Computer Vision Group
Prof. Daniel Cremers

Technische Universität München

Robotic 3D Vision

Lecture 16: 3D Object Detection 2 – 3D Keypoints, Iterative Closest Point

WS 2020/21

Dr. Niclas Zeller

Artisense GmbH

What We Will Cover Today

- 3D keypoint detectors and local descriptors

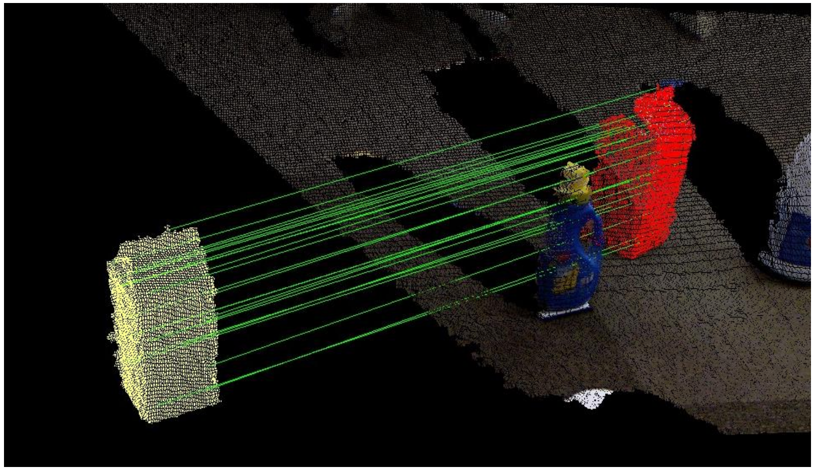
- Global 3D object descriptors

- Iterative closest points algorithm

3D Object Detection with Local Keypoints

- Detect and match a set of local keypoints between model and scene

- Locality of keypoints provides robustness against occlusions



- Local keypoints should be distinctive and repeatable, combined properties of detector and descriptor!

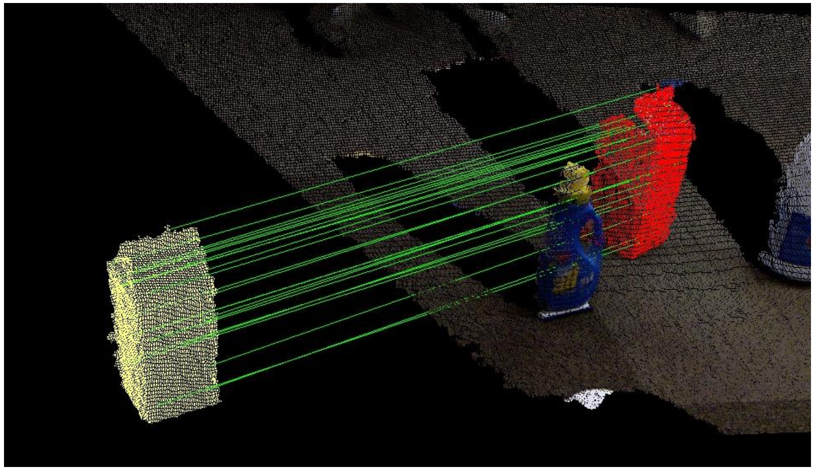
- Alignment for pose estimation:
  - 3D-to-3D alignment
  - Pose voting from keypoint match through local reference frames

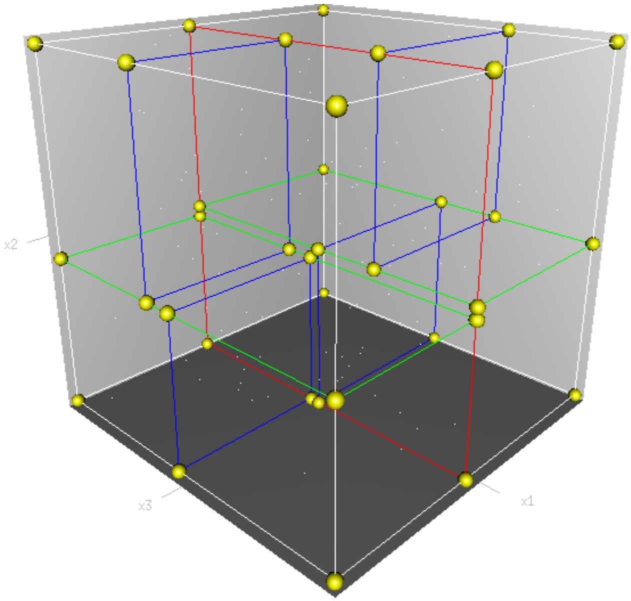
Image from pointclouds.org

3D Object Detection with Local Keypoints

- Render views of 3D CAD models and extract keypoints for rendered views

- Or Extract keypoints directly from 3D object models (f.e. CAD or scanned)
    - Rely only on geometry
    - Not on visual appreance





3D Keypoint Detectors

- Strategy 1: Uniform spatial sampling
- Strategy 2: Detection of keypoints at maxima of 3D interest measures
  - Intrinsic Shape Signatures (ISS) Detector, Zhong 2009
  - Harris3D
  - …
- Extraction of a local reference frame

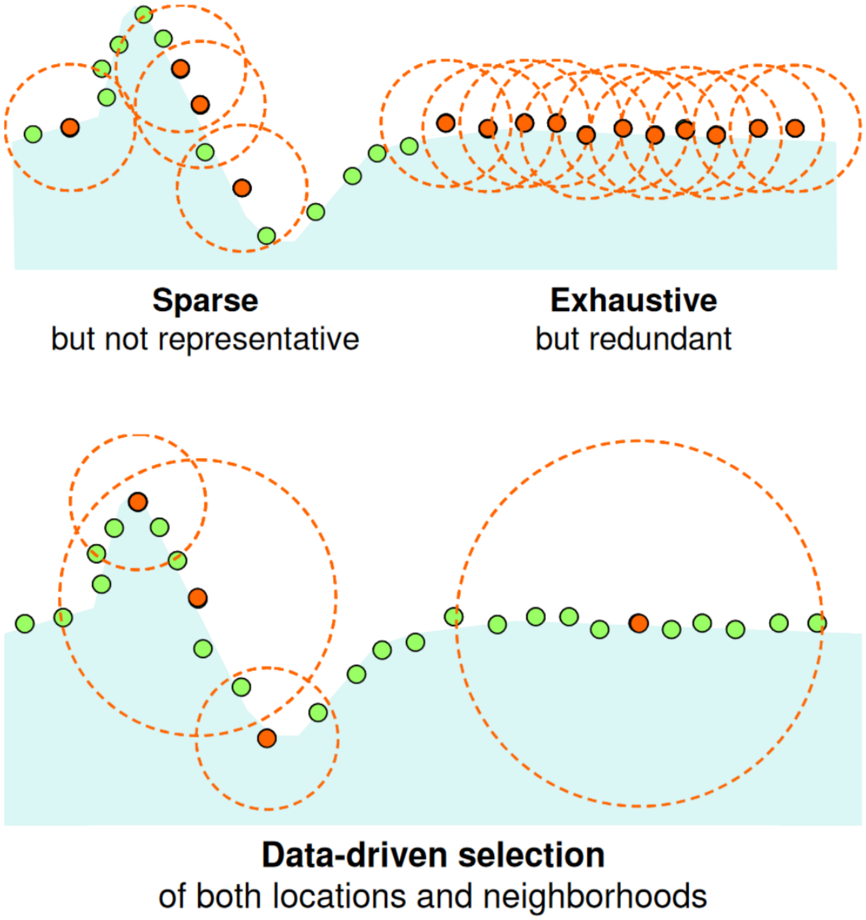
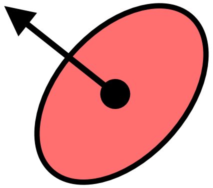

Sparse
but not representative
Exhaustive
but redundant
Data-driven selection
of both locations and neighborhoods

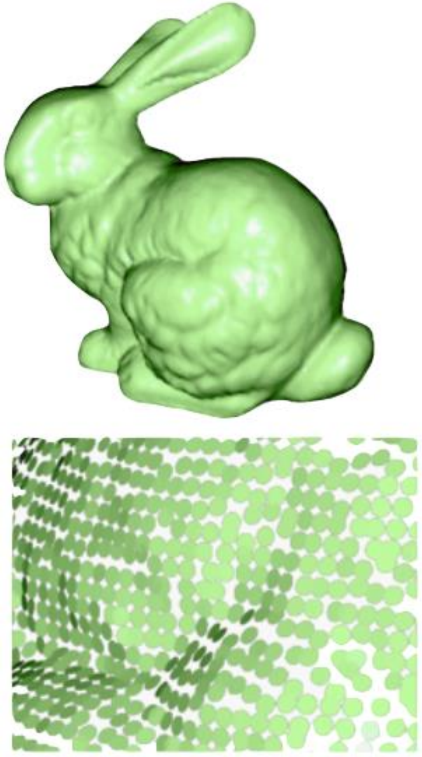
Image from Unnikrishnan and Hebert, 2008

3D Surface Representation

- 3D points in general represent object surface
  - 3D points don’t give insight on surface orientation and which points belong to the same surface

- Use surface elements (Surfels) to represent object surface
  - Point on a surface is defined by its
    - 3D localtion
    - Surface normal
    - Color
    - etc.





Pfister et al., SIGGRAPH 2000

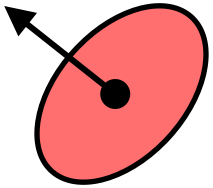
Image from Y. Sheikh

3D Surface Representation

- How to obtain surface normals for a set of 3D points

$$\bar{\mathbf{p}}_i = \frac{1}{N} \sum_{j:\left|\mathbf{p}_j - \mathbf{p}_i\right| < r} \mathbf{p}_j \qquad \text{with} \quad N = \left|j: \left|\mathbf{p}_j - \mathbf{p}_i\right| < r\right|$$

$$\boldsymbol{\Sigma}(\mathbf{p}_i) = \sum_{j:\left|\mathbf{p}_j - \mathbf{p}_i\right| < r} \left(\mathbf{p}_j - \bar{\mathbf{p}}_i\right)\left(\mathbf{p}_j - \bar{\mathbf{p}}_i\right)^T$$



  - Sometimes $\bar{\mathbf{p}}_i$ is replace by the point $\mathbf{p}_i$ itself

  - We obtain the surface normal as the eigenvector corresponding to the smallest eigenvalue of the covariance matrix $\boldsymbol{\Sigma}(\mathbf{p}_i)$

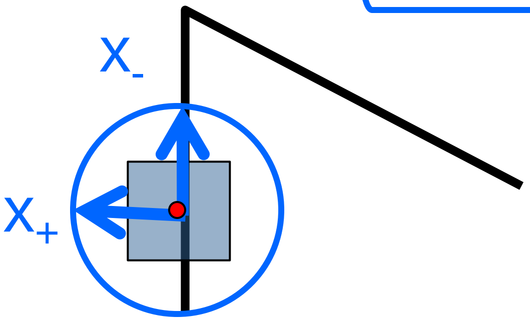
  - Unique direction can be obtained based on sensor view point for instance

Pfister et al., SIGGRAPH 2000

Recap: Structure Tensor

$$E(u,v) = [u \;\; v] \underbrace{\left(\sum_{(x,y)\in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \right)}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$

„structure tensor“



Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of largest increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of smallest increase in E.
- λ- = amount of increase in direction x_-

$$\begin{aligned} H x_+ &= \lambda_+ x_+ \\ H x_- &= \lambda_- x_- \end{aligned}$$

Slide adapted from Steve Seitz

Recap: Harris Operator

- “Harris operator” for corner detection

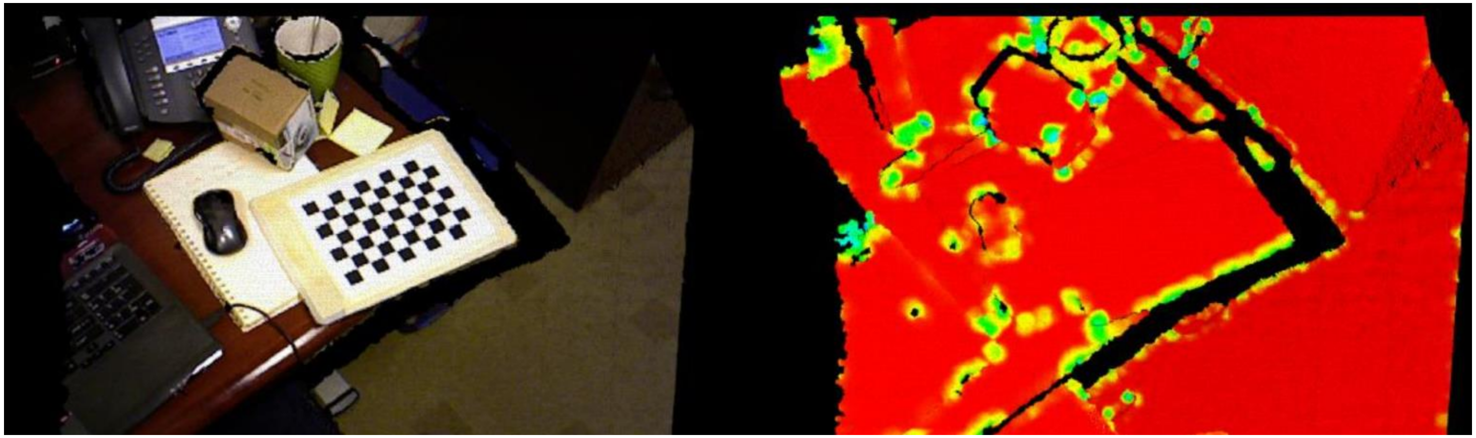
$$f = \frac{\lambda_- \lambda_+}{\lambda_- + \lambda_+}$$

$$= \frac{determinant(H)}{trace(H)}$$

- The trace is the sum of the diagonals, i.e., trace(H) = h_{11} + h_{22}
- Very similar to λ_- but less expensive (no square root)
- Called the “Harris Corner Detector” or “Harris Operator”
- Lots of other detectors, this is one of the most popular

Slide adapted from Steve Seitz

Harris3D



- Replace image gradients with surface normals

$$H = \sum_{i:\mathbf{p}_i \in W} \mathbf{n}_i \mathbf{n}_i^\top$$

W: 3D window, f.e. sphere

- Different response functions:

$$f = \det(H) - 0.04\,\mathrm{trace}(H)^2$$

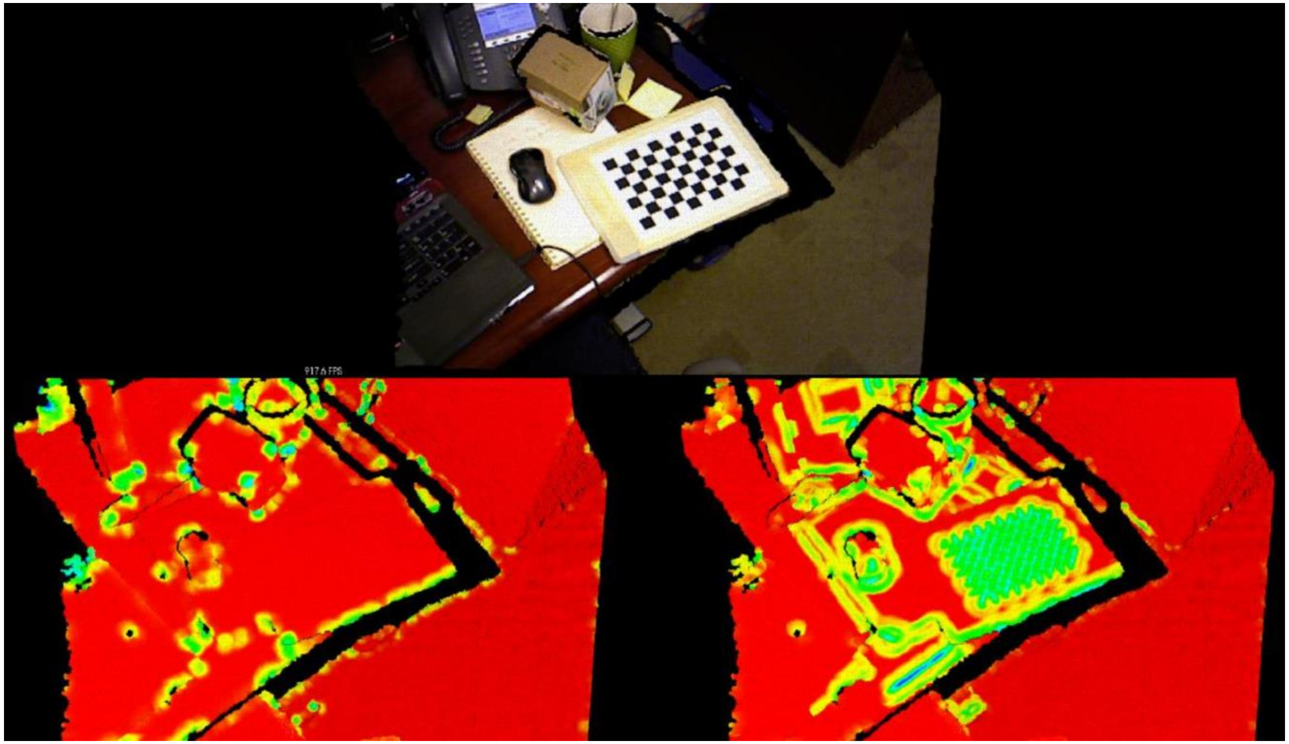
$$f = \det(H)/\,\mathrm{trace}(H)$$

$$f = \det(H)/\,\mathrm{trace}(H)^2$$

$$f = \lambda_{\mathrm{min}}$$

Image from S Gedikli and S Holzer

Harris5D



- Can be extended to combined use both color and geometry
- By stacking image gradients and normals

Image from S Gedikli and S Holzer

Intrinsic Shape Signatures (ISS) Detector

- Interest measure based on covariance of local point distribution

$$\mathbf{\Sigma}(\mathbf{p}_i) = \frac{1}{\sum_{j:|\mathbf{p}_j - \mathbf{p}_i| < r} w_j} \qquad j \neq i$$

$$\sum_{j:|\mathbf{p}_j - \mathbf{p}_i| < r} w_j (\mathbf{p}_i - \mathbf{p}_j)(\mathbf{p}_i - \mathbf{p}_j)^\top$$

- Weights account for varying point density

$$w_i := \frac{1}{|j : |\mathbf{p}_j - \mathbf{p}_i| < r_d|}$$

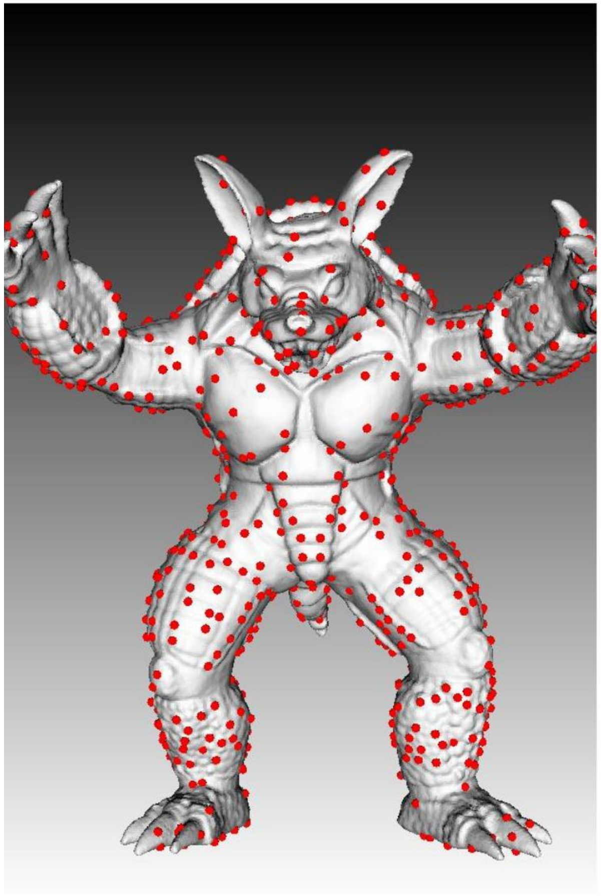
- Compute eigenvalues of local covariance

$$\lambda_1 > \lambda_2 > \lambda_3$$

- Find local maxima of smallest eigenvalue
  - Constrain by thresholds

$$\frac{\lambda_2}{\lambda_1} < \gamma_{21} \qquad \frac{\lambda_3}{\lambda_2} < \gamma_{32}$$

  - to find points with well conditioned eigen vector directions



Y. Zhong, Intrinsic shape signatures: A shape descriptor for 3D object recognition, 2009

Image from F. Tombari

Local Reference Frame

- Extract local reference frames from eigen vectors to align rotation-variant descriptor
  - Similar to orientation of 2D image keypoint
- 4 possible cases for right-handed

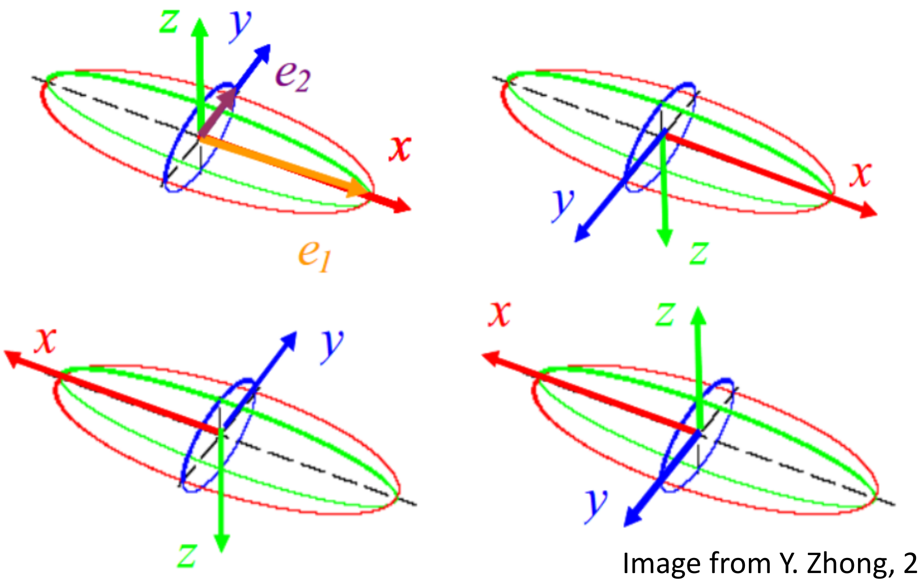


Image from Y. Zhong, 2009

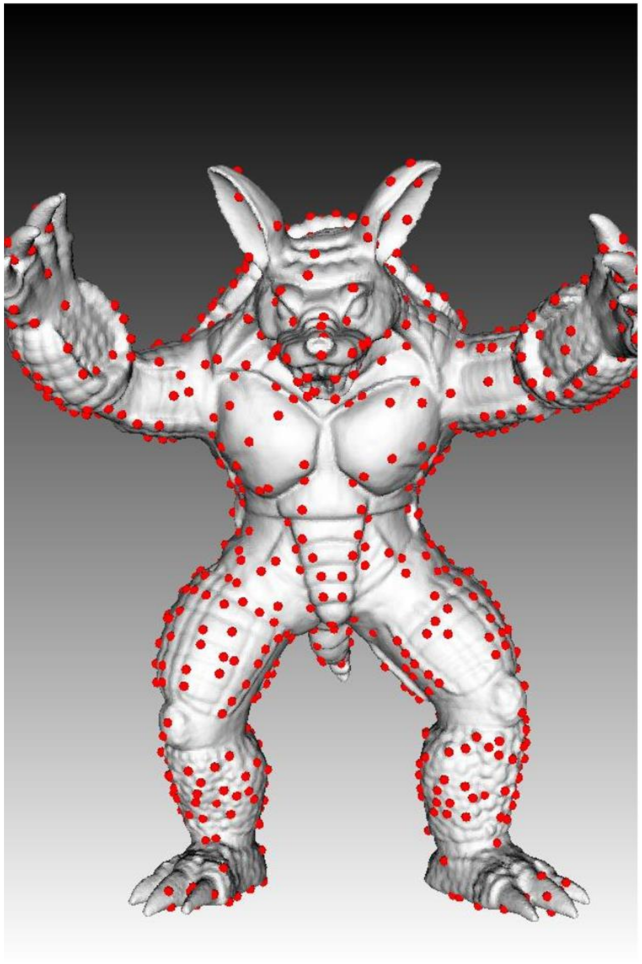


Image from F. Tombari

Local Reference Frame: Disambiguation

- Disambiguate the 4 possible cases by quantifying the support of the directions
- Directions and opposite directions of eigenvectors:

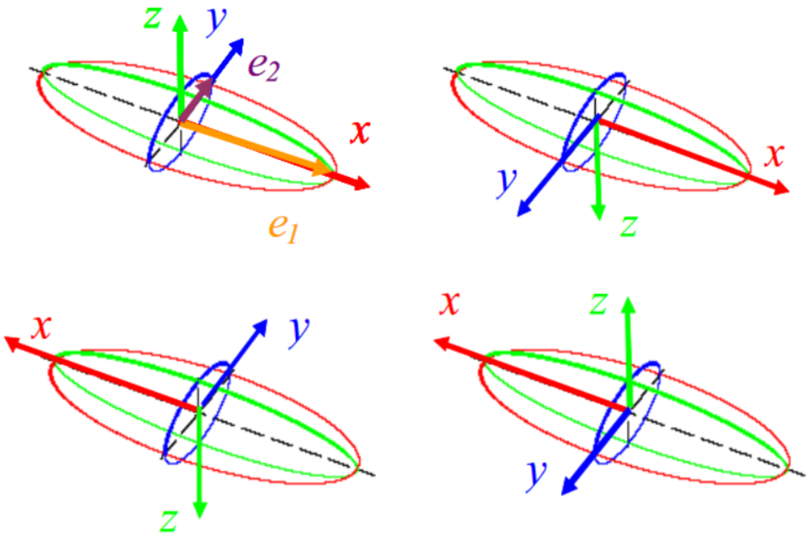
$$\mathbf{x}^+, \mathbf{y}^+, \mathbf{z}^+ \qquad\qquad \mathbf{x}^-, \mathbf{y}^-, \mathbf{z}^-$$

- Choose x-axis according to strongest support

$$S_x^+ \doteq \{i : d_i \le R \wedge (\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{x}^+ \ge 0\}$$

$$S_x^- \doteq \{i : d_i \le R \wedge (\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{x}^- > 0\}$$

$$\mathbf{x} = \begin{cases} \mathbf{x}^+, & |S_x^+| \ge |S_x^-| \\ \mathbf{x}^-, & \text{otherwise} \end{cases}$$



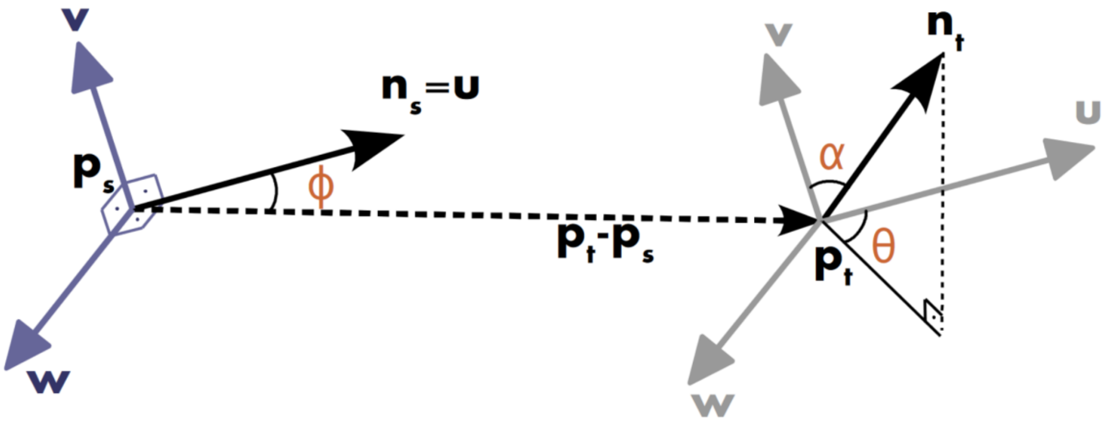
- z-direction analogously, y through $\mathbf{z} \times \mathbf{x}$

Image from Y. Zhong, 2009

3D Keypoint Descriptors

- Typical approach: Describe local distribution of points and/or surface normals
- How to achieve rotation invariance?
- Popular descriptors:
  - Fast Point Feature Histograms (FPFH)
  - Signature of Histograms of Orientations (SHOT)
  - …

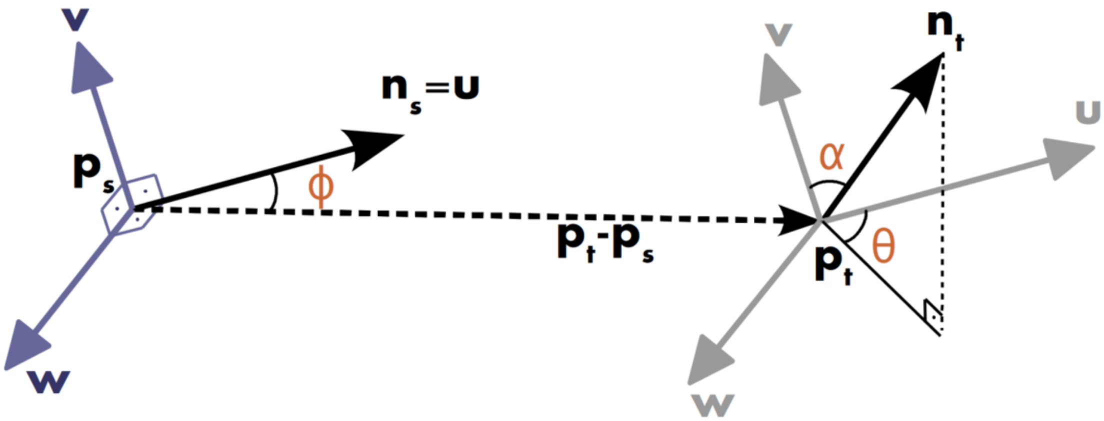
Surfel-Pair Relations



- Define descriptor based on relationship between surfels
- Surfel $(\mathbf{p}, \mathbf{n}) \in \mathbb{R}^3 \times \mathbb{R}^3$: point $\mathbf{p}$ with normal $\mathbf{n}$
- Surfel-pair:
  - Source: $(\mathbf{p}_s, \mathbf{n}_s)$
  - Target: $(\mathbf{p}_t, \mathbf{n}_t)$

Image from R Rusu, Diss. 2009

Surfel-Pair Relations



$$\mathbf{u} = \mathbf{n}_s$$

$$\mathbf{v} = \frac{(\mathbf{p}_t - \mathbf{p}_s)}{\|\mathbf{p}_t - \mathbf{p}_s\|_2} \times \mathbf{u}$$

$$\mathbf{w} = \mathbf{u} \times \mathbf{v}$$

- Features: geometric relations between two surfels

$$f_1 = \mathbf{v}^\mathrm{T}\mathbf{n}_t \qquad f_3 = \mathrm{atan2}(\mathbf{w}^\mathrm{T}\mathbf{n}_t)$$

$$f_2 = \mathbf{u}^\mathrm{T} \qquad f_4 = \|\mathbf{p}_t - \mathbf{p}_s\|_2$$

- Construct repeatable local coordinate frame between surfels
- Compute 4 features from constructed frame, normal and point coordinates
- Rotation-invariant features!

Image from R Rusu, Diss. 2009

Point Feature Histogram (PFH)

- Describe local neighborhood of a point by histogram of surfel-pair relations
  - Neighborhood is defined by a certain radius r

- Calculate 4D histogram based on surfel-pair relation features
  - 4D histogram can be stacked in a 1D vector
  - f_4 heavily depends on local point density → omit f_4 e.g. if point density depends on sensor view point
    - RGB-D camera: density depends on depth (distance to sensor)

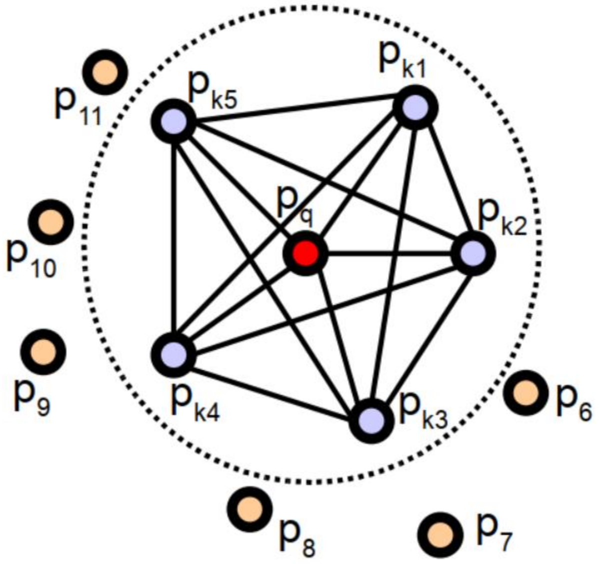


Image from R Rusu, Diss. 2009

Point Feature Histogram (PFH)

- Examples of Point feature Histograms
  - Similarity measure based on histogram intersection

$$d(PFH_1, PFH_2) = \sum_{i=1}^{N_{\text{bins}}} \min(PFH_1[i], PFH_2[i])$$

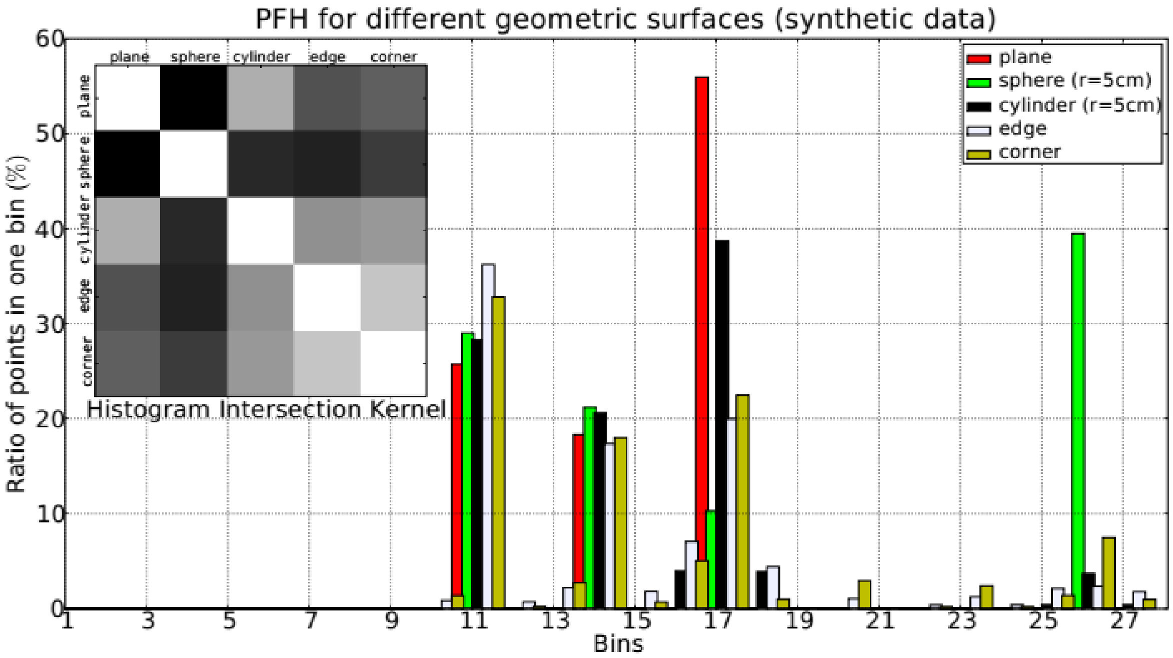
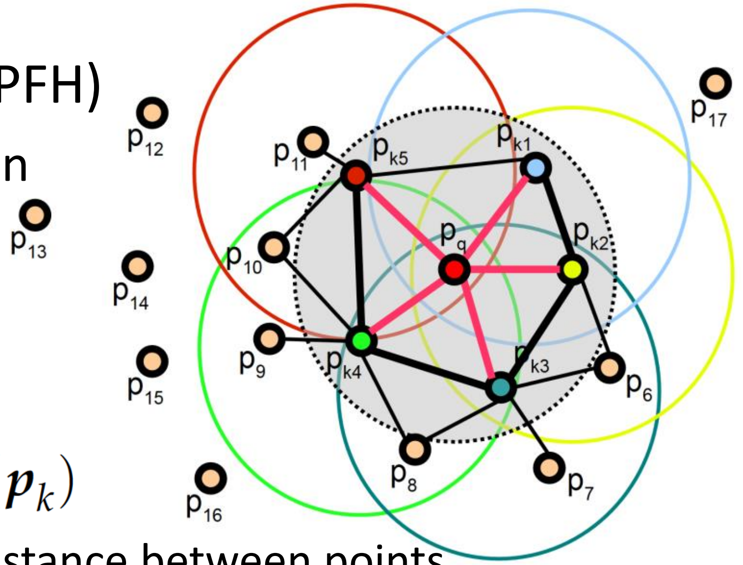


Image from R Rusu, Diss. 2009

Fast Point Feature Histogram (FPFH)

- PFH has complexity $O(nk^2)$
  - where n is the number of points in the point cloud and k is the number of considered neighbors for each point
- Fast Point Feature Histogram (FPFH)
  - Simplified Point Feature Histogram (SPFH)
    - Based on surfel-pair relations between point and its local neighbors
  - Accumulate SPFHs in local point neighborhood to obtain FPFH



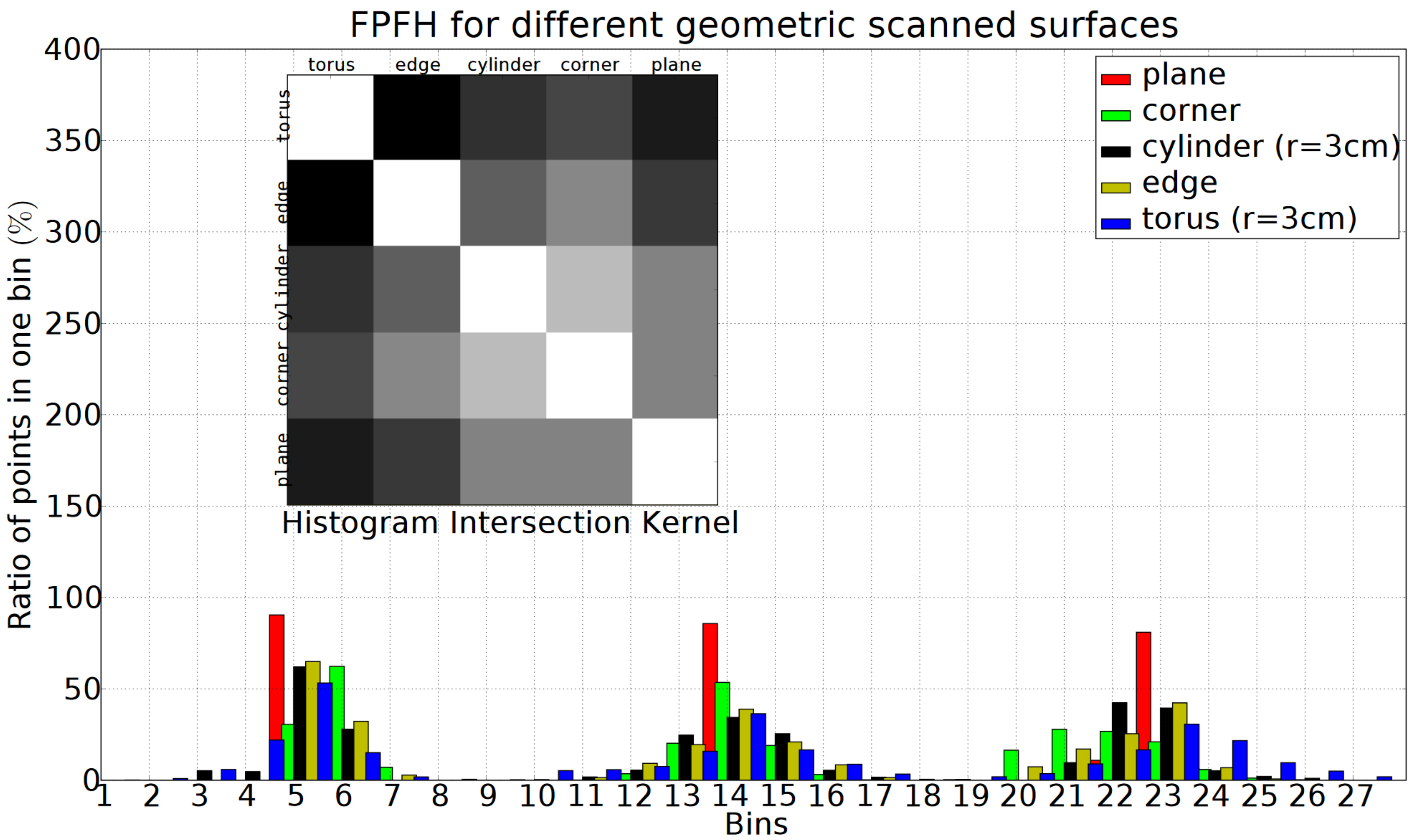
$$FPFH(\boldsymbol{p}_q) = SPFH(\boldsymbol{p}_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k}\cdot SPFH(\boldsymbol{p}_k)$$

Distance between points

  - Some relations are contributing twice
  - Additional relations are added

Image from R Rusu, Diss. 2009

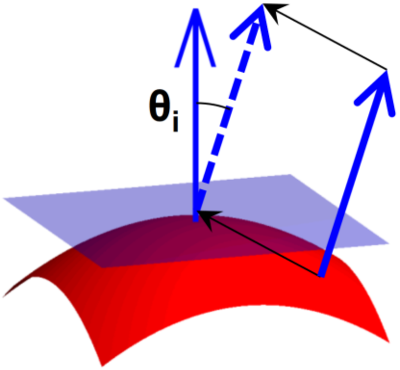
Fast Point Feature Histogram (FPFH)

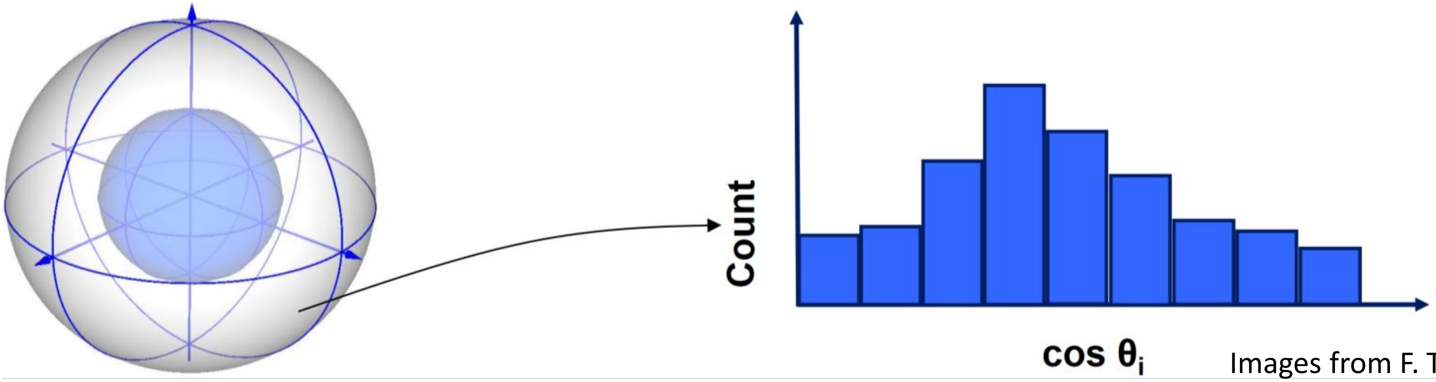


Images from R Rusu, Diss. 2009

Signature of Histograms of Orientations (SHOT)

- Describe spatial distribution of relative surface orientation around a keypoint
  - Discretize spherical volume around keypoint
  - Discretize spatial bins into angular bins
  - For each neighboring point, determine spatial bin and the angular bin for the angle between its surface normal and the normal of the keypoint
  - Align spherical grid with local reference frame to obtain rotation-invariance





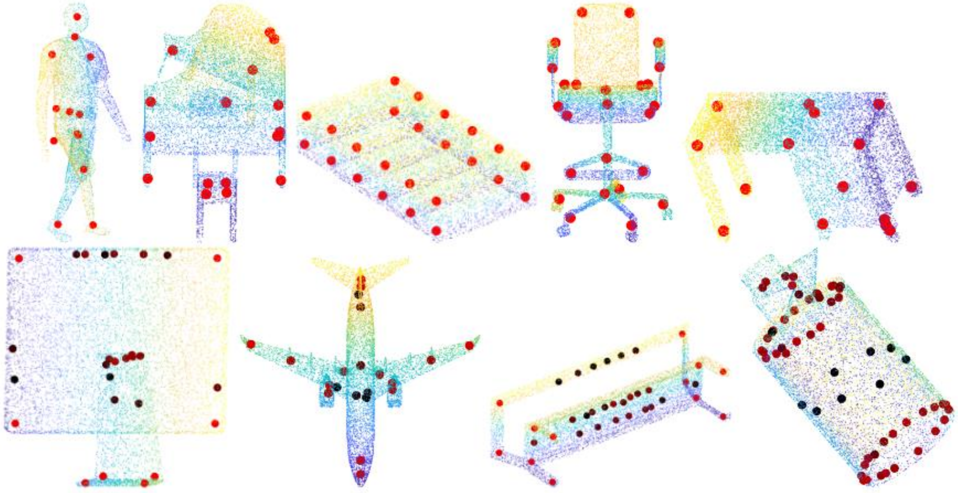
Images from F. Tombari

Deep Learning Based Features

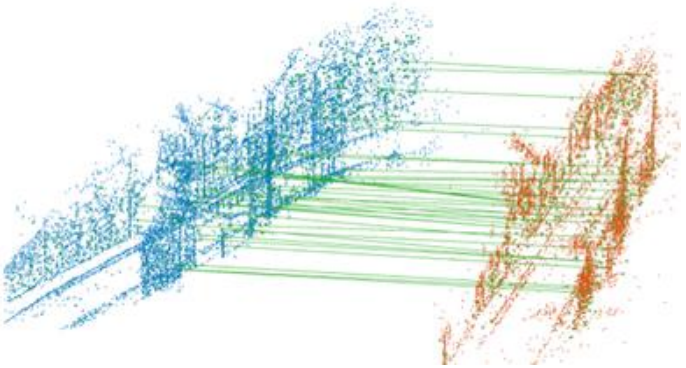
- Deep Learning based 3D detectors and descriptors did gain popularity in recent years
  - Challenge is the irregular structure of 3D point clouds
  - Initial approaches didn’t consider spatial neighborhood of points, e.g.
    - PointNet, CVPR 2017
  - Recent approaches try to mimic convolutions on point clouds based on local neighborhood operations, e.g.
    - Groh et al. Flex-Convolution, ACCV 2018
    - Li et al. PointCNN, NIPS 2018
    - …
  - In gereral require fixed size point cloud

Deep Learning Based Features

- Keypoint detector
  - USIP: Unsupervised Stable Interest Point Detection from 3D Point Clouds
    - Li et al., ICCV 2019

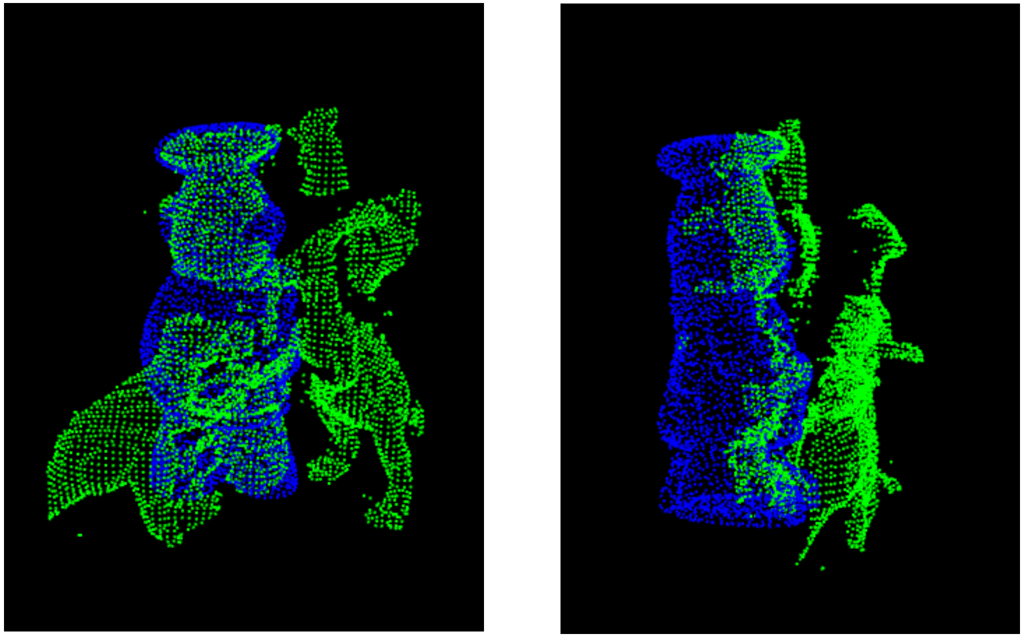


- Local 3D point descriptor
  - DH3D: Deep Hierarchical 3D Descriptors for Robust Large-Scale 6DoF Relocalization
    - Du et al., ECCV 2020



Pose Refinement

- So far, detection strategies provide only a coarse pose estimate
  - Based on keypoint associations (only subset of points)
- Popular strategy for pose refinement
  - Iterative Closest Points (ICP)
- Align scene measurements with model point cloud
  - Using all available points



Scene

Model

Images from pointclouds.org

Iterative Closest Points (ICP)

- Key Idea
  - If we knew the correspondences of points between scene and model, we could directly solve for the 3D-to-3D motion (rotation/translation) estimate

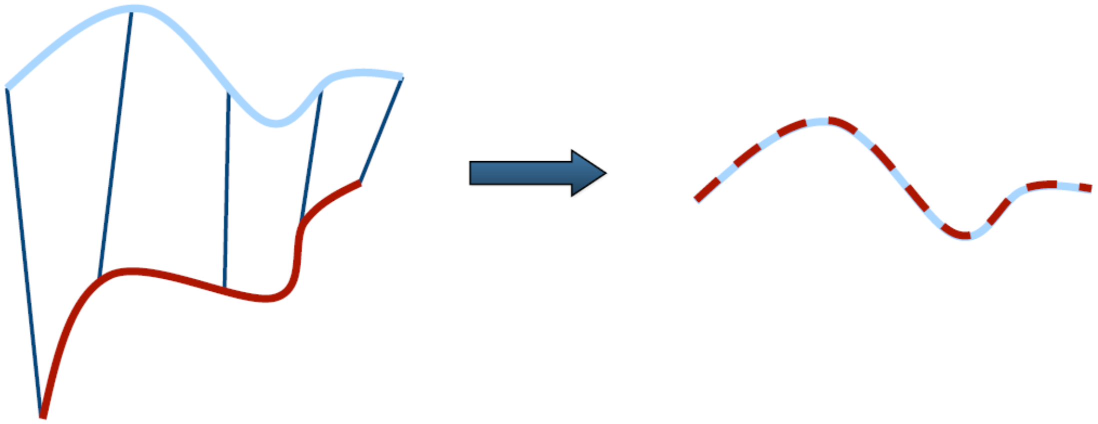


Image from Cyrill Stachniss

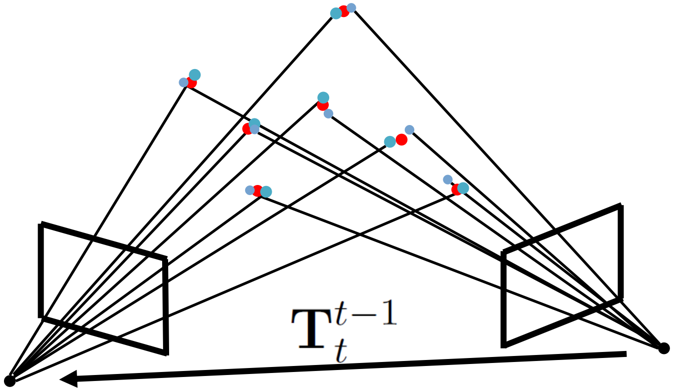
Recap: 3D-to-3D Motion Estimation

- Given corresponding 3D points in two camera frames

  $$\mathcal{X}_{t-1} = \{\mathbf{x}_{t-1,1}, \ldots, \mathbf{x}_{t-1,N}\}$$

  $$\mathcal{X}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,N}\}$$

  determine relative camera pose $\mathbf{T}_t^{t-1}$



- Idea: determine rigid transformation that aligns the 3D points
- Geometric least squares error: $E\left(\mathbf{T}_t^{t-1}\right) = \sum_{i=1}^{N} \left\| \overline{\mathbf{x}}_{t-1,i} - \mathbf{T}_t^{t-1} \overline{\mathbf{x}}_{t,i} \right\|_2^2$
- Closed-form solutions available, f.e. Arun et al., 1987
- Applicable e.g. to RGB-D cameras or also Lidar
  - Should only be used if we have very accurate depth

Recap: 3D Rigid-Body Motion from 3D-to-3D Matches

- Arun et al., Least-squares fitting of two 3-d point sets, IEEE PAMI, 1987
- Corresponding 3D points, $N \geq 3$

$$\mathcal{X}_{t-1} = \{\mathbf{x}_{t-1,1}, \ldots, \mathbf{x}_{t-1,N}\} \qquad \mathcal{X}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,N}\}$$

- Determine means of 3D point sets

$$\boldsymbol{\mu}_{t-1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{t-1,i} \qquad \boldsymbol{\mu}_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{t,i}$$

- Determine rotation from

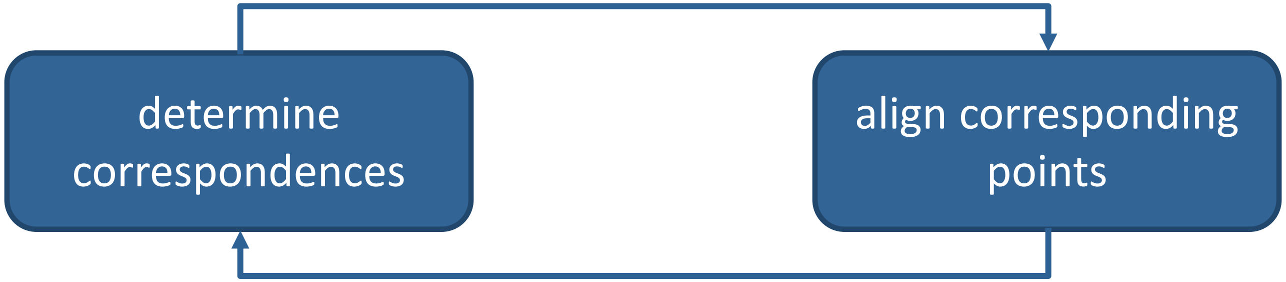
$$\mathbf{A} = \sum_{i=1}^{N} \left(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}\right) \left(\mathbf{x}_t - \boldsymbol{\mu}_t\right)^\top \qquad \mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top \qquad \mathbf{R}_{t-1}^t = \mathbf{V}\mathbf{U}^\top$$

- Determine translation as $\mathbf{t}_{t-1}^t = \boldsymbol{\mu}_t - \mathbf{R}_{t-1}^t \boldsymbol{\mu}_{t-1}$

Iterative Closest Points (ICP)

- If the correct correspondences are not known, it is generally impossible to determine the optimal relative motion (rotation/translation) in one step

- Idea: Iteratively and alternatingly estimate correspondences and pose alignment between point sets $P = \{\mathbf{p}_i\}_{i=1}^{N}$ and $Q = \{\mathbf{q}_j\}_{j=1}^{M}$



$\operatorname{argmin}_c p(P \mid Q, \boldsymbol{\xi}, c)$

$\operatorname{argmin}_{\boldsymbol{\xi}} p(P \mid Q, \boldsymbol{\xi}, c)$

Iterative Closest Points (ICP)

- Idea: Iteratively and alternatingly estimate correspondences and pose alignment between point sets $P = \{\mathbf{p}_i\}_{i=1}^{N}$ and $Q = \{\mathbf{q}_j\}_{j=1}^{M}$

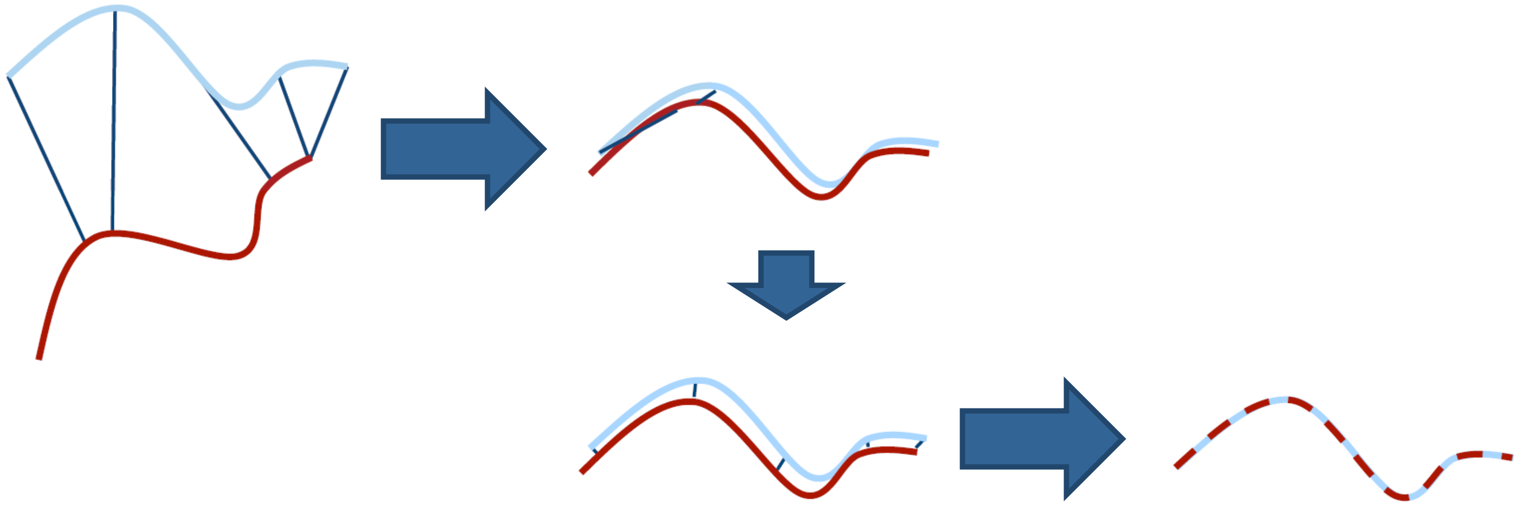
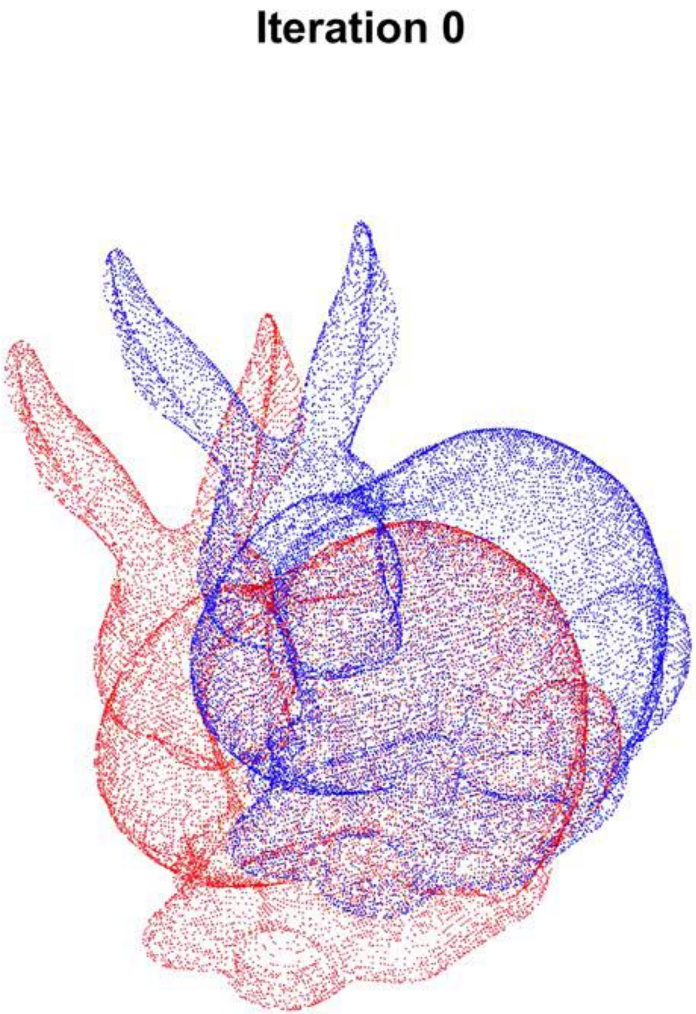


Image adapted from Cyrill Stachniss

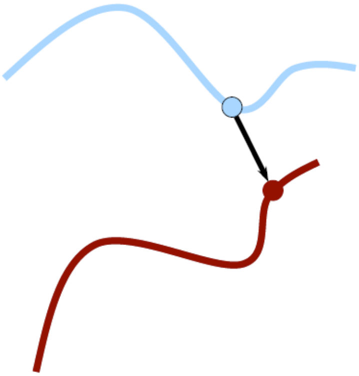
Keypoint Alignment and ICP Example



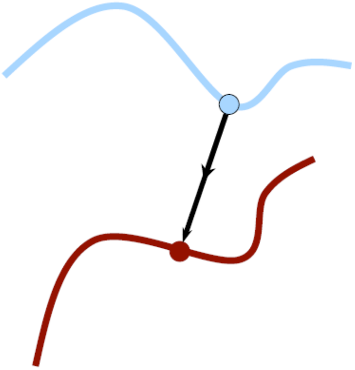
https://www.youtube.com/watch?v=uzOCS_gdZuM

Data Association for ICP

- Closest-points matching



- Normal shooting
  - Requires normal calculation
  - Better convergence than closest-point for smooth structures



Images from Cyrill Stachniss

Projective Data Association

- For aligning depth or point measurements from a sensor, we can use projective data association

- Warping of measured 3D point

- Analogous association as in direct image alignment!

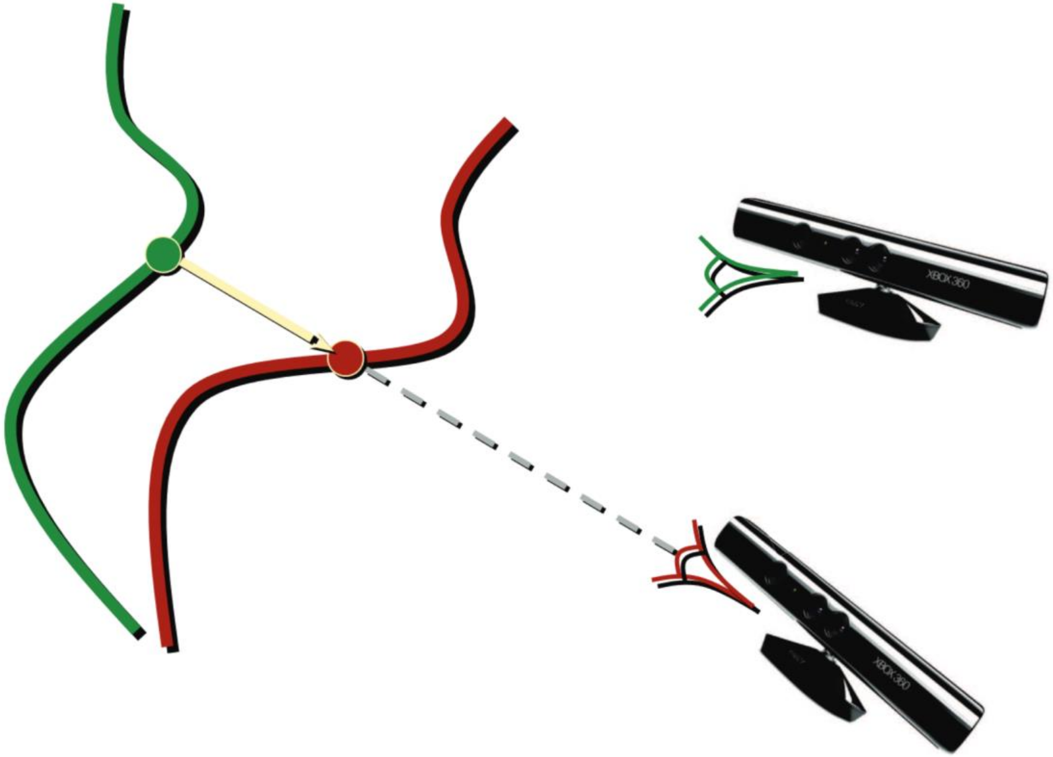
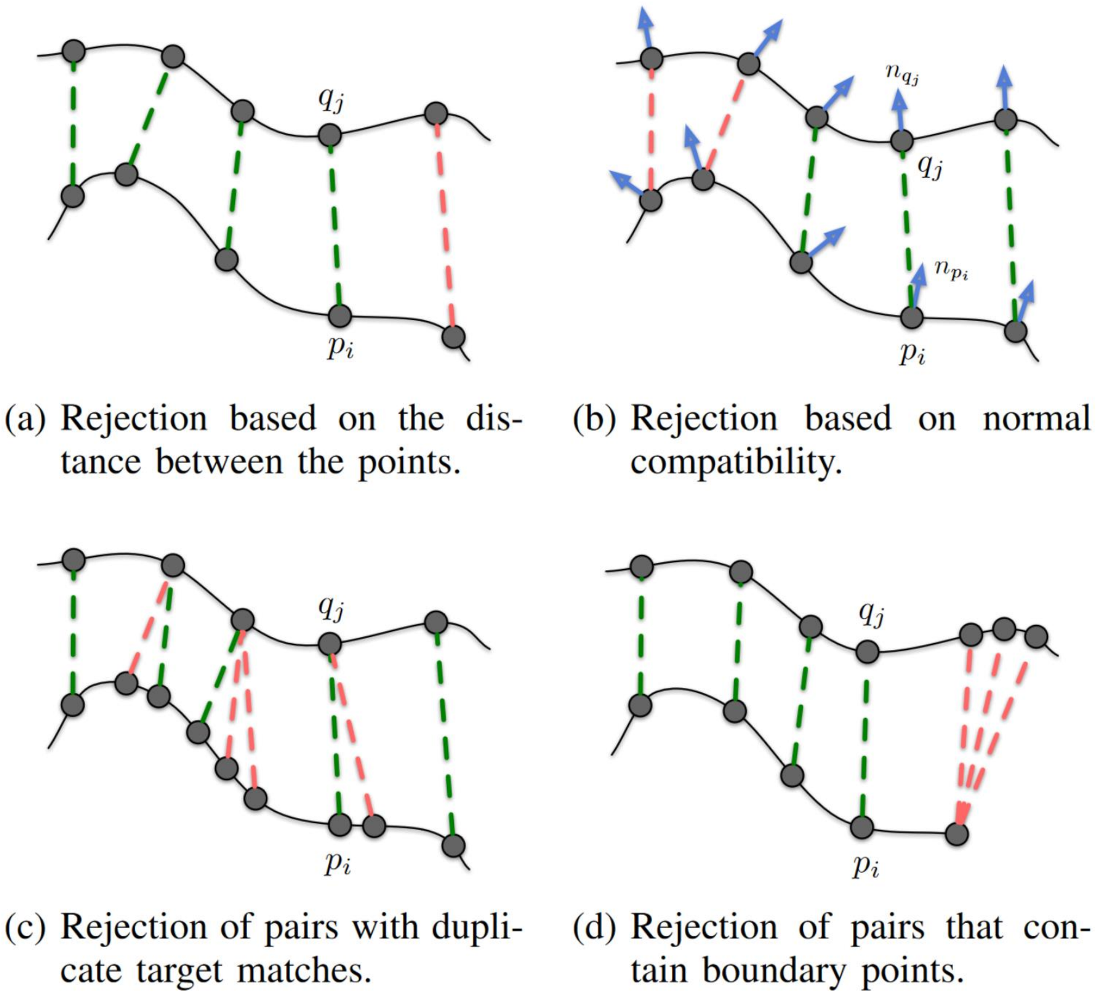


Image from R. Newcombe 2013

Outlier Rejection for ICP

- Optionally perform outlier rejection



(a) Rejection based on the distance between the points.

(b) Rejection based on normal compatibility.

(c) Rejection of pairs with duplicate target matches.

(d) Rejection of pairs that contain boundary points.

Images from Holz et al., 2015

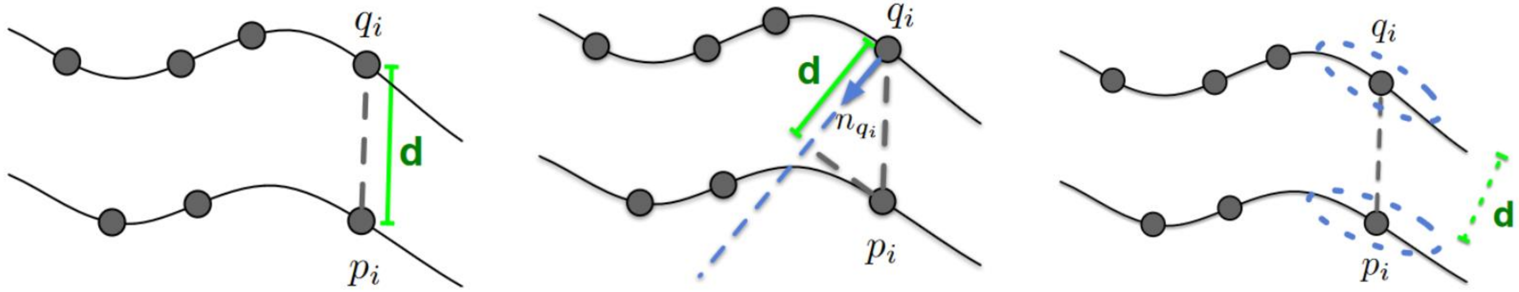
ICP Alignment Objectives

- Alignment objectives: point-point, point-plane, GICP

$$E_{\text{point-to-point}}\left(\boldsymbol{T}\right) = \sum_{k=1}^{N} w_k ||\boldsymbol{T}\,\boldsymbol{p}_k - \boldsymbol{q}_k||^2, \text{ and}$$

$$E_{\text{point-to-plane}}\left(\boldsymbol{T}\right) = \sum_{k=1}^{N} w_k \left(\left(\boldsymbol{T}\,\boldsymbol{p}_k - \boldsymbol{q}_k\right) \cdot \boldsymbol{n}_{\boldsymbol{q}_k}\right)^2$$

$$E_{\text{Generalized-ICP}}\left(\boldsymbol{T}\right) = \sum_{k=1}^{N} \boldsymbol{d}_k^{(\boldsymbol{T})^T} \left(\boldsymbol{\Sigma}_k^Q + \boldsymbol{T}\boldsymbol{\Sigma}_k^P \boldsymbol{T}^T\right)^{-1} \boldsymbol{d}_k^{(\boldsymbol{T})}$$

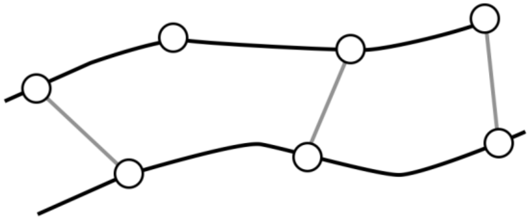


(a) Point to point error (b) Point to plane error (c) Generalized-ICP

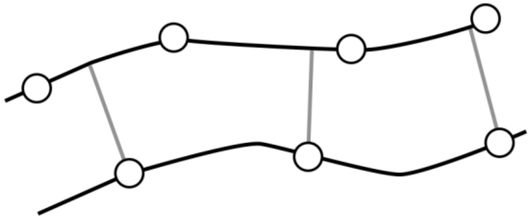
Images from Holz et al., 2015

ICP Alignment Objectives

- Point-to-Point vs. Point-to-plane
  - Requires normal calculation for one of the point clouds
  - Each iteration is generally slower than point-to-point version
  - However, often significantly better convergence rate
  - Using point-to-plane distance instead of point-to-point lets flat regions slide along each other



point-to-point



point-to-plane

Images from Cyrill Stachniss

ICP Alignment Objectives

- Generalized ICP
    - Probabilistic modelling of point clouds
    - Where to get covariance matrices from
        - directly available from sensor measurements
        - Can be estimated from point distribution
        - Covariance matrix needs to be calculated from both point clouds

Lessons Learned Today

- 3D object detection with local 3D keypoints
  - 3D keypoint detector derived from 2D detector, e.g. Harris3D
  - Intrinsic Shape Signatures detector: points at strong surface curvature
  - 3D keypoint description
    - Extraction of local 3D reference frame from point distribution
    - PFH, SHOT descriptors
- Iterative Closest Points algorithm for point cloud alignment

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).