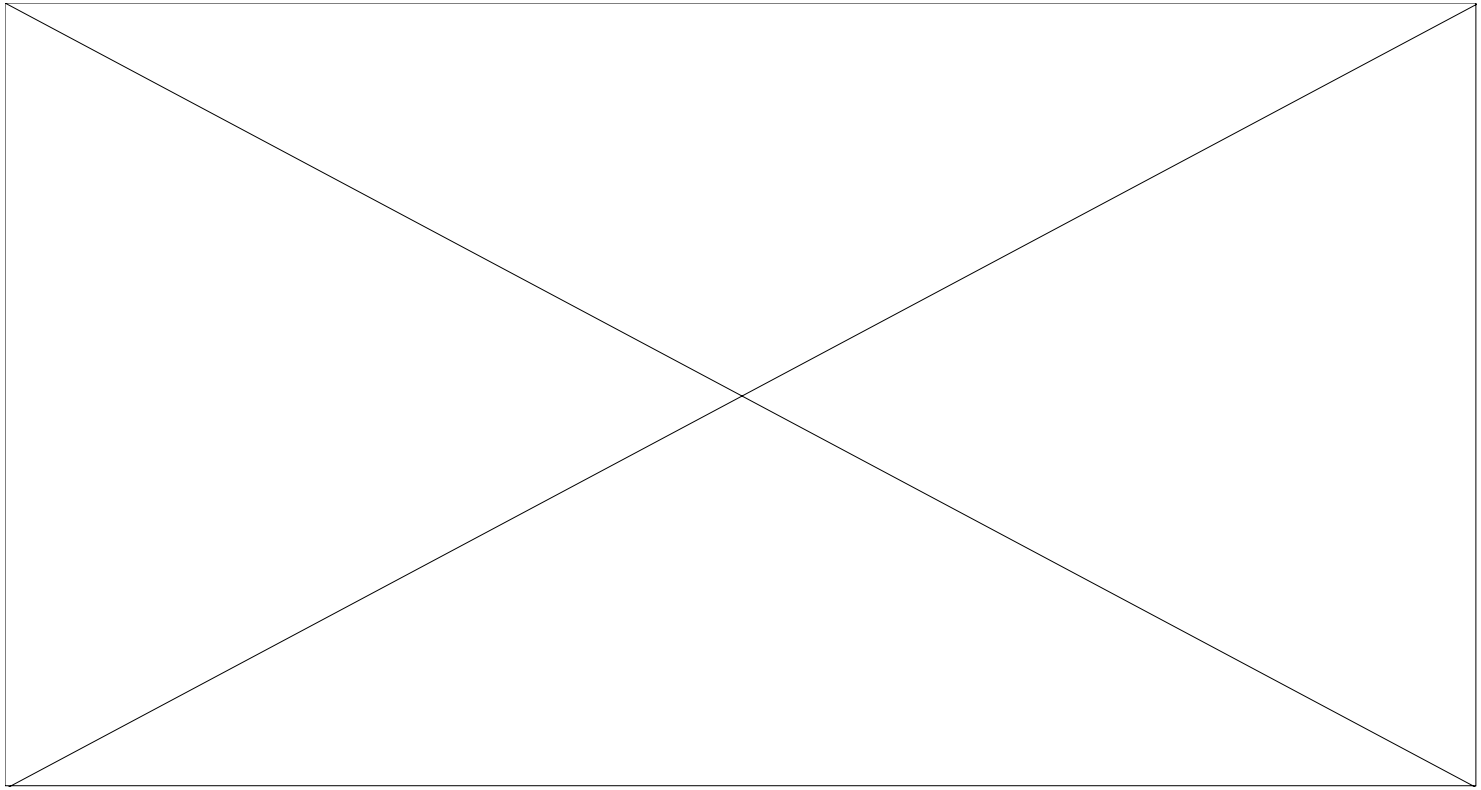
Projection of feature points

camera

points

RGB-D mapping [Henry et al. 2010]

- Map visualization



Data structure: 2.5 D

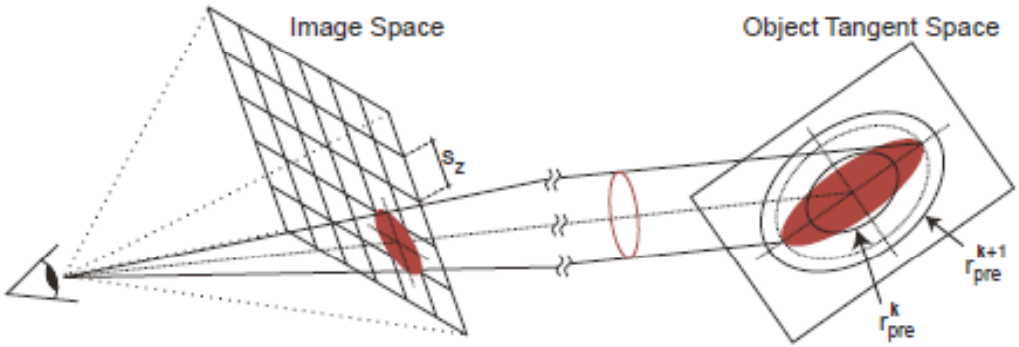
- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid

Data structure: 2.5 D

- **Point cloud**
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid

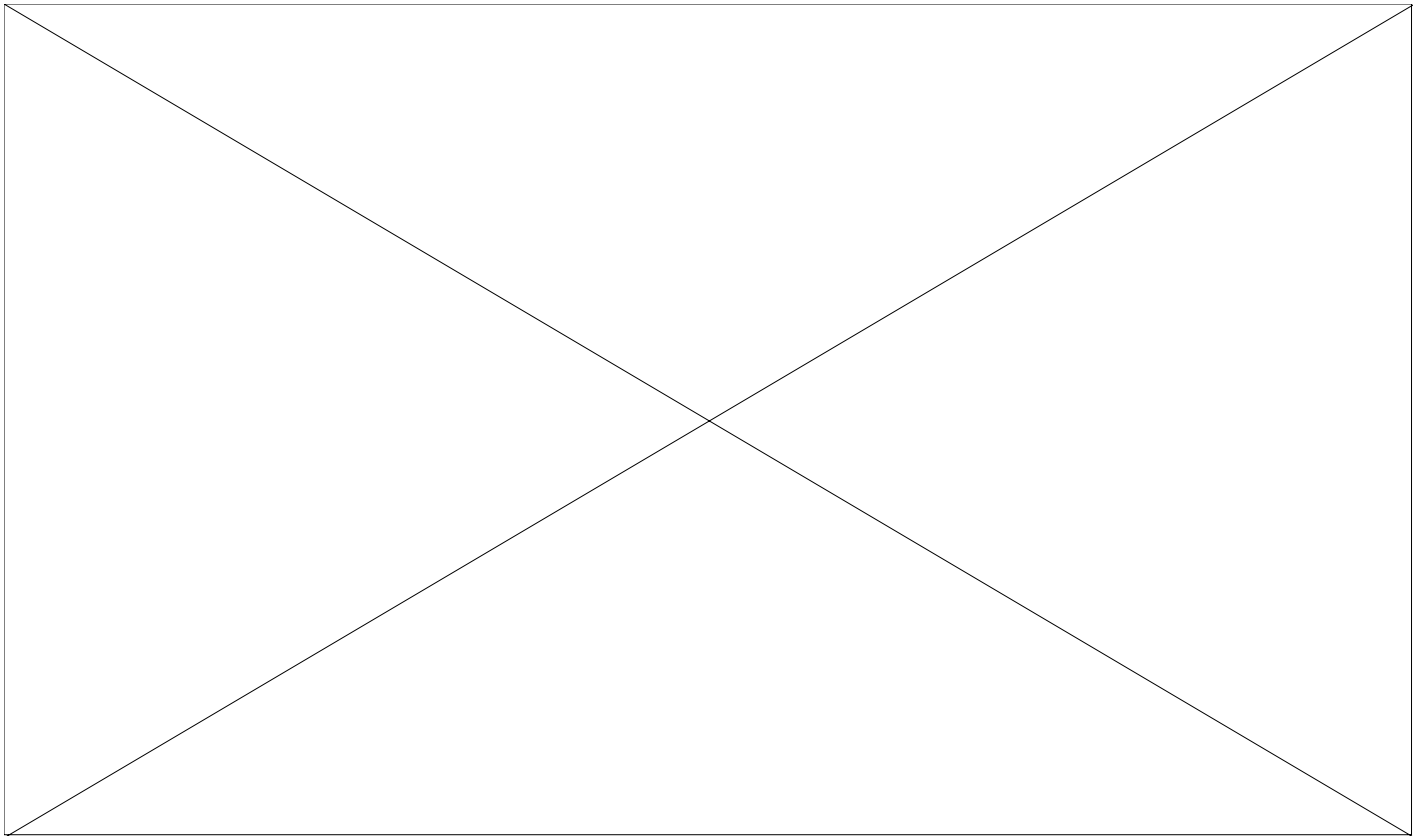
Surfels [Pfister et al. 2000]

- Display purpose
- Components
  - Location
  - Normal
  - Patch size: inferred from distance & pixel size
  - Color: choose the most direct view
  - Confidence: calculated from the histogram of accumulated normal



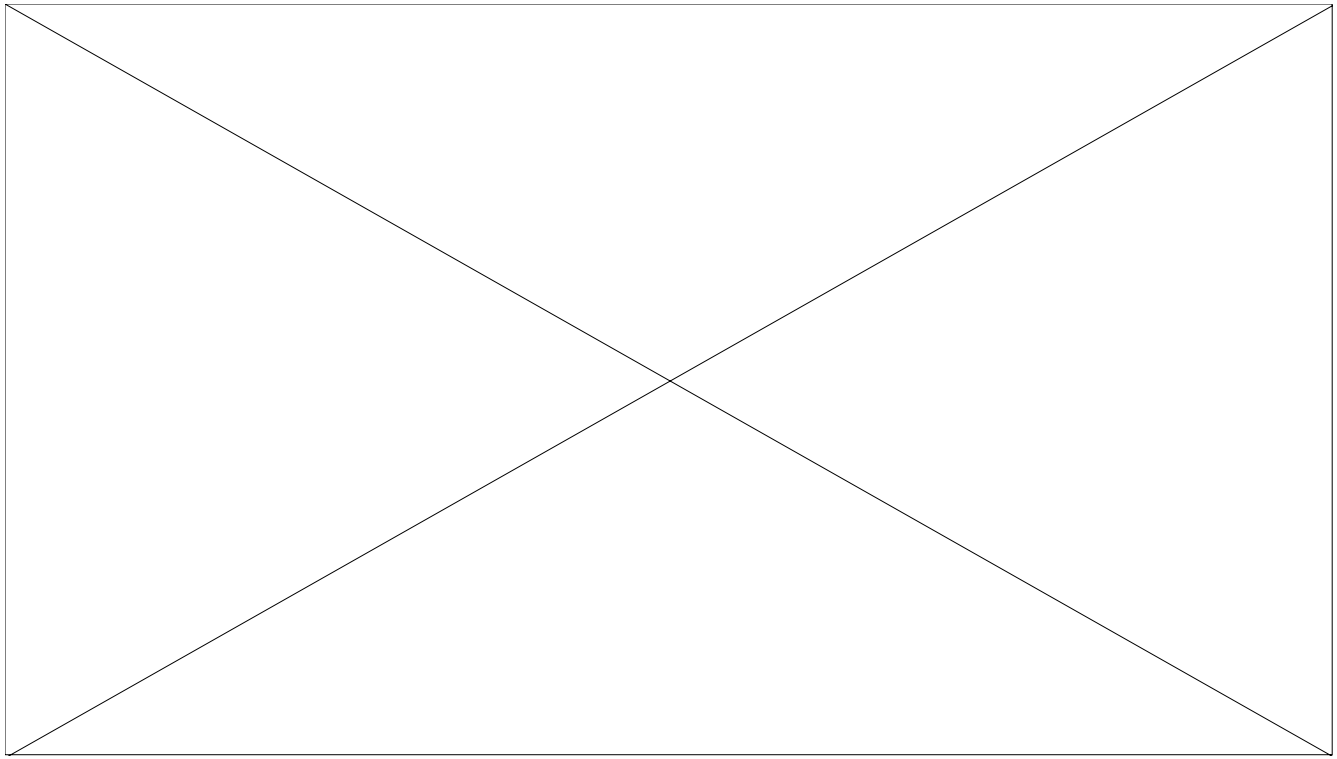
RGB-D mapping [Henry et al. 2010]

- Surfels



In-hand 3D object modeling [Krainin et al., 2011]

- Real-time aspect

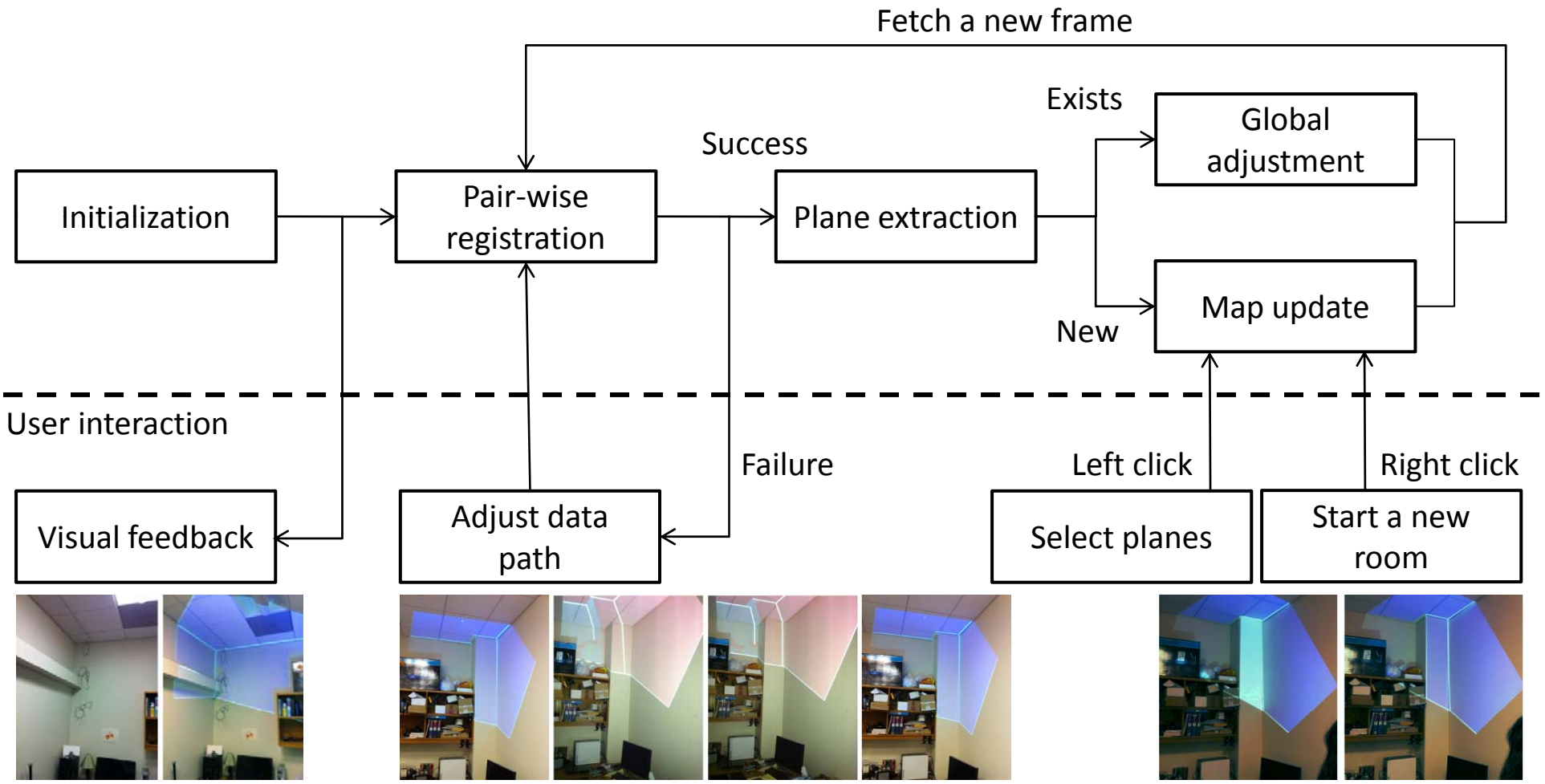


Interactivity

[Mistry et al 2009]

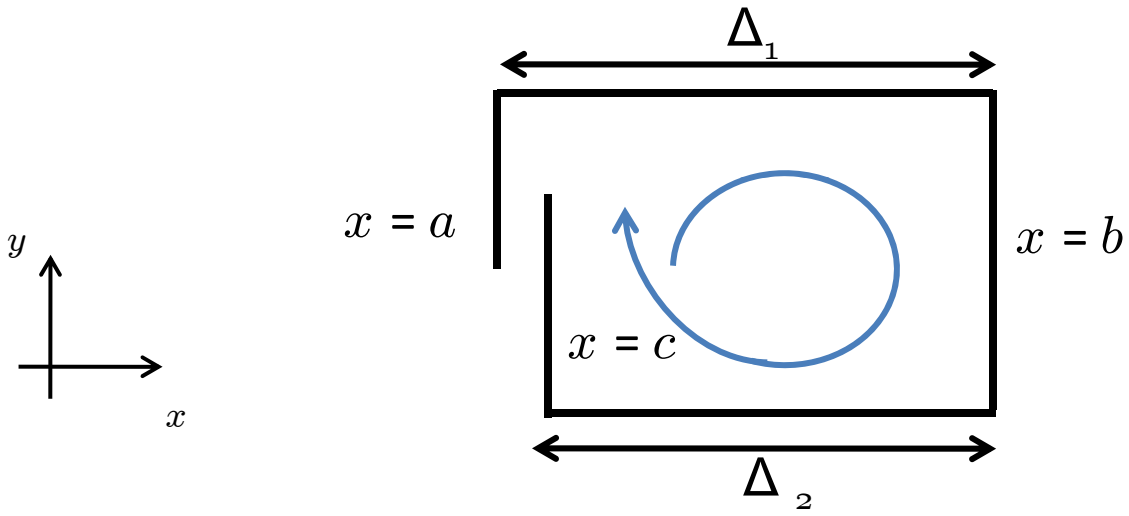
XBOX 360

Algorithm

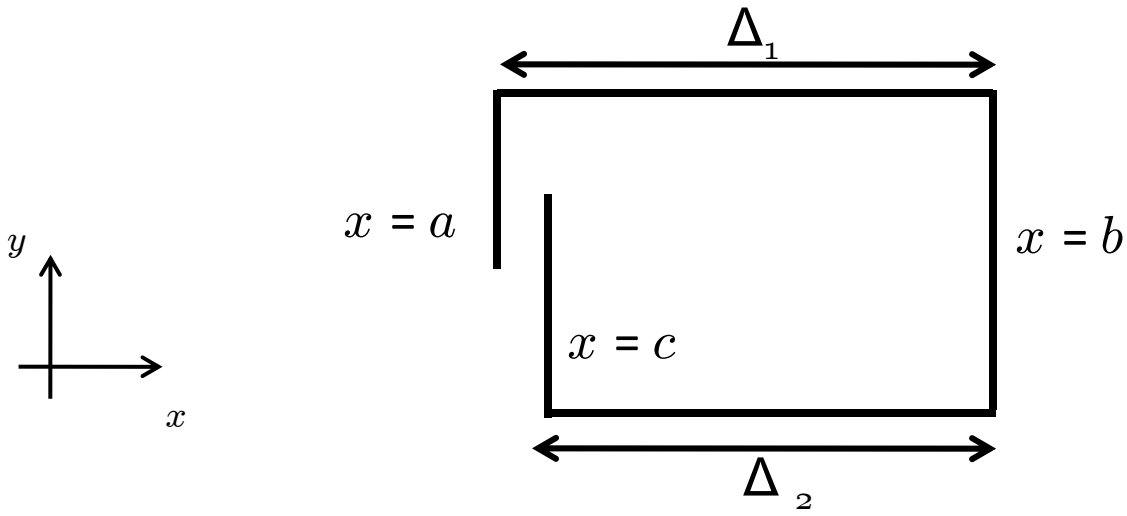

Fetch a new frame
Exists
Global adjustment
Success
Initialization
Pair-wise registration
Plane extraction
Map update
New
User interaction
Failure
Left click
Right click
Visual feedback
Adjust data path
Select planes
Start a new room

Registration failure

Global Adjustment

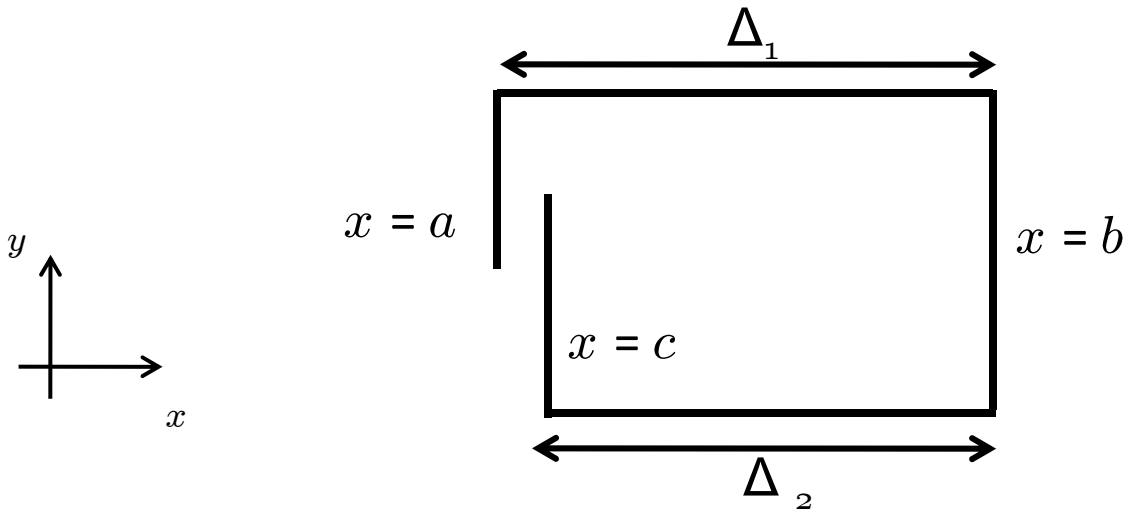


Global Adjustment



a=c

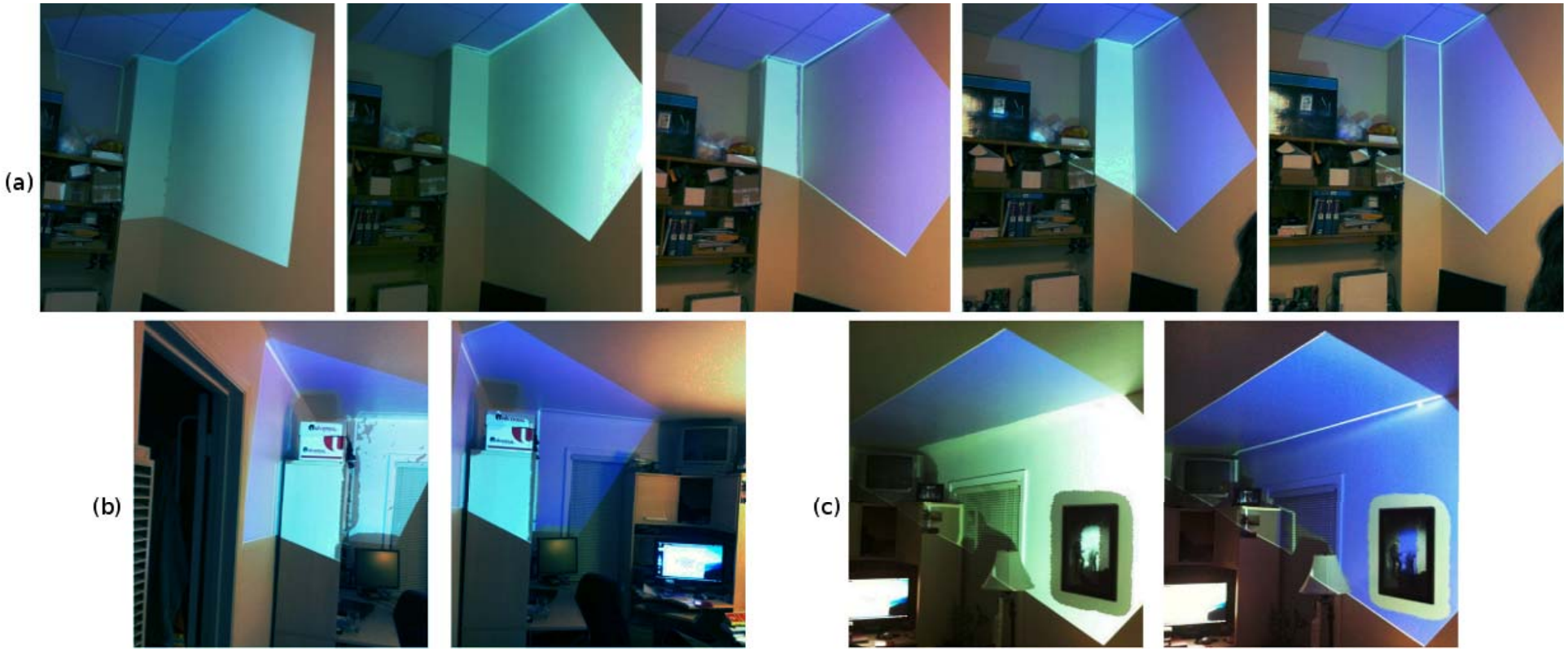
Global Adjustment



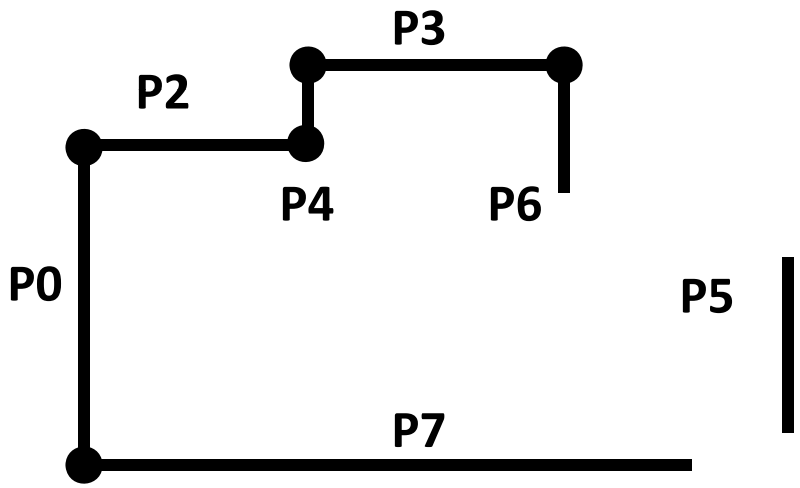
a=c

$$\min_{\mathcal{S}^x} \sum_i \left(\frac{\|\Delta_i - m_i\|^2}{\sigma_i^2} \right) \quad \text{s. t. } c_{j1} = c_{j2}, \ \forall (c_{j1}, c_{j2}) \in \mathcal{C}^x$$

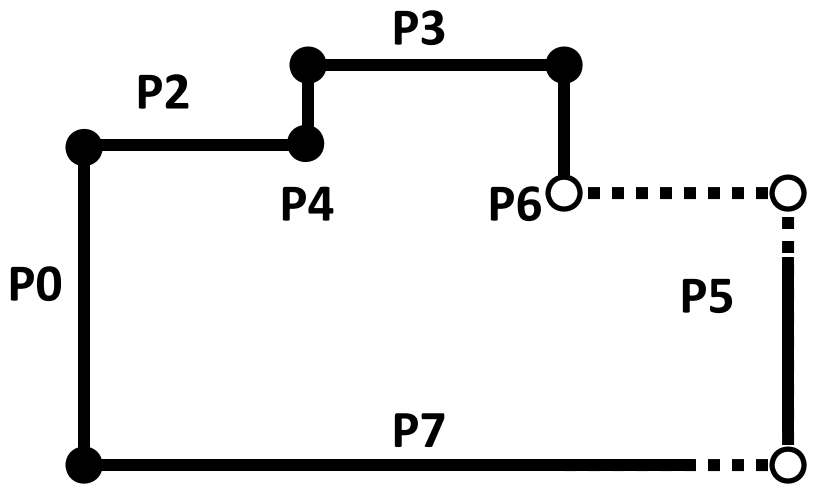
Selecting components

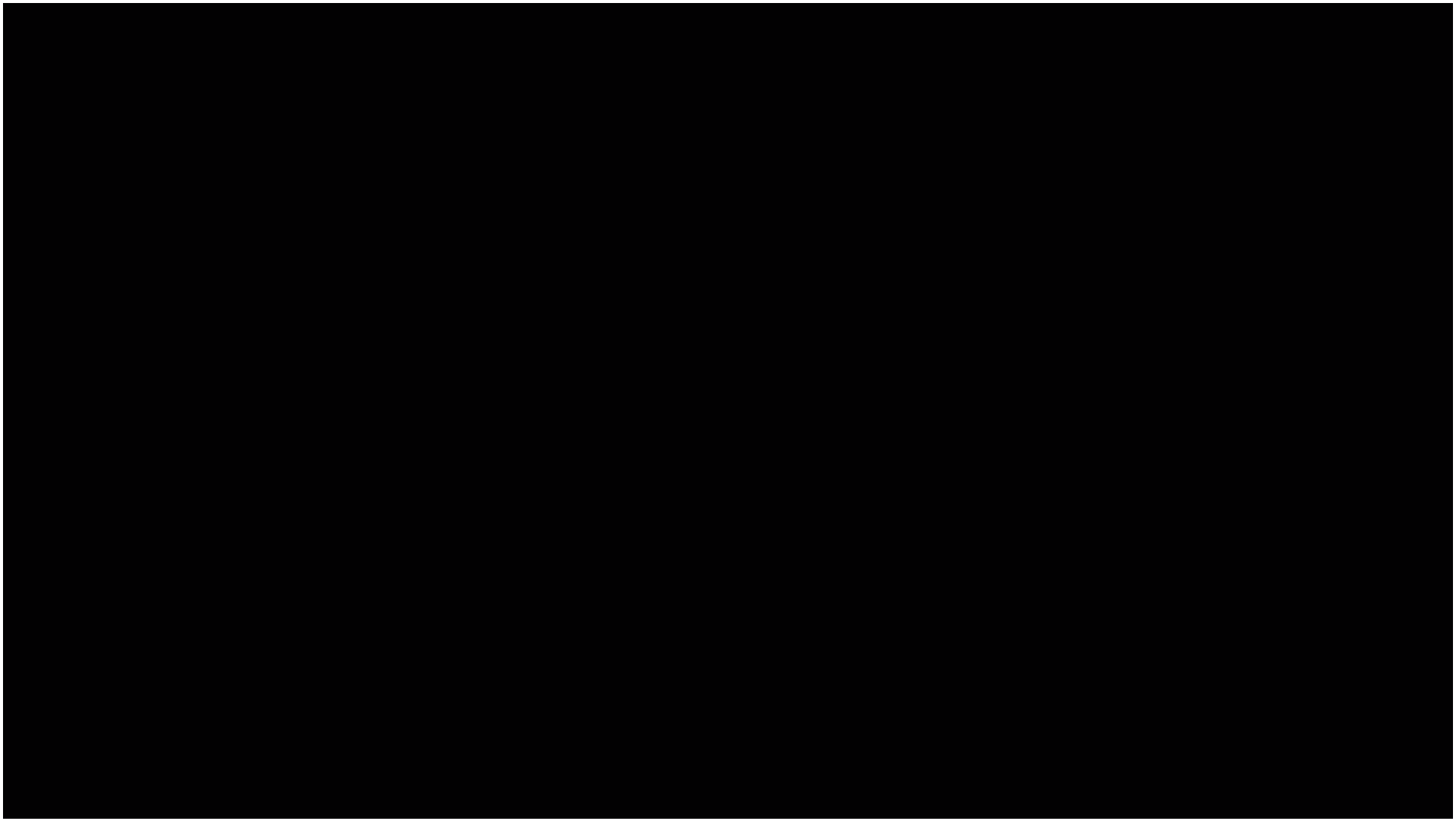


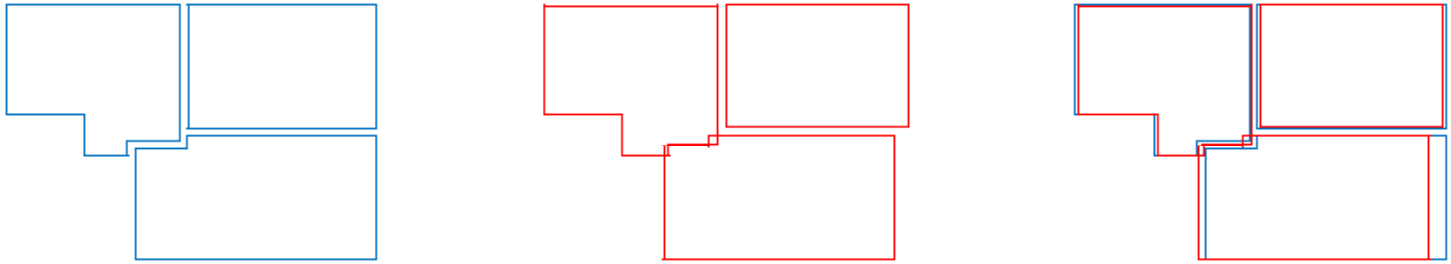
Floor plan generation



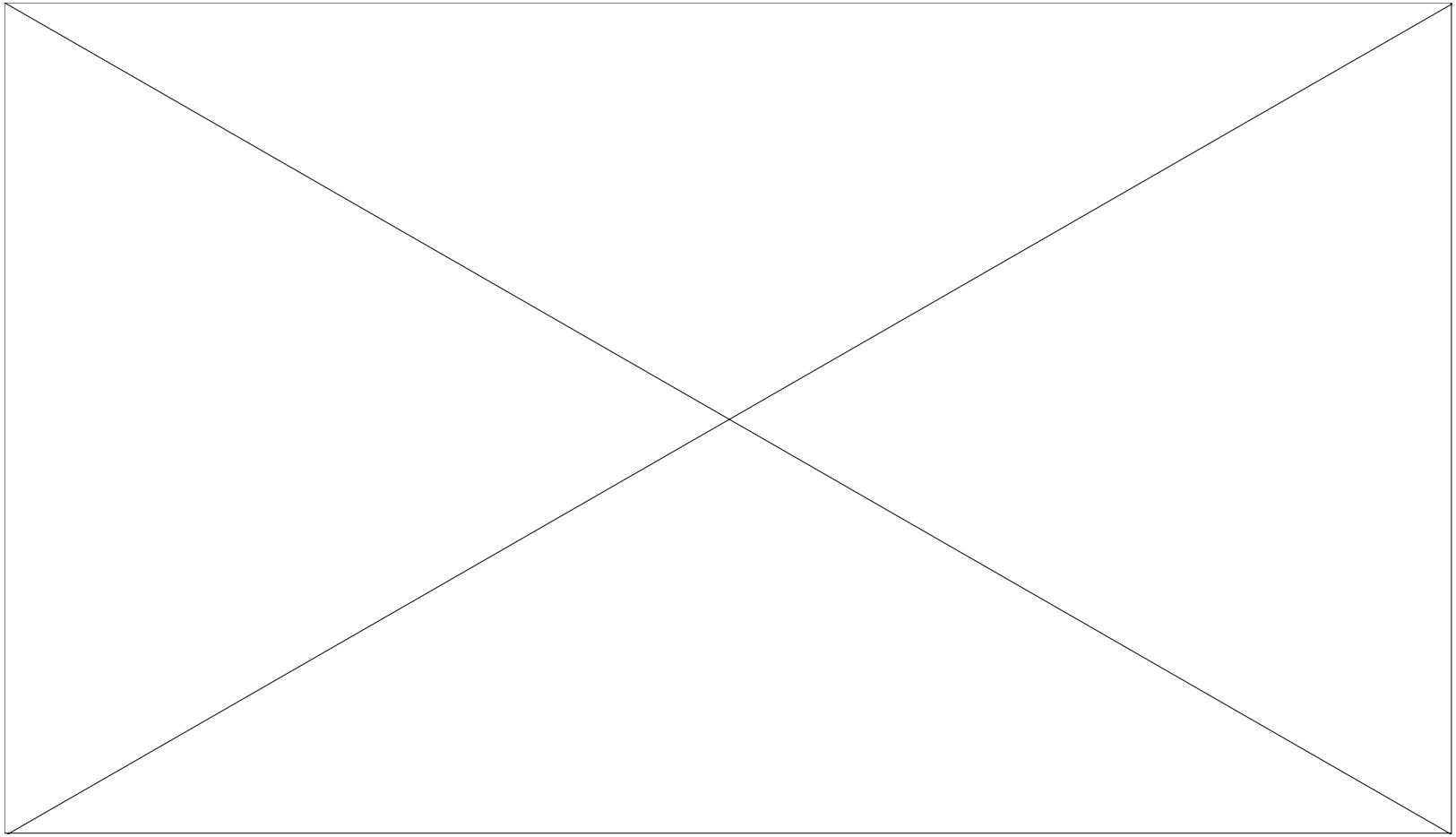
Floor plan generation



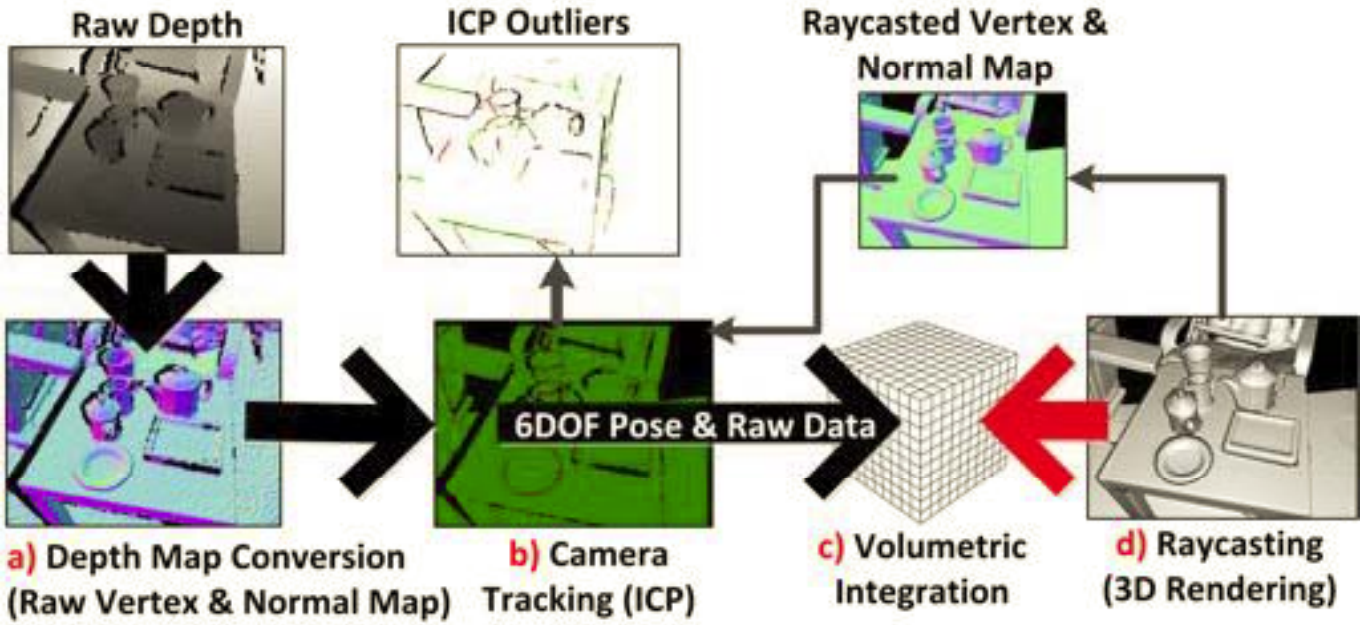




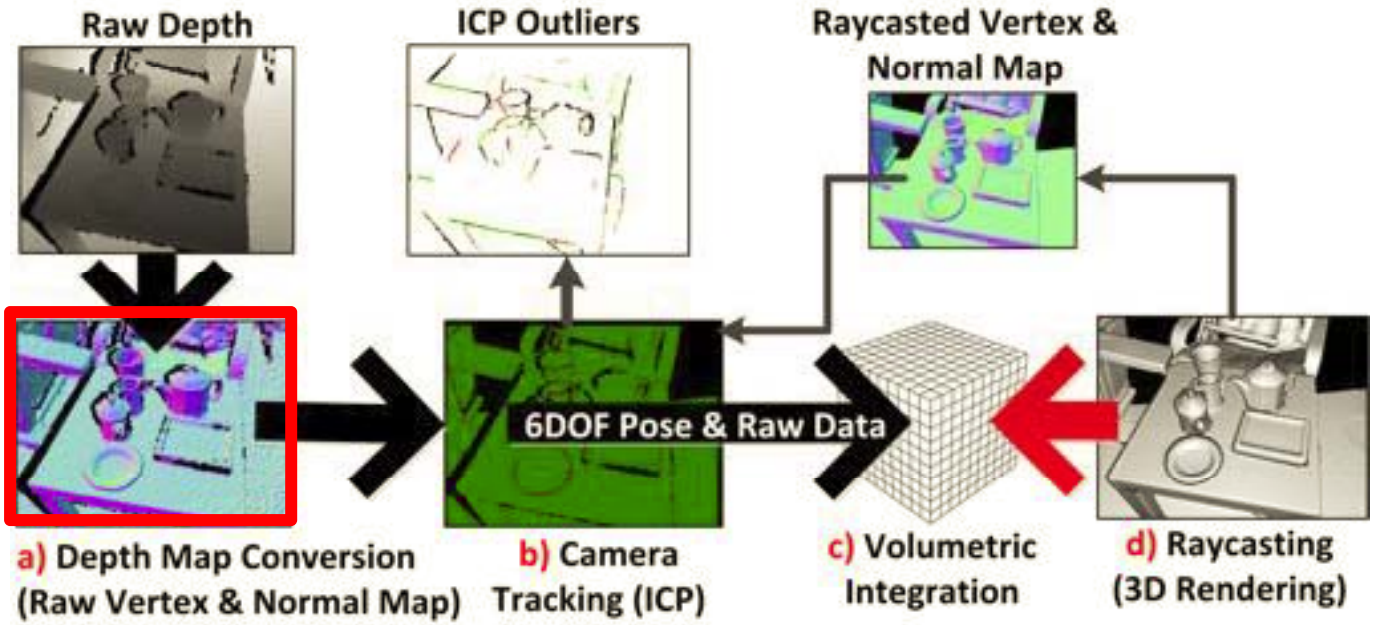
Kinect Fusion [Izadi et al. 2011]



Kinect Fusion [Izadi et al. 2011]

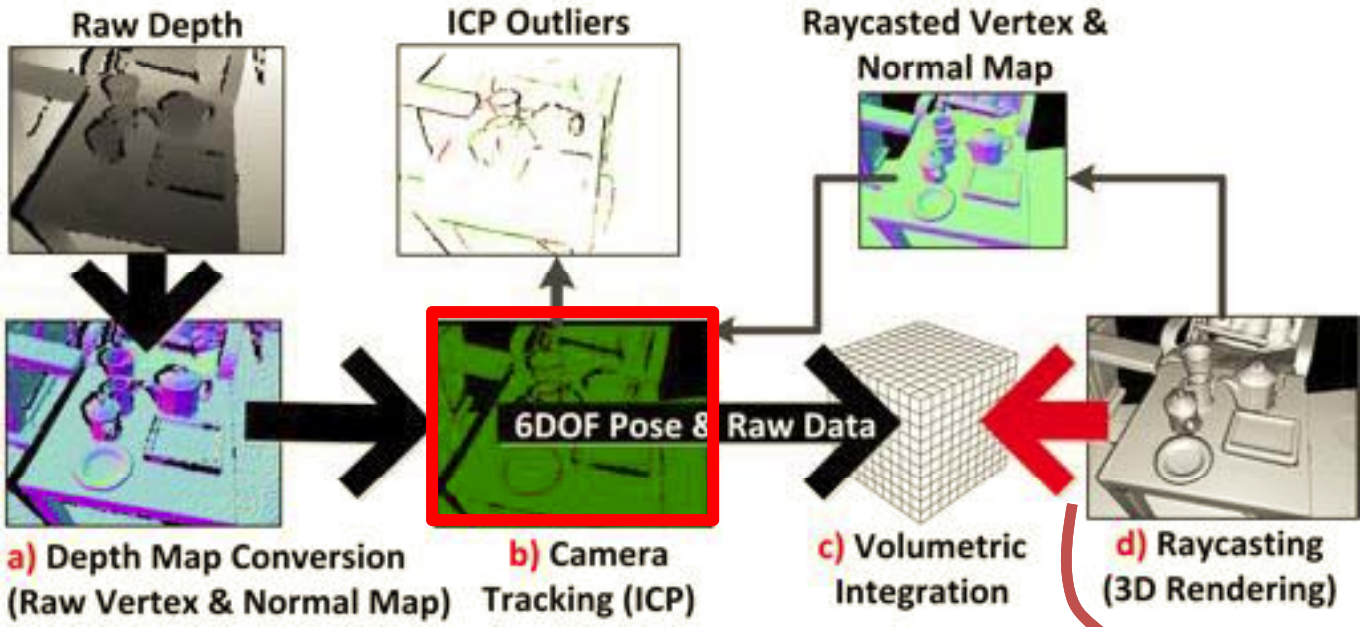


Kinect Fusion [Izadi et al. 2011]



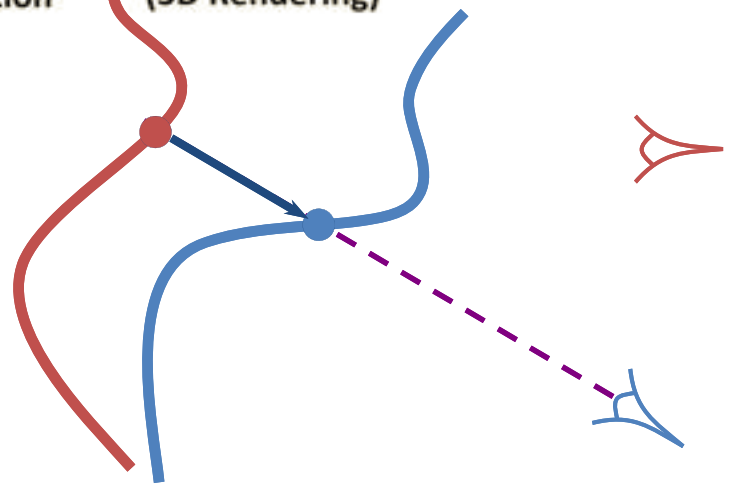
Use 2-D grid to estimate the normals $\frac{\partial \mathbf{x}}{\partial s} \times \frac{\partial \mathbf{x}}{\partial t}$.

Kinect Fusion [Izadi et al. 2011]

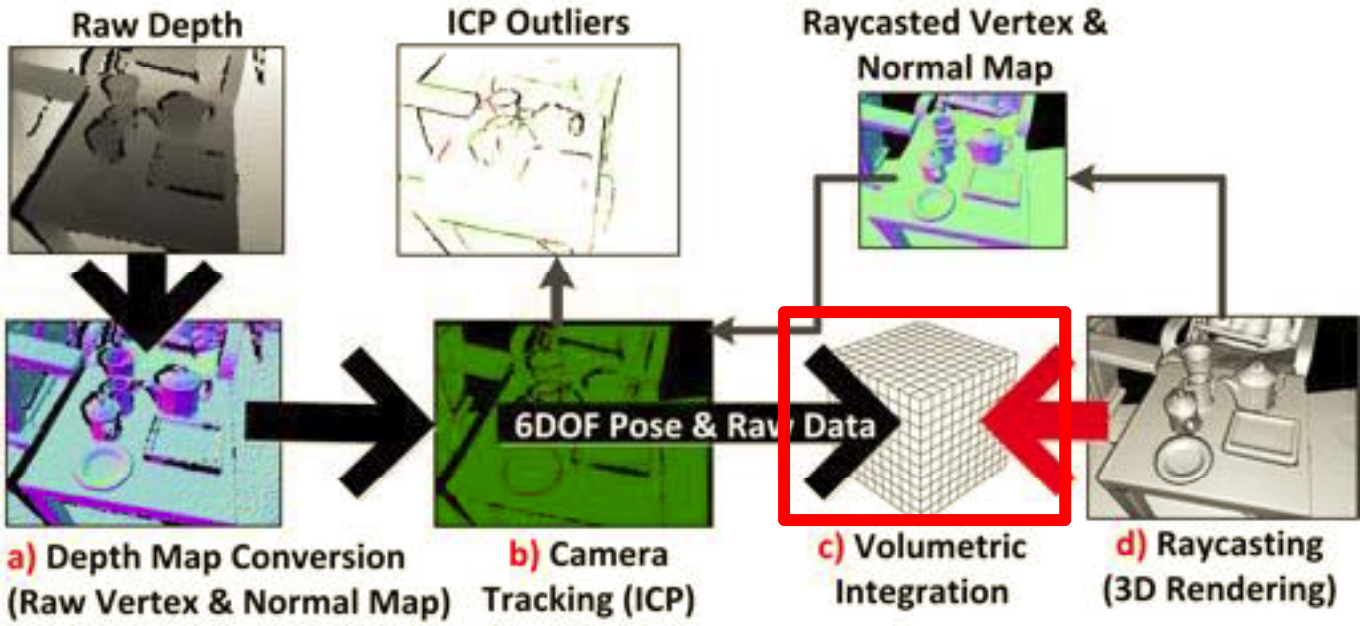


Dense ICP using GPU

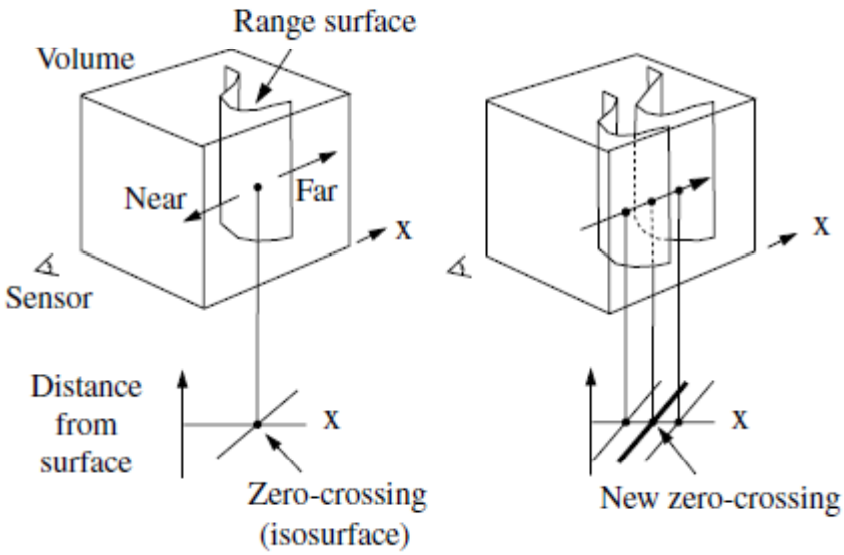
Projective data association



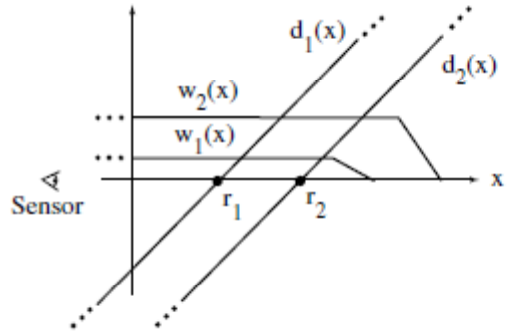
Kinect Fusion [Izadi et al. 2011]

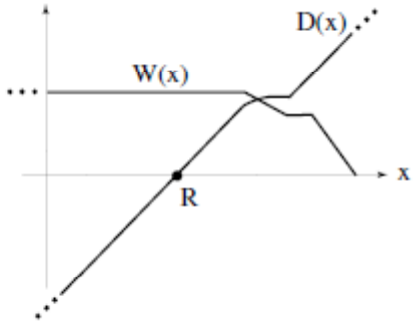


Signed distance function [Curless et al. 1996]

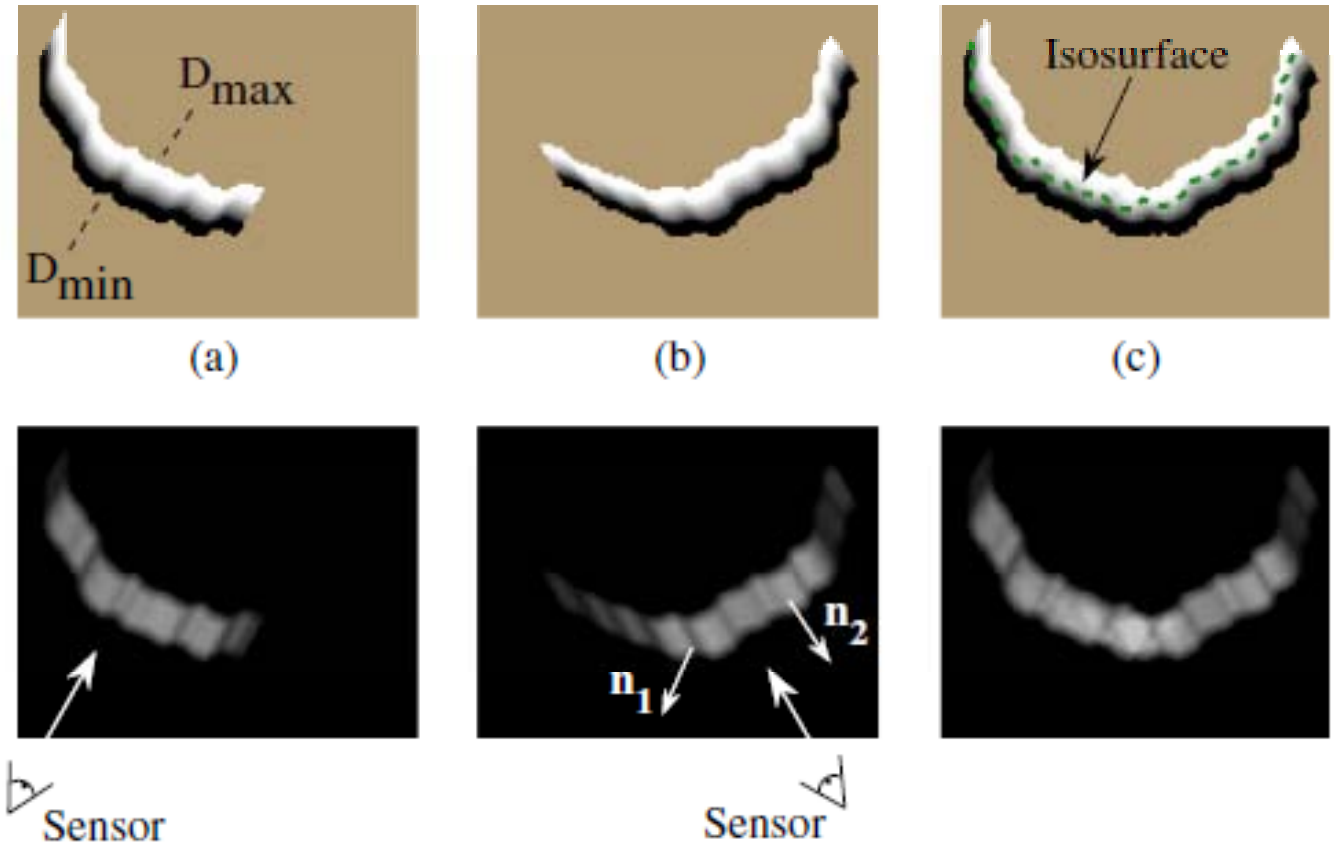


$$D(\mathbf{x}) = \frac{\Sigma w_i(\mathbf{x}) d_i(\mathbf{x})}{\Sigma w_i(\mathbf{x})}$$

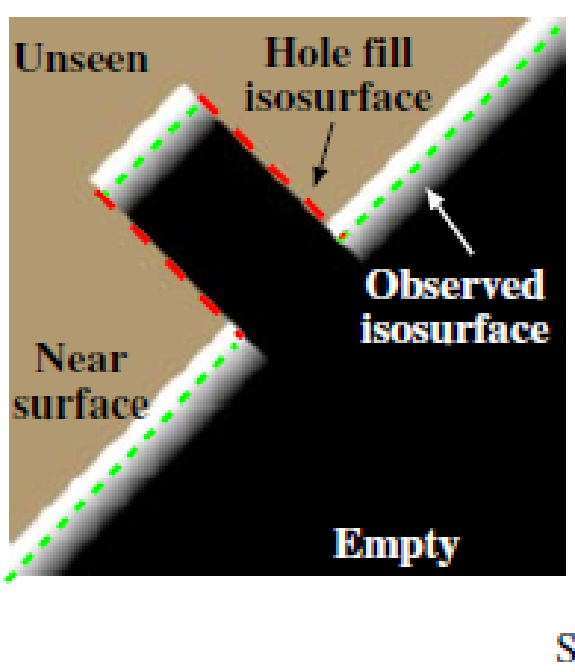


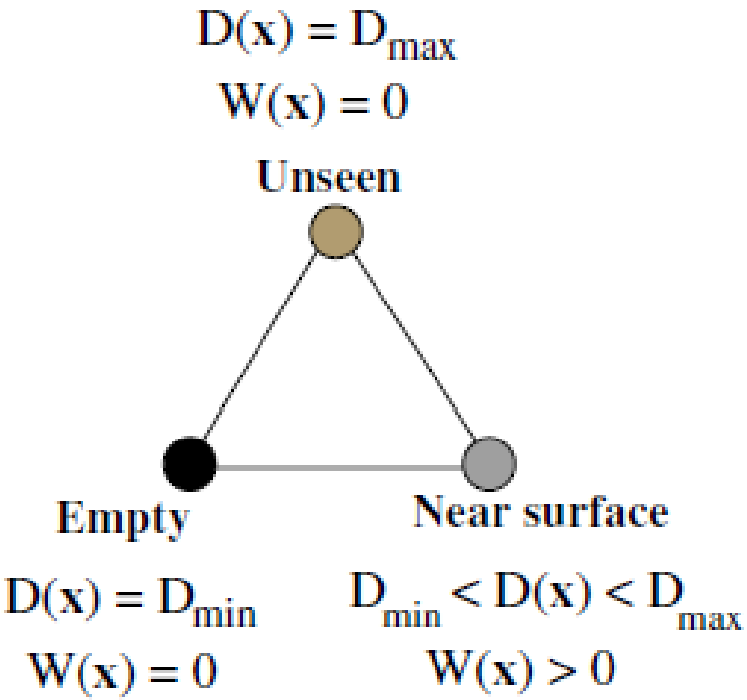


Signed distance function [Curless et al. 1996]



Signed distance function [Curless et al. 1996]





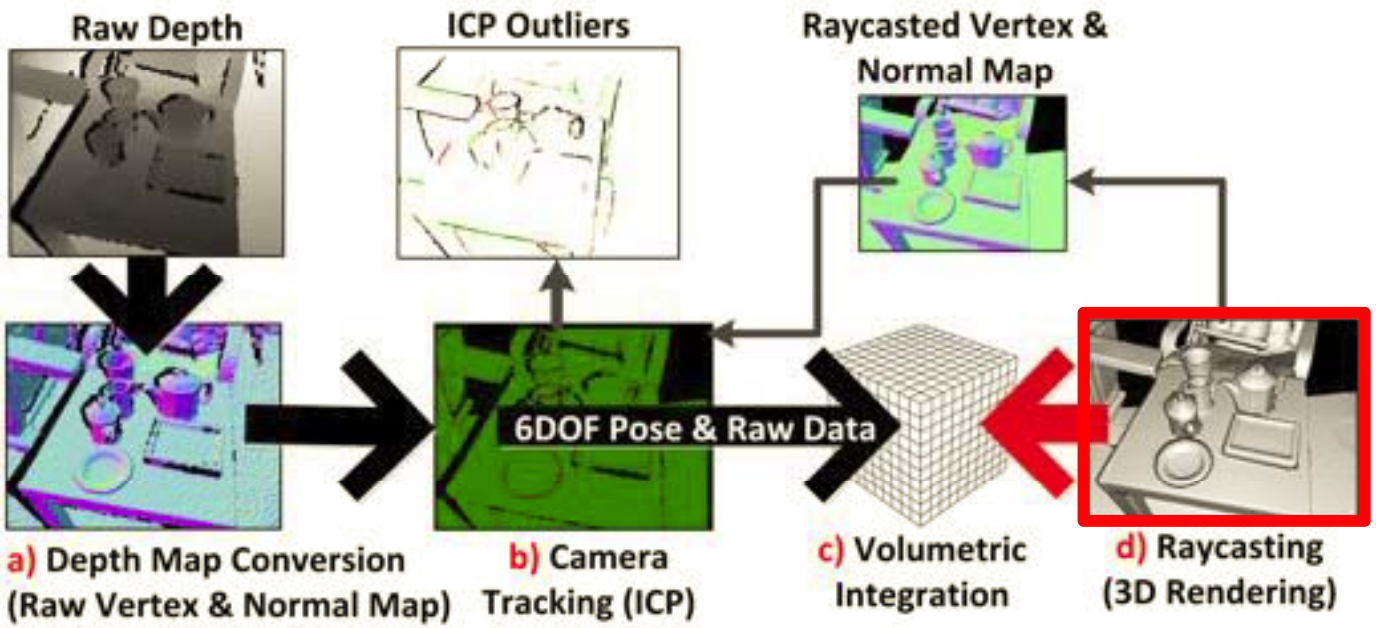
Data structure: 2.5 D

- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid

Data structure: 2.5 D

- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- **Robotics**
  - **Ray-based model, voxel grid**

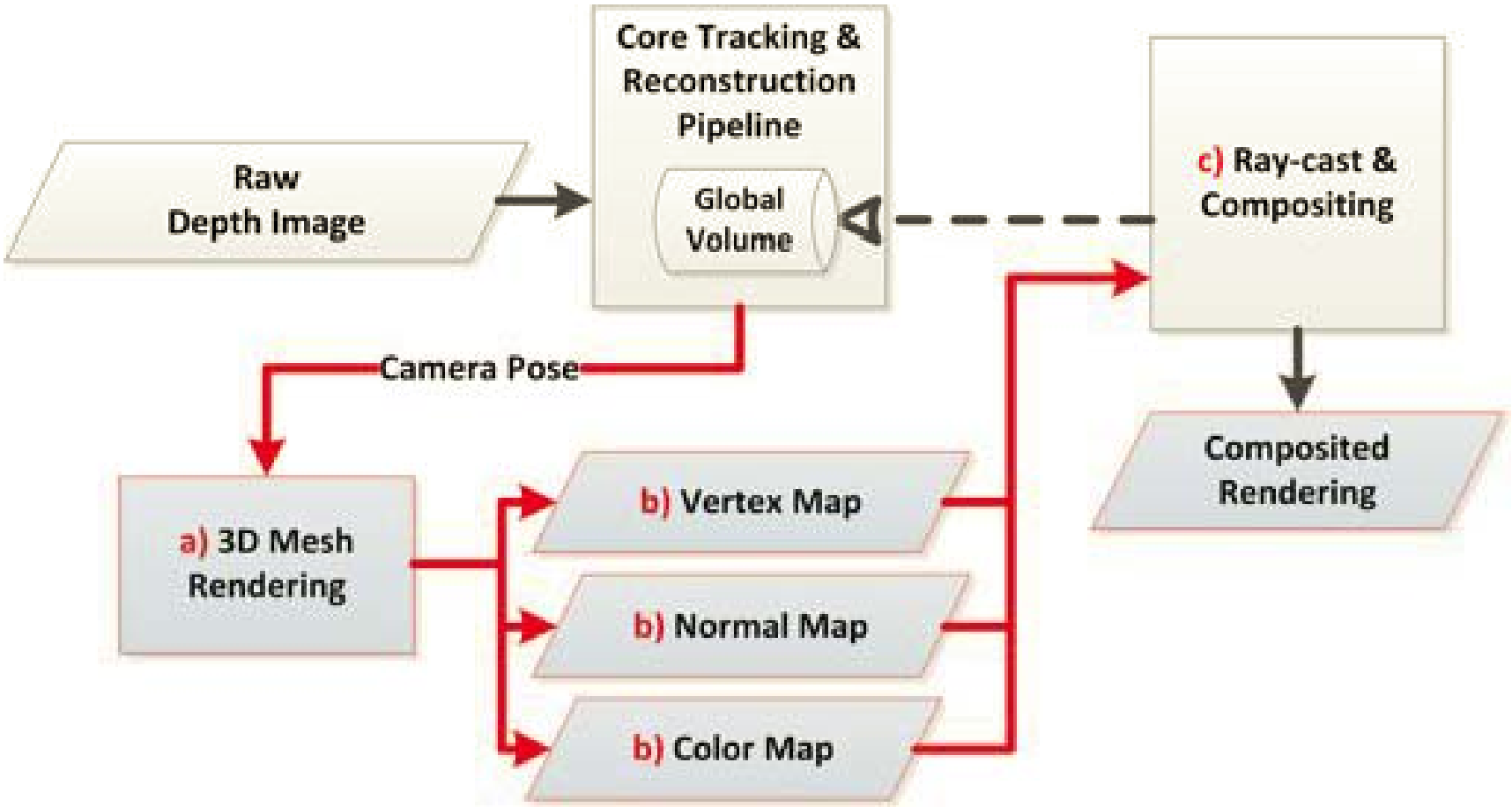
Kinect Fusion [Izadi et al. 2011]



Position: tri-linear interpolated grid position

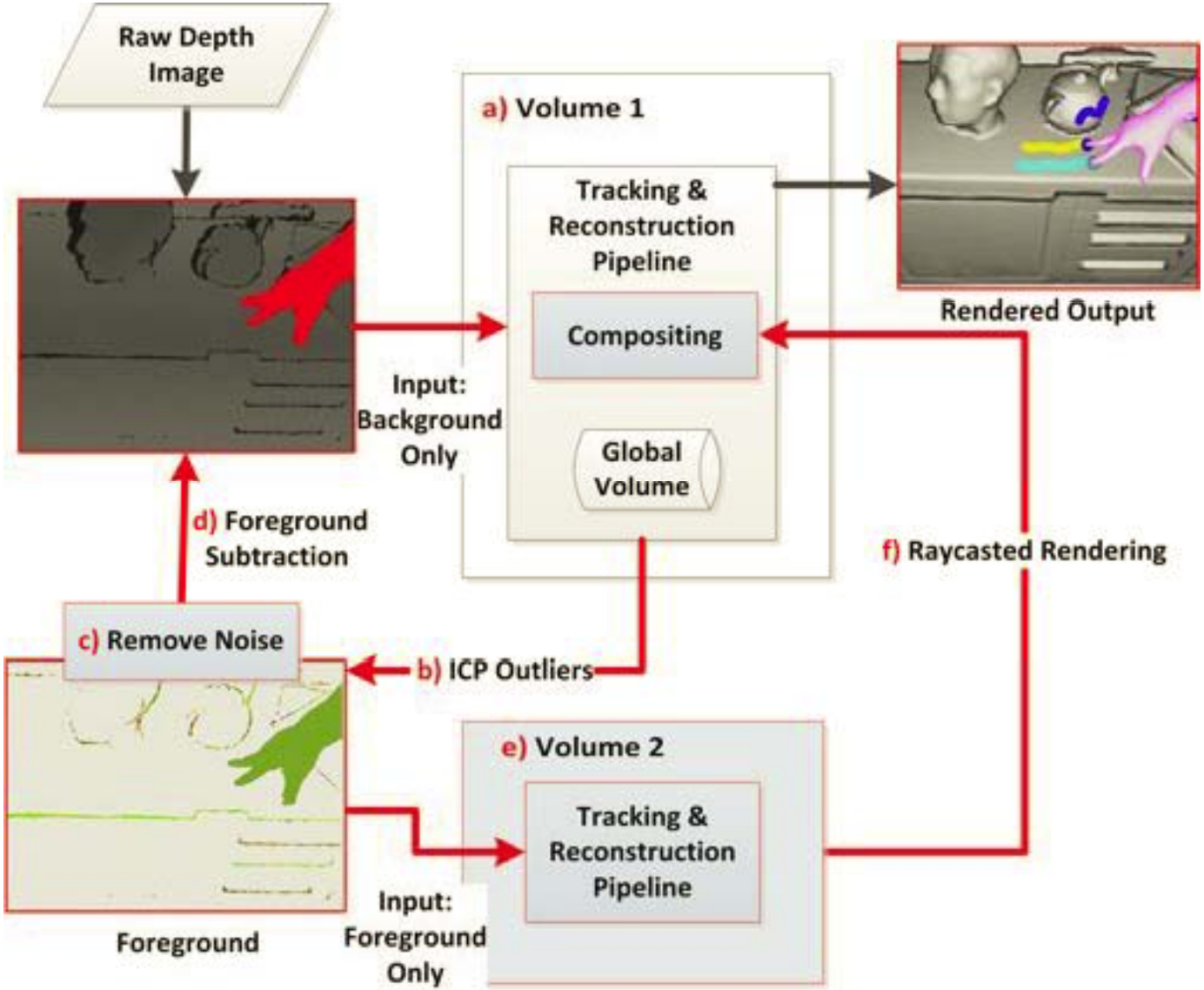
Normal: $\nabla \mathrm{sdf}(\mathrm{p})$

Kinect Fusion [Izadi et al. 2011]

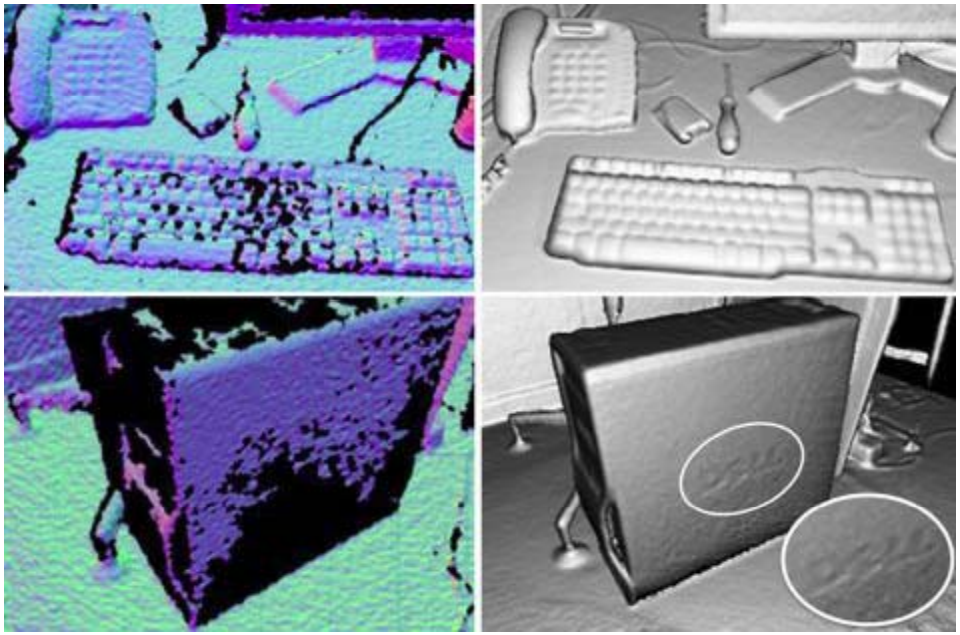


Kinect Fusion [Izadi et al. 2011]

Kinect Fusion [Izadi et al. 2011]

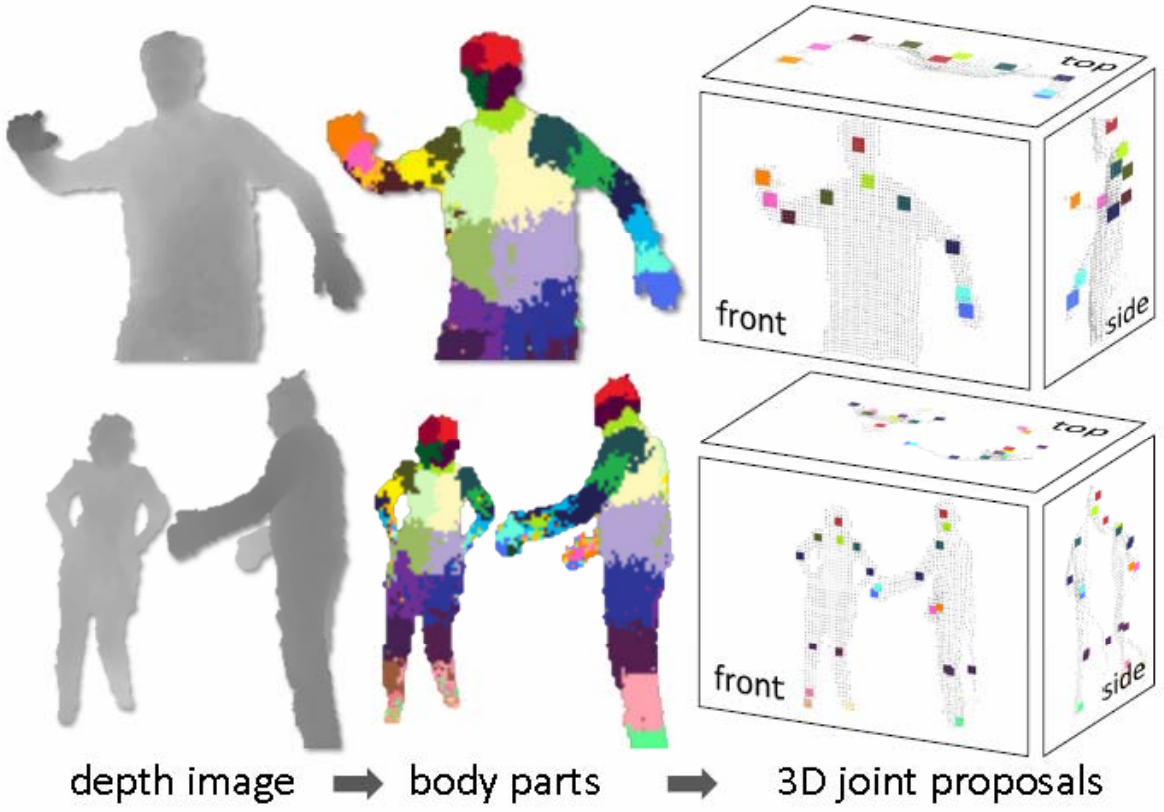


Kinect Fusion [Izadi et al. 2011]



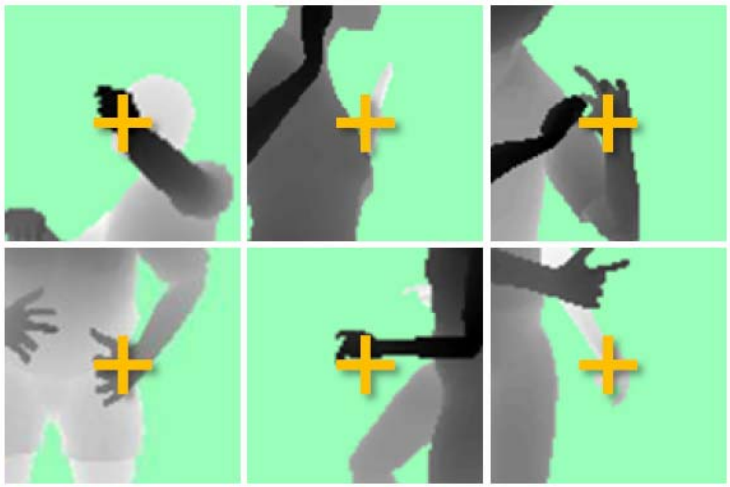
Skeleton tracking [Shotton et al., 2011]

- What the Kinect is mainly used for
- Adapts idea from object recognition with parts



Skeleton tracking [Shotton et al., 2011]

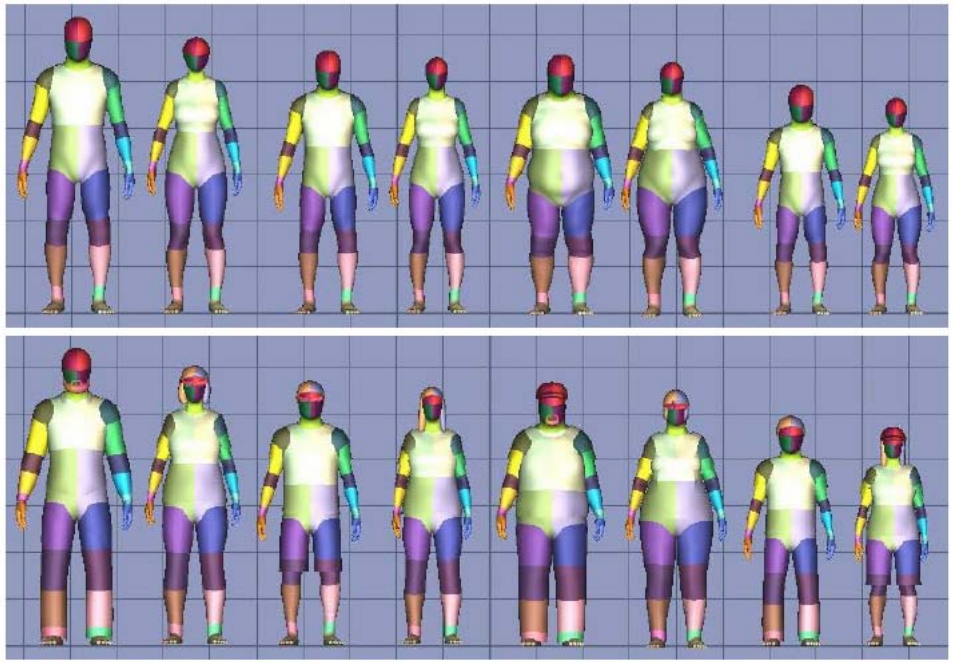
Skeleton tracking [Shotton et al., 2011]



- Independent solution
  - Per pixel classification
  - Per frame classification
- Training data
  - Synthetic depth images from motion capture data
- Deep randomized decision forest, implemented with GPU (200 fps)
- Find joint proposal

Generating synthetic training data

- Motion capture data
  - Cover variety of poses (not motion)
  - Furthest neighbor clustering
- Generating synthetic data

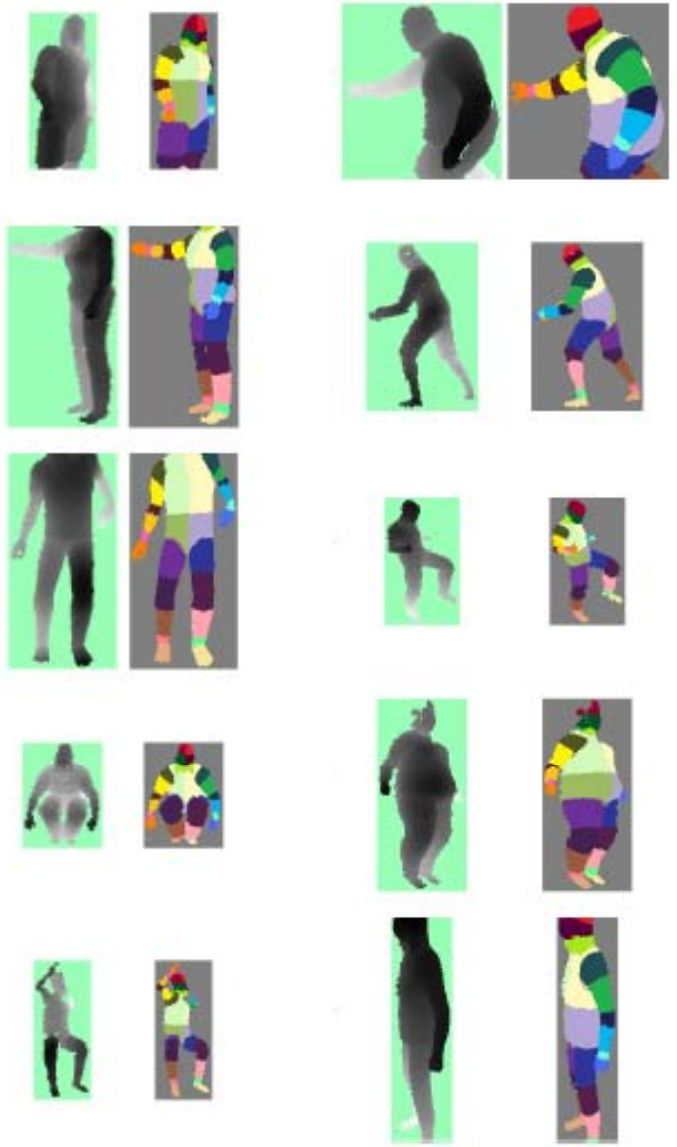


Base character

Skinning hair and clothing

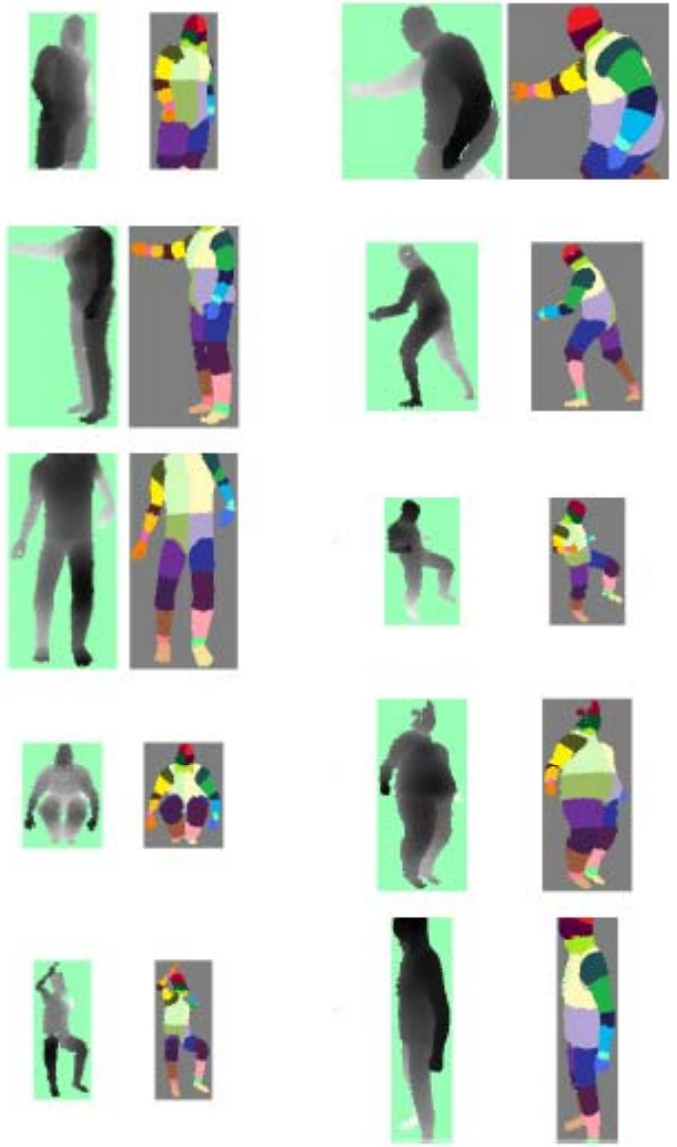
Generating synthetic training data

- 15 base characters
- Pose from motion capture data, mirroring with prob. 0.5
- Rotation and translation of character
- Hair and clothing
- Weight and height variation
- Camera position and orientation
- Camera noise



Body part labeling

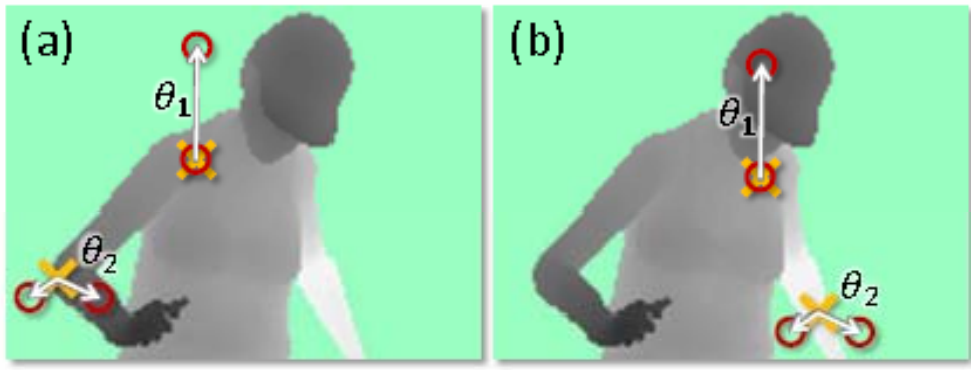
- Intermediate representation
  - Can readily be solved by efficient classification algorithms



Depth image features

- Notation
  - Depth of pixel $\mathbf{x}$ at image I $d_I(\mathbf{x})$
  - Parameters $\theta = (\mathbf{u}, \mathbf{v})$
- Depth image feature

$$f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right)$$



Data structure: 2.5 D

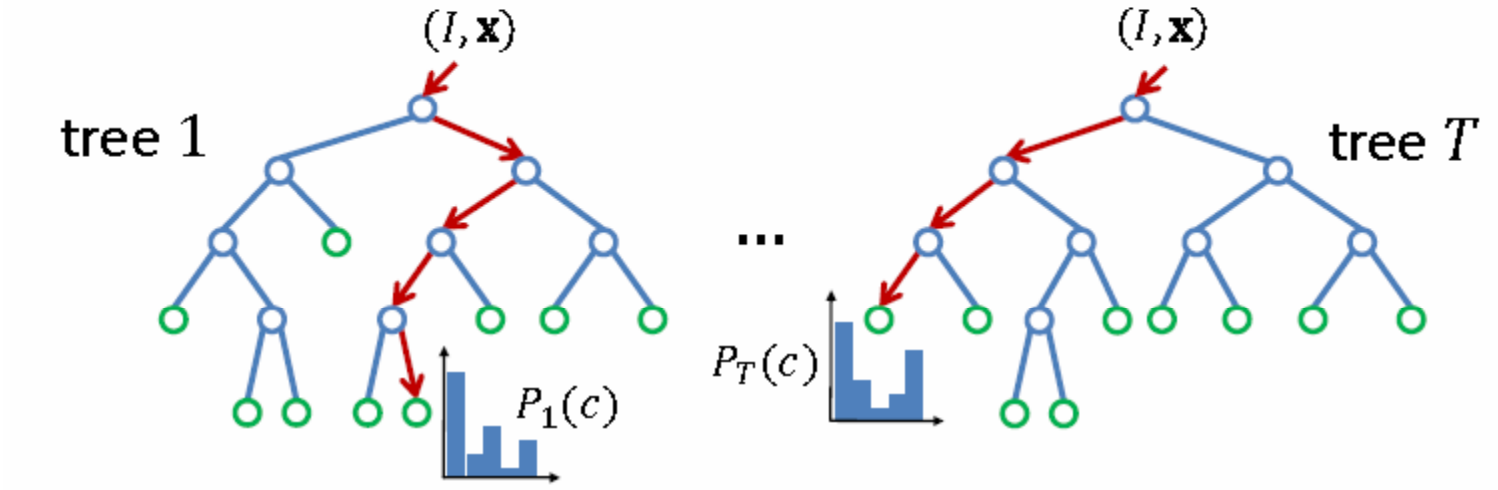
- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid

Data structure: 2.5 D

- Point cloud
- Geometry processing
  - Mesh
- Computer vision
  - Frames of 2-D grids
- Robotics
  - Ray-based model, voxel grid

Randomized decision forests

- An ensemble of T decision trees
- Split node has feature f_θ and threshold τ
- Leaf node has distribution over body part c



$$P(c|I,\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,\mathbf{x})$$

Training [Lepetit et al., 2005]

1. Randomly propose a set of splitting candidates $\phi = (\theta, \tau)$ (feature parameters θ and thresholds τ).
2. Partition the set of examples $Q = \{(I, \mathbf{x})\}$ into left and right subsets by each ϕ:

$$Q_{\mathrm{l}}(\phi) = \{ (I, \mathbf{x}) \mid f_\theta(I, \mathbf{x}) < \tau \} \quad (3)$$

$$Q_{\mathrm{r}}(\phi) = Q \setminus Q_{\mathrm{l}}(\phi) \quad (4)$$

3. Compute the ϕ giving the largest gain in information:

$$\phi^\star = \operatorname*{argmax}_\phi G(\phi) \quad (5)$$

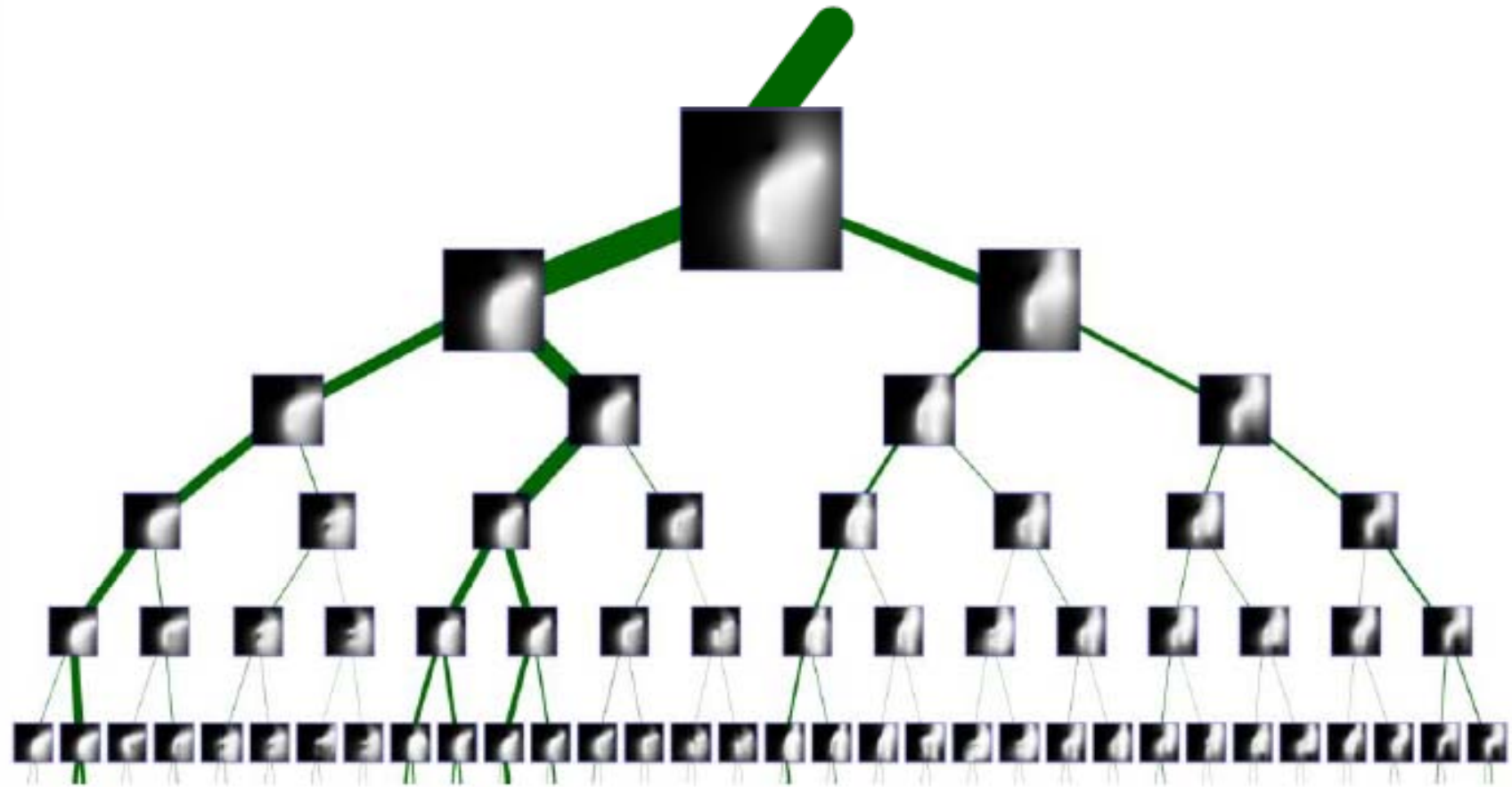
$$G(\phi) = H(Q) - \sum_{s \in \{\mathrm{l},\mathrm{r}\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)) \quad (6)$$

where Shannon entropy $H(Q)$ is computed on the normalized histogram of body part labels $l_I(\mathbf{x})$ for all $(I, \mathbf{x}) \in Q$.

4. If the largest gain $G(\phi^\star)$ is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets $Q_{\mathrm{l}}(\phi^\star)$ and $Q_{\mathrm{r}}(\phi^\star)$.

Training 3 trees to depth 20 from 1 million images takes about a day on a 1000 core cluster

Randomized decision forests



Joint position proposals

- Mean shift with a weighted Gaussian kernel

$$f_c(\hat{\mathbf{x}}) \propto \sum_{i=1}^{N} w_{ic} \exp\left(-\left\|\frac{\hat{\mathbf{x}} - \hat{\mathbf{x}}_i}{b_c}\right\|^2\right)$$

$$w_{ic} = P(c|I, \mathbf{x}_i) \cdot d_I(\mathbf{x}_i)^2$$

- Pushed backwards

Classification accuracy

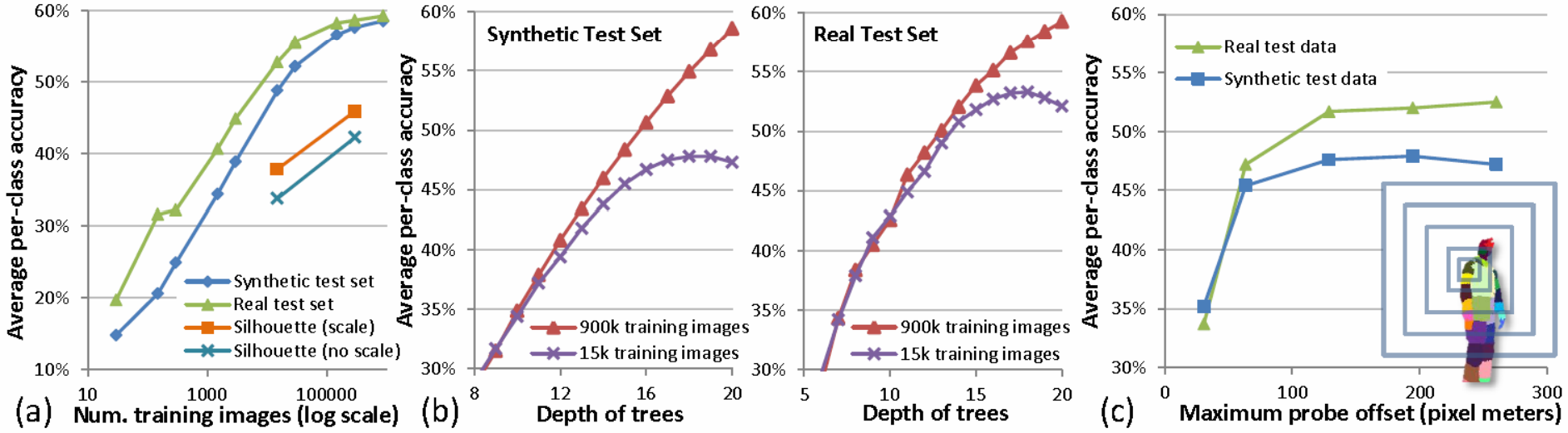


Figure 6. **Training parameters *vs.* classification accuracy.** (a) Number of training images. (b) Depth of trees. (c) Maximum probe offset.

Conclusion

- Kinect revolution
  - 3-D data is available to everyone
- Data structure: between 2-D and 3-D
  - RGB-D mapping
  - Floor plan generation
  - Kinect fusion
  - Skeleton tracking
  - …what else?

References

- RGB-D mapping
  - Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren and Dieter Fox, RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments., International Journal of Robotics Research (IJRR), 2012.
  - P. J. Besl and N. D. McKay. A method for registration of 3D shapes. IEEE Trans. Pattern Anal. Mach. Intell., 14:239–256, February 1992.
  - Martin A. Fischler and Robert C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. of the ACM* 24 (6): 381–395
  - Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard: Non-linear Constraint Network Optimization for Efficient Map Learning., IEEE Transactions on Intelligent Transportation Systems, Volume 10, Issue 3, Pages 428-439, 2009 (link)
  - Lourakis M and Argyros A (2009) SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software 36: 1–30.*
  - Pfister H, Zwicker M, van Baar J and Gross M (2000) Surfels: Surface elements as rendering primitives. In: SIGGRAPH 2000, Proceedings of the 27th Annual Conference on Computer Graphics, pp. 335–342.

References

- Interactive system
  - Michael Krainin, Brian Curless and Dieter Fox , Autonomous Generation of Complete 3D Object Models Using Next Best View Manipulation Planning, International Conference on Robotics and Automation, 2011.
  - Y.M. Kim, J. Dolson, M. Sokolsky, V. Koltun, S.Thrun, *Interactive Acquisition of Residential Floor Plans, IEEE International Conference on Robotics and Animation (ICRA), 2012*
- Kinect fusion
  - Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon, KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, ACM Symposium on User Interface Software and Technology, October 2011
  - B. Curless and M. Levoy. A volumetric method for building complex models from range images. ACM Trans. Graph., 1996
- Skeleton tracking
  - Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake, Real-Time Human Pose Recognition in Parts from a Single Depth Image, CVPR, June 2011
  - V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In Proc. CVPR, pages 2:775–781, 2005