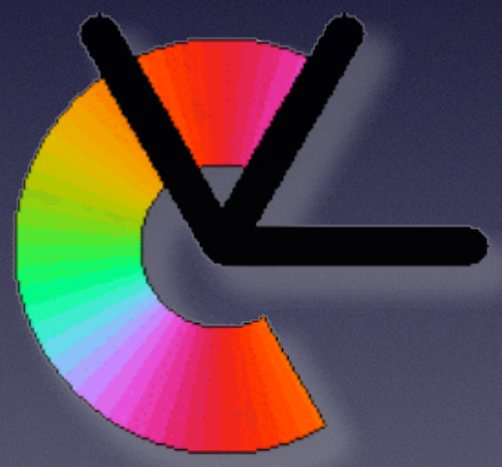
Range cameras II ++

More on point-set registration



Per-Erik Forssén
Division of Computer Vision
Department of Electrical Engineering
Linköping University

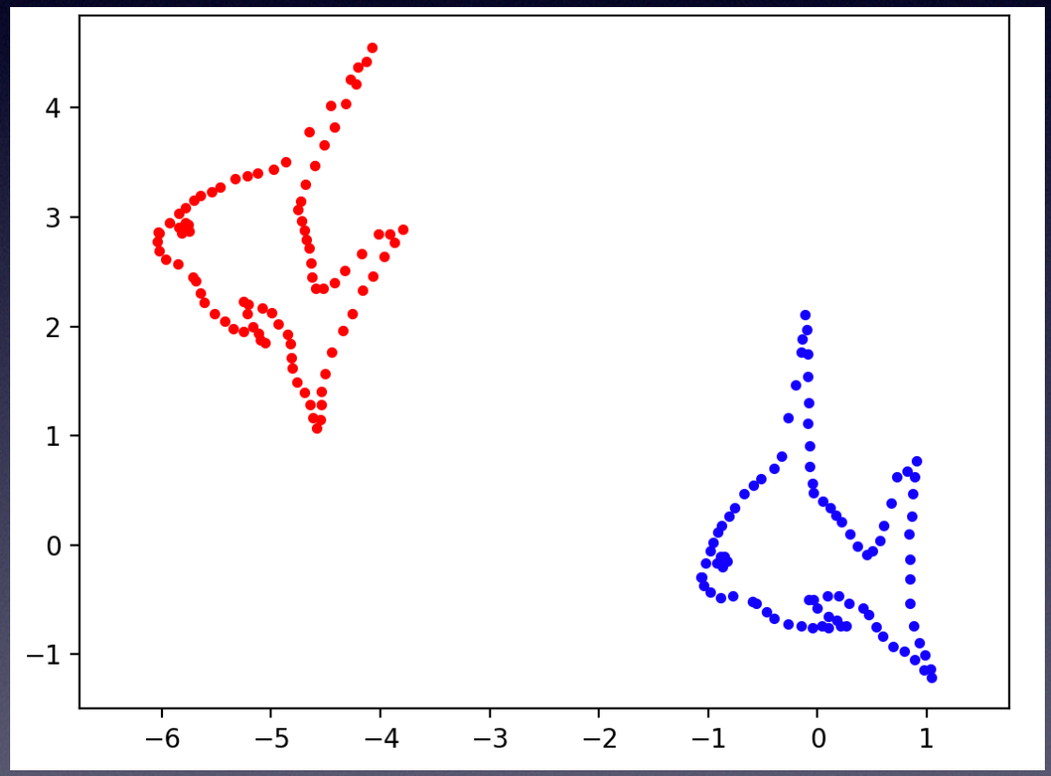
Point-set registration

- Recap of ***point-set registration:***

Goal: Find a transformation

$$\mathbf{Y}'_k = \mathbf{R}\mathbf{X}_k + \mathbf{t}$$

- That moves a set of points $\{\mathbf{X}_k\}_{k=1}^{K}$
- to be aligned with another set of points $\{\mathbf{Y}_l\}_{l=1}^{L}$

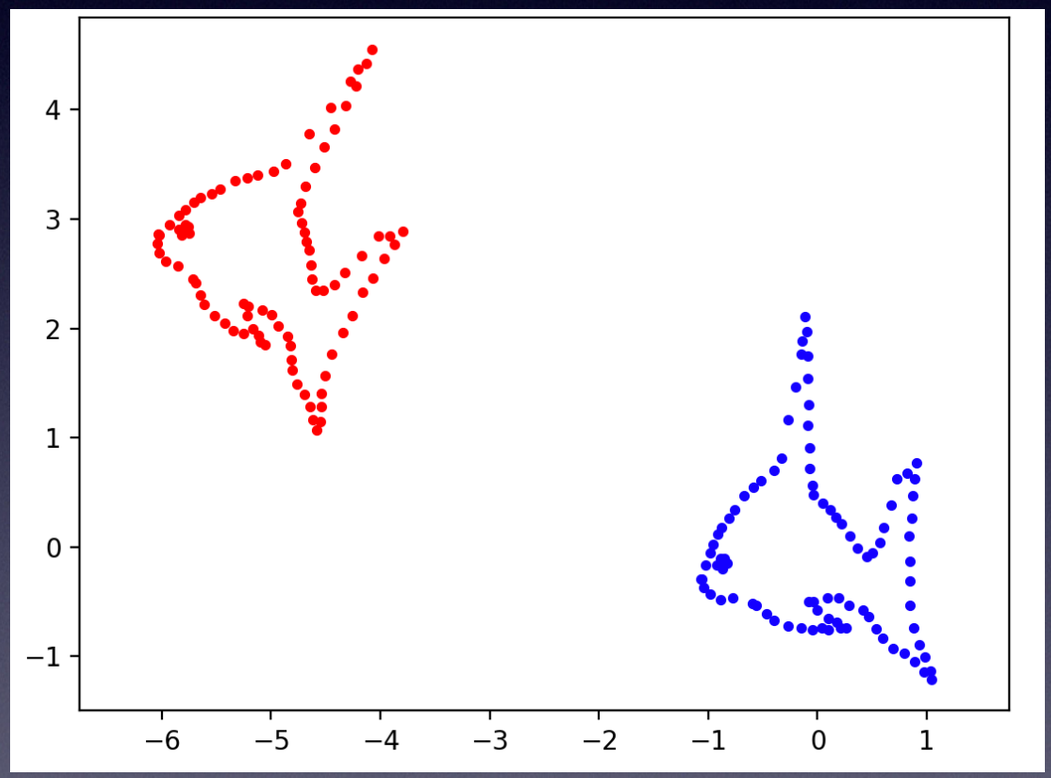


- ICP Classical method [Chen&Medioni ICRA91]

Point-set registration

- Iterated Closest Point (ICP) [Chen&Medioni ICRA91]

1. **Guess correspondence**
   For each point: find the closest point in the other set

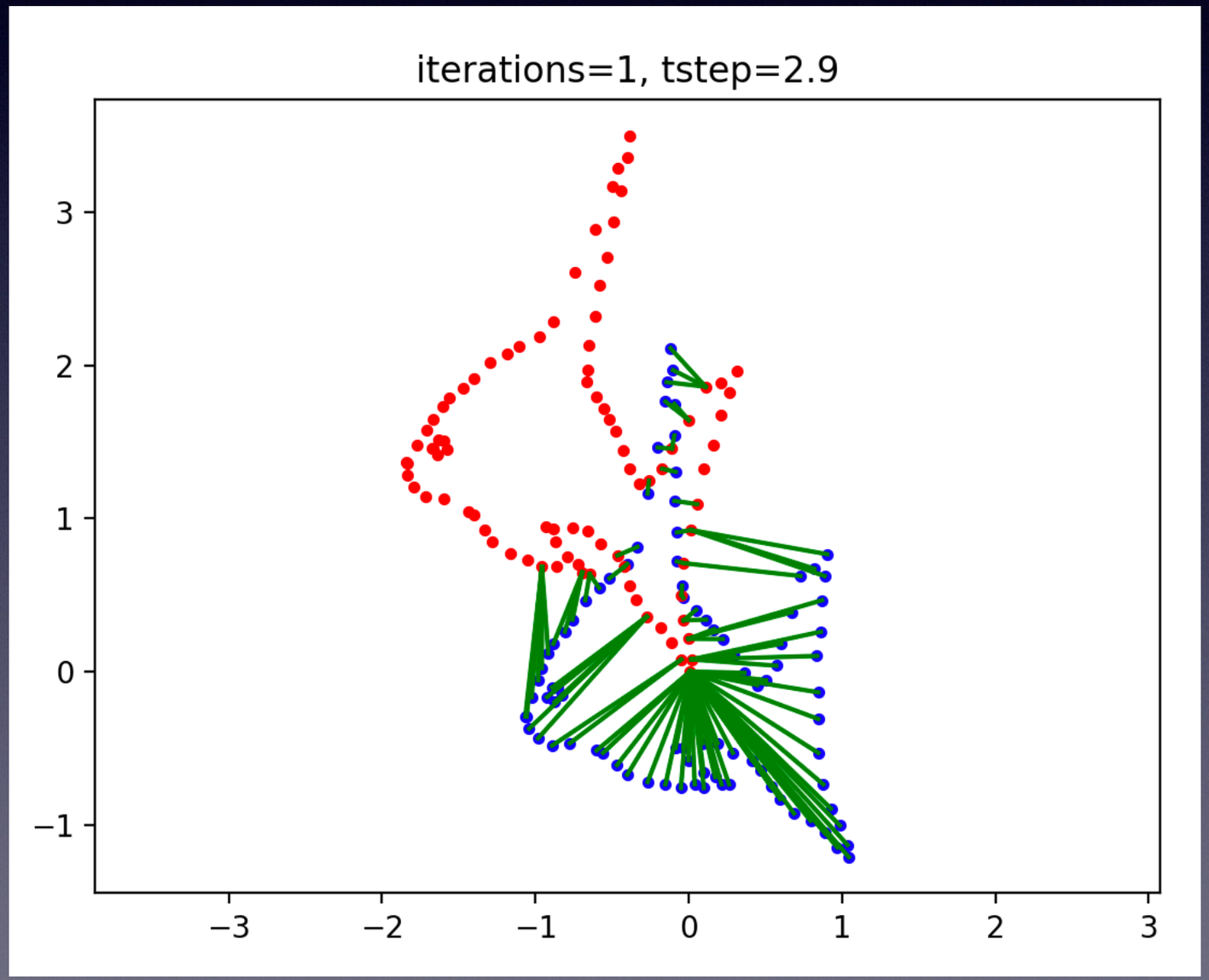


2. **Align correspondences**
   Center and solve the Orthogonal Procrustes Problem (OPP) to fint **R**, and **t**

3. **Apply transformation** and goto 1.

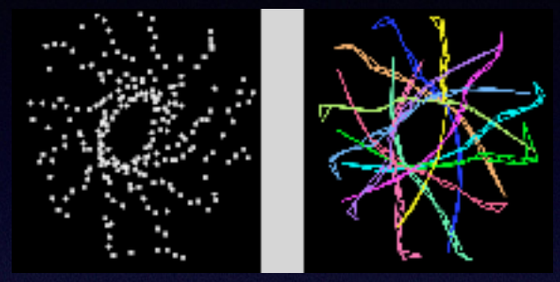
Point-set registration

- ICP demo 2D (now with more pauses)

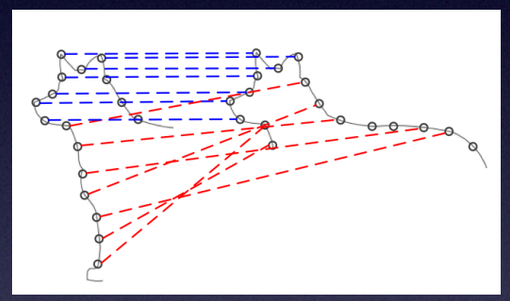


Point-set registration

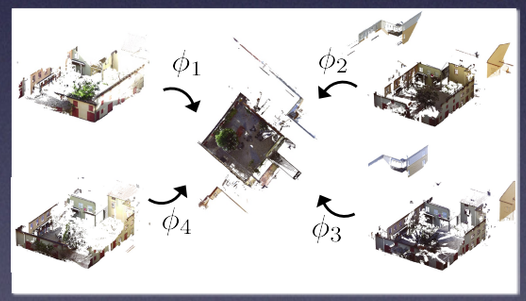
- **Real-time point set registration**
  KinectFusion and TSDF



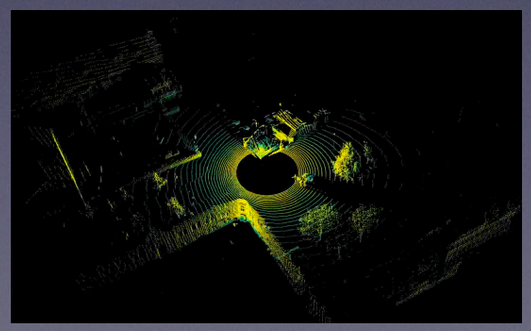
- **Feature based registration**
  FPFH, FGR and DGR



- **Multiple point-set registration**
  JRMPS, RLL

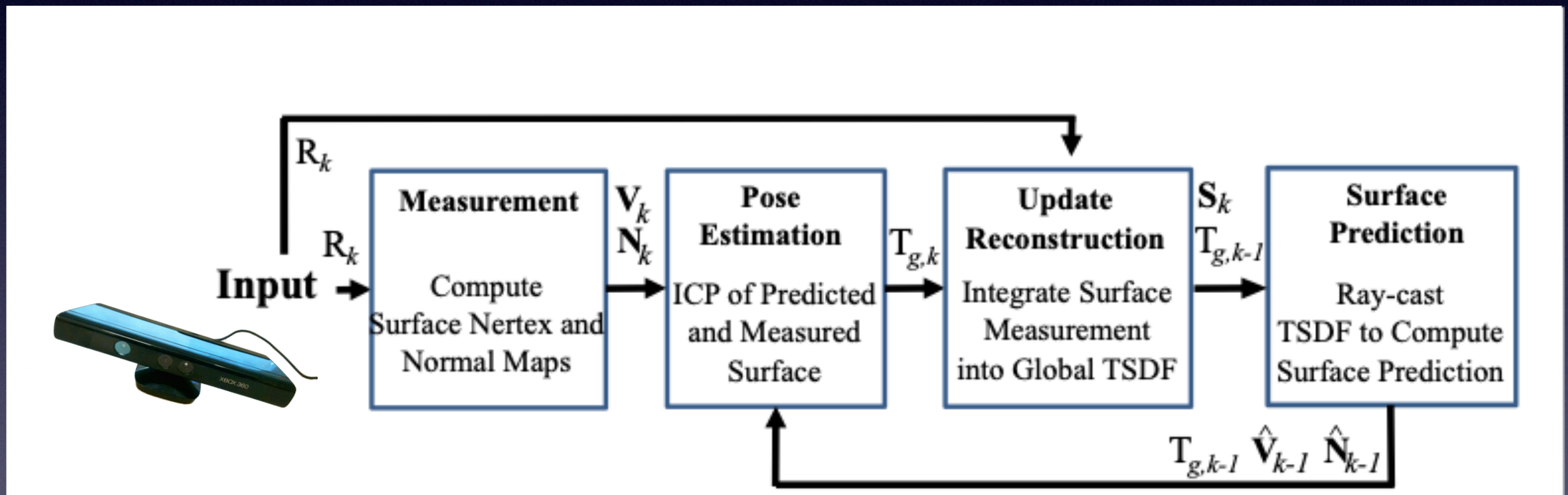


- **Point-cloud simplification**
  Farthest point sampling, PointNet++



Real-time point-set registration

R.A. Newcombe et al., *KinectFusion: Real-Time Dense Surface Mapping and Tracking*, **ISMAR'11**



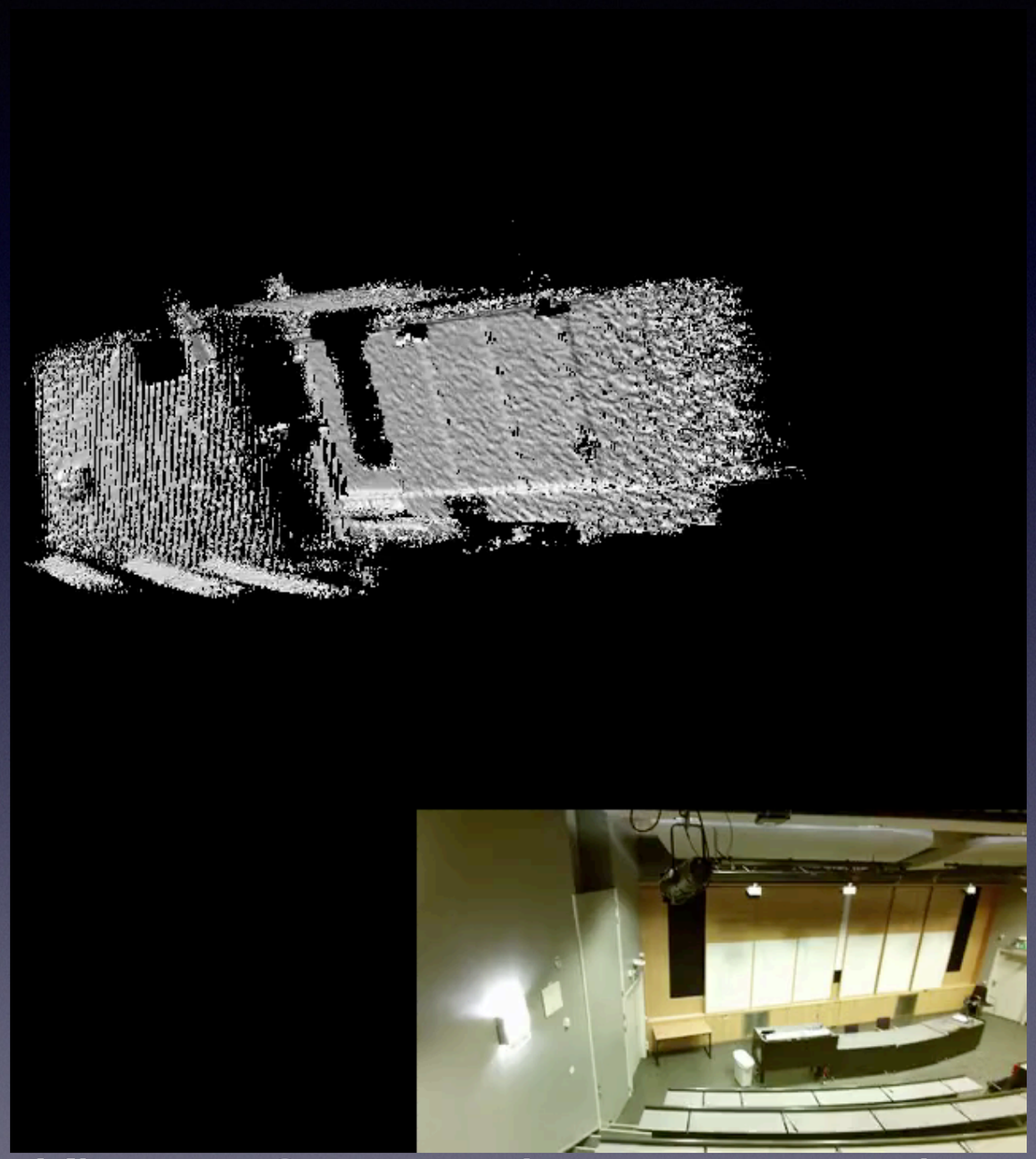
R.A. Newcombe et al., *KinectFusion: Real-Time Dense Surface Mapping and Tracking*, **ISMAR'11**

Based on Truncated Signed Distance Field (TSDF), and ray-casting+ICP to match new scan and model

Point-set registration

KinectFusion algorithm
[Newcombe et al. ISMAR11]

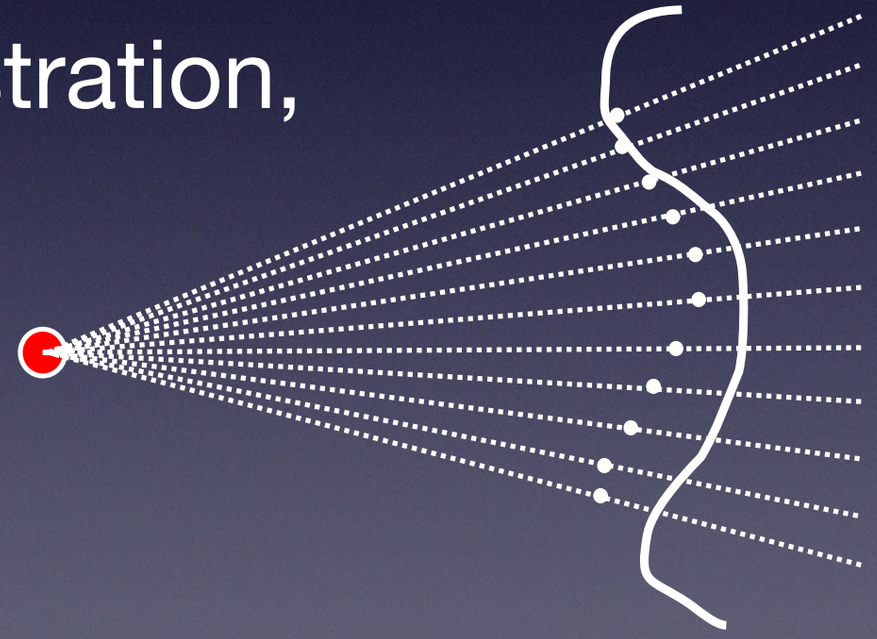
+ Real-time (on GPU)
- Drift
- Limited model size



Kinect v2 example reconstruction

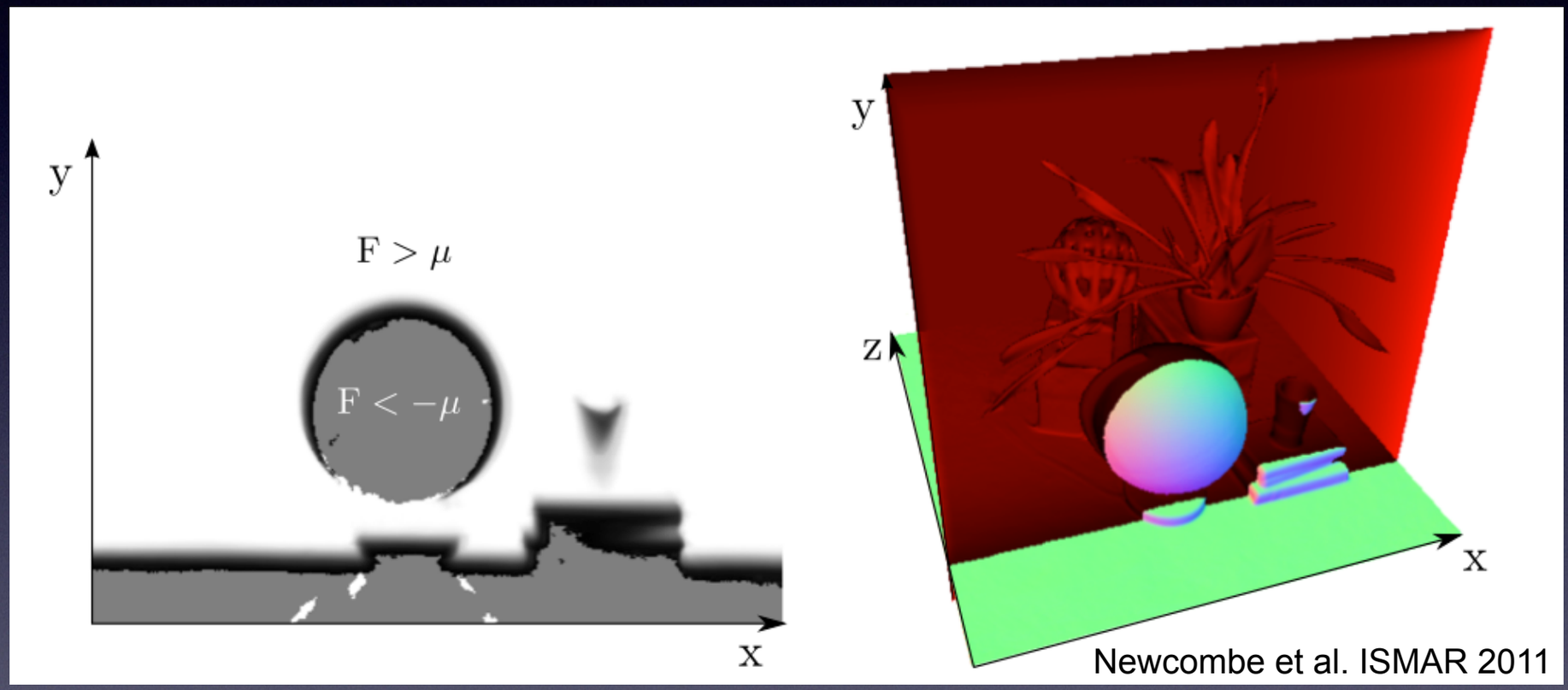
Model to point cloud registration

- The ICP assignments are expensive to compute, in point-cloud to point-cloud registration, as they require finding the nearest neighbour.
- In point-cloud to model registration, we can find approximate near neighbours by ray-casting each pixel onto the model.



TSDF

- Averaging in a voxel volume. **Truncated-Signed-Distance-Field** (TSDF)



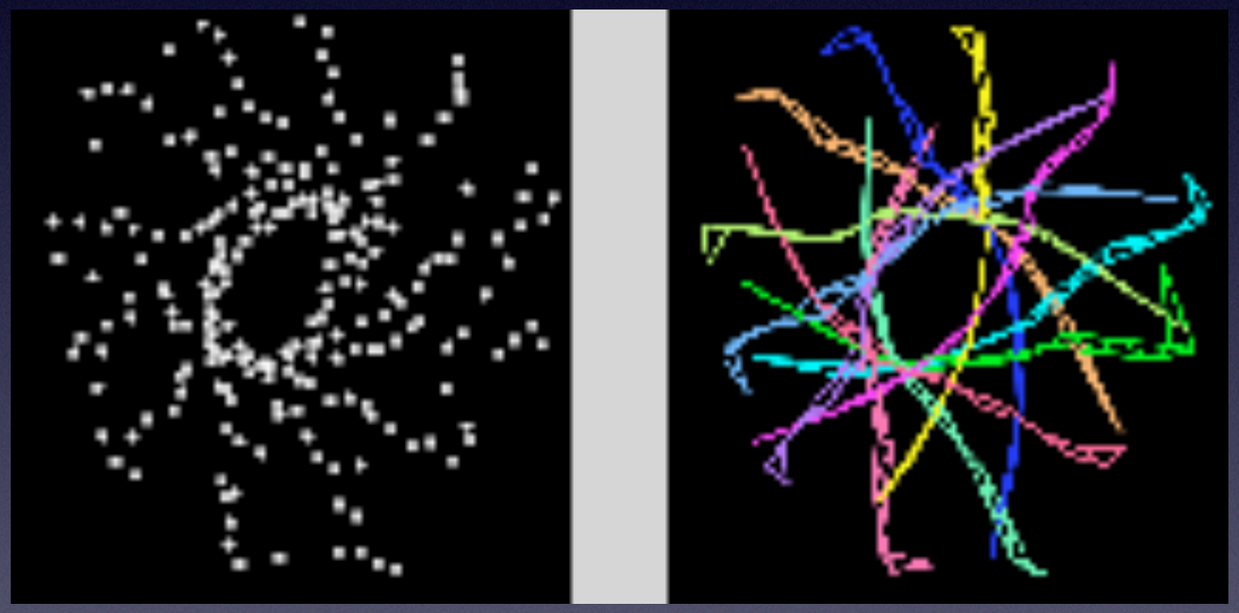
- Surface distance in normal direction $F(\mathbf{p})$, and number of measurements $W(\mathbf{p})$ are stored in each voxel [optional task in TSBB33].

  B. Curless and M. Levoy. *A volumetric method for building complex models from range images*. **SIGGRAPH'96**

Alignment

- Successive 2.5D views are aligned by registration.

Images: Curless and Levoy SIGGRAPH'96

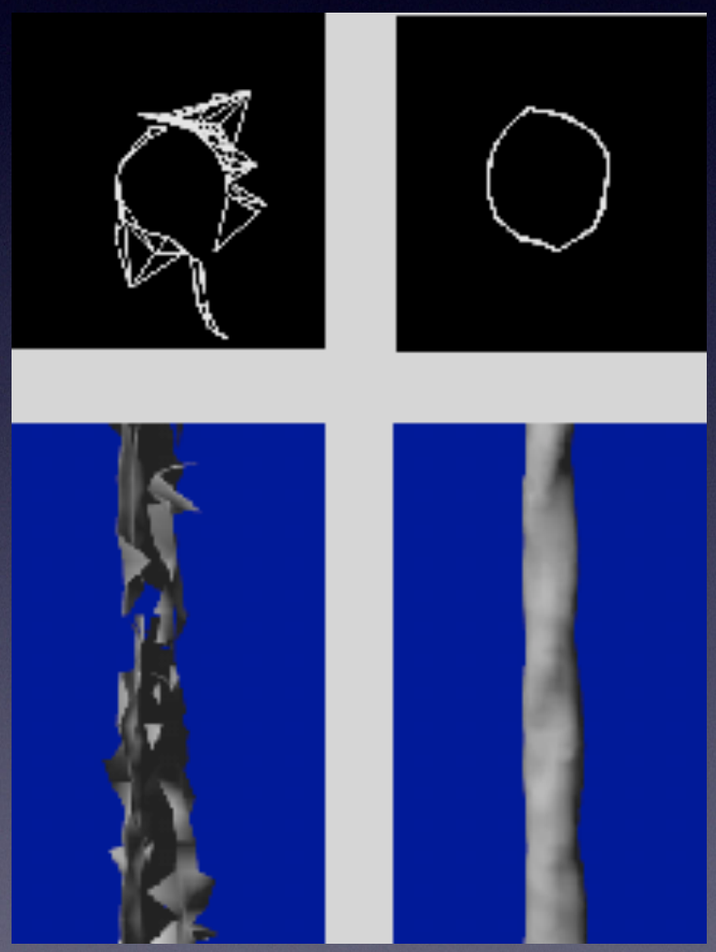


Depth from 12 passive stereo pairs

Alignment

- Successive 2.5D views are aligned by registration.

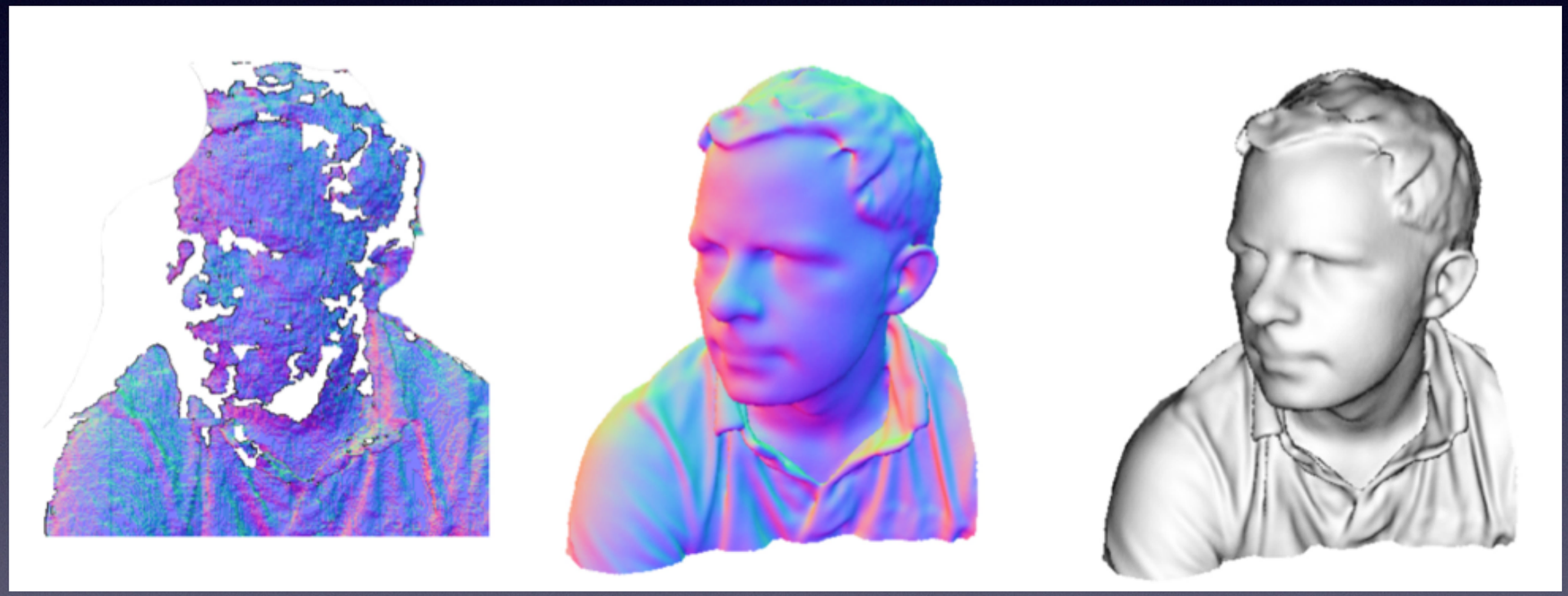
Images: Curless and Levoy SIGGRAPH'96



Mesh-fusion TSDF-fusion

TSDF

- The result of fusion (averaging) in a Truncated Signed Distance Field (TSDF)



Single Kinect frame | Fusion result | Newcombe et al. ISMAR 2011

Feature based registration

- ICP consists of two steps:
  1. Guess correspondences
  2. Align correspondences

Feature based registration

- ICP consists of two steps:
  1. Guess correspondences ⇐ this can be improved!
  2. Align correspondences
- In step 1, proximity of the $\mathbf{p}_i = (X\ Y\ Z)^T$ vectors are used for matching.
- In e.g. TLS scanners and the Kinect, RGB is also available in each point.
- Idea: use proximity in (X,Y,Z,R,G,B)-space instead!
- This could improve matching, but if $X, Y, Z \in [0,130]$ and $R, G, B \in [0,1]$, nothing much changes.

Feature based registration

- Fix for the scale: Balance the different dimensions using a transformation found from a training set $\{\mathbf{f}_i\}_1^N$:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_i \quad \mathbf{C} = \frac{1}{N-1}\sum_{i=1}^{N}\mathbf{f}_i\mathbf{f}_i^T - \mathbf{m}\mathbf{m}^T$$

$$\hat{\mathbf{f}} = \mathbf{C}^{-1/2}\mathbf{f}$$

- Often just the diagonal elements of **C** are used, this is called ***variance normalization***. With a full **C**, the technique is called ***whitening***.
- In general $\mathbf{f}$ can also contain ***local features***, i.e. elements of **f** can be computed from a neighbourhood of the point, e.g. using a CNN.

Feature based registration

- What are local features?

edge

flat surface

corner

peak

…

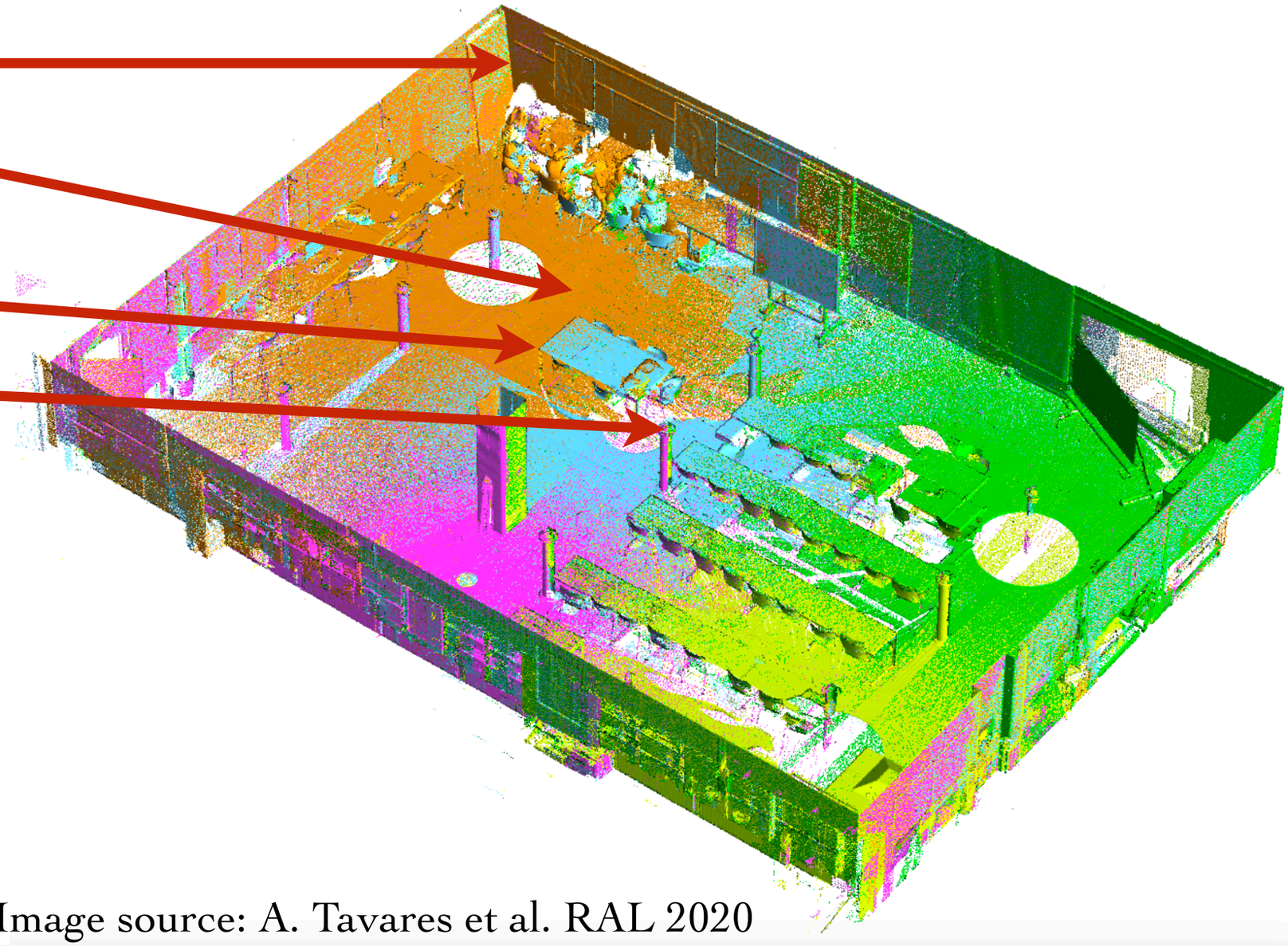
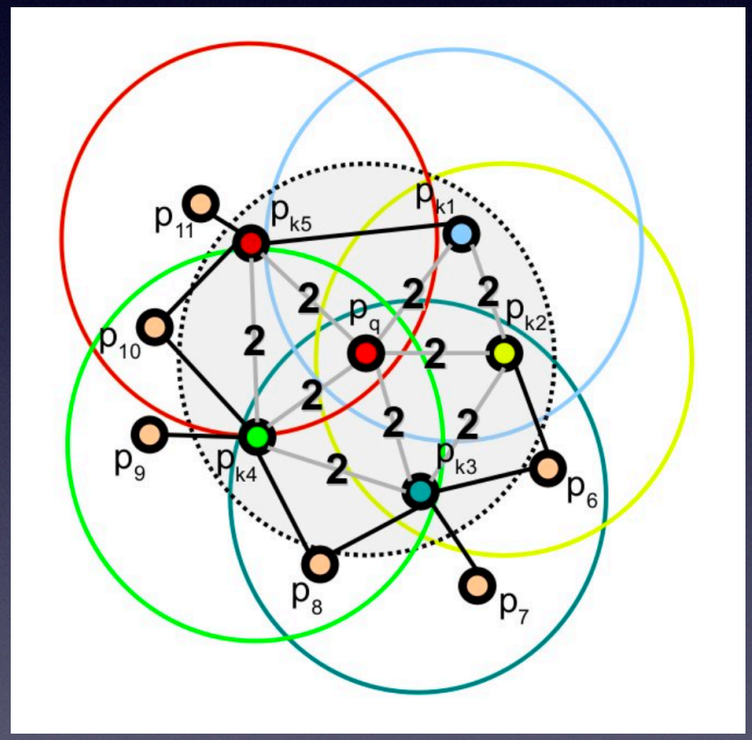


Image source: A. Tavares et al. RAL 2020

Feature based registration

- Local feature example: R. Rusu et al., *Fast Point Feature Histograms (FPFH) for 3D Registration,* **ICRA 2009**

- For each point in a pointcloud $\mathbf{p}_i \in \mathcal{X}$
  1. Compute the surface normal $\mathbf{n}_i$ at $\mathbf{p}_i$
  2. Extract the *k* spatial neighbours to $\mathbf{p}_i$
  3. For each $\mathbf{p}_j \in \mathcal{N}(\mathbf{p}_i, \mathcal{X})$:
     compute three angles, α, ϕ, θ
     based on $\mathbf{p}_i, \mathbf{p}_j, \mathbf{n}_i, \mathbf{n}_j$
  4. Divide ranges of α, ϕ, θ into *q* bins, and compute $\mathbf{f}_i$ as 3 stacked histograms over the **k** neighbours
  5. Finally update $\mathbf{f}_i$ by computing a weighted average of the neghbours' histograms:



$$\mathbf{f}_i^* = \mathbf{f}_i + \frac{1}{k} \sum_{k \in \mathcal{N}(\mathbf{p}_i)} \frac{1}{d(\mathbf{p}_i, \mathbf{p}_k)} \mathbf{f}_k$$

Feature based registration

- Note #1: the point position is not part of FPFH.
- Note #2: one feature vector per point in the point cloud is computed, so instead of 3N values, we now have (3+3q)N (q #bins in each histogram, e.g. 5 or 9)
- Note #3: It is still beneficial to do whitening/variance normalization, but not critical, and it is often omitted.

- It is also possible to use machine learning to find useful local features. This can be both faster, and lead to better features. Such feature learning is based on ***contrastive learning*** [see TSBB19] or registration loss learning (more on this later).

Fast Global Registration (FGR)

FGR is a popular method that uses local features to improve correspondences, and **robust error norms** [See TSBB33] and graduated non-convexity to improve the search for the alignment transformation.

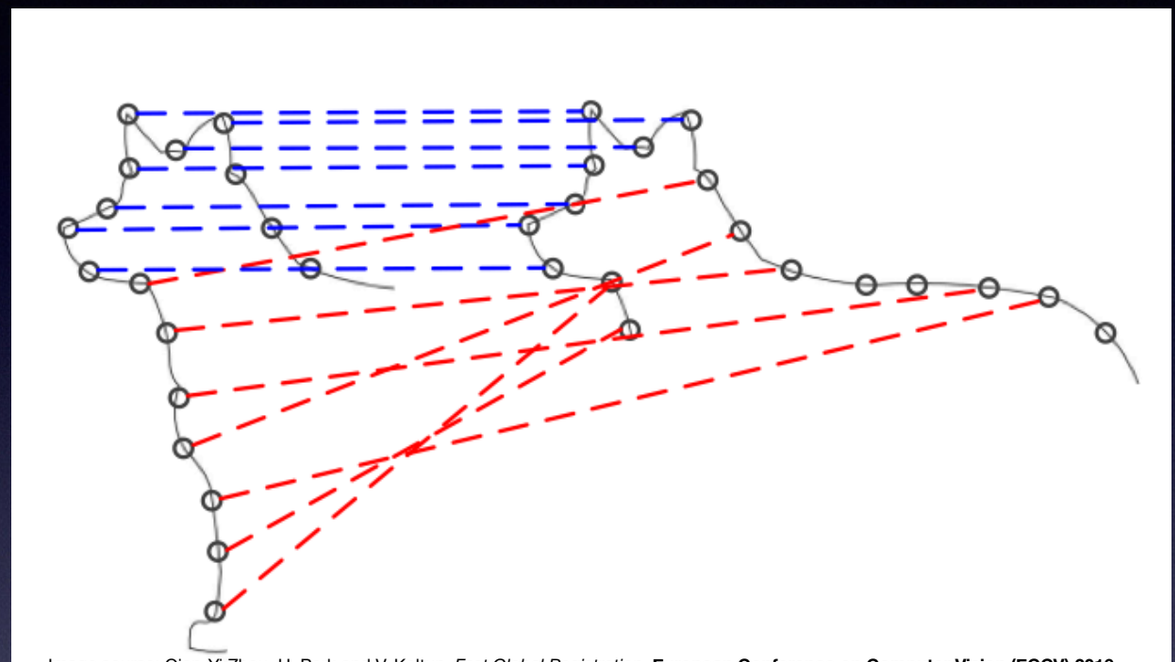


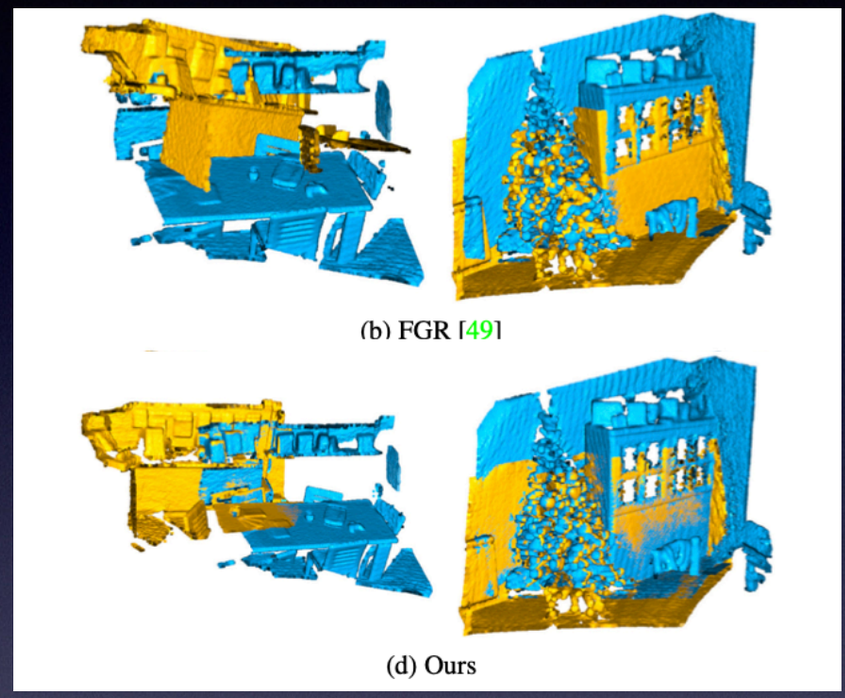
Image source: Qian-Yi Zhou, H. Park and V. Koltun, *Fast Global Registration*, **European Conference on Computer Vision (ECCV) 2016**

Qian-Yi Zhou, H. Park and V. Koltun, *Fast Global Registration*, **European Conference on Computer Vision (ECCV) 2016**

- After feature based matching, ICP is run as a fine-tuning step to improve accuracy.

FGR and DGR

- FGR was later extended to Deep Global Registration (DGR). In this method, Weighted Procrustes was used in the optimization instead. This allows the use of end-to-end learning (backpropagation) to find good features.


(b) FGR [49]

(d) Ours

- C. Choy, W. Dong and V. Koltun, *Deep Global Registration*, **International conference on Computer Vision and Pattern Recognition (CVPR) 2020**

- DGR also uses ICP as a fine-tuning step after the deep feature matching step.

Multiple point set registration

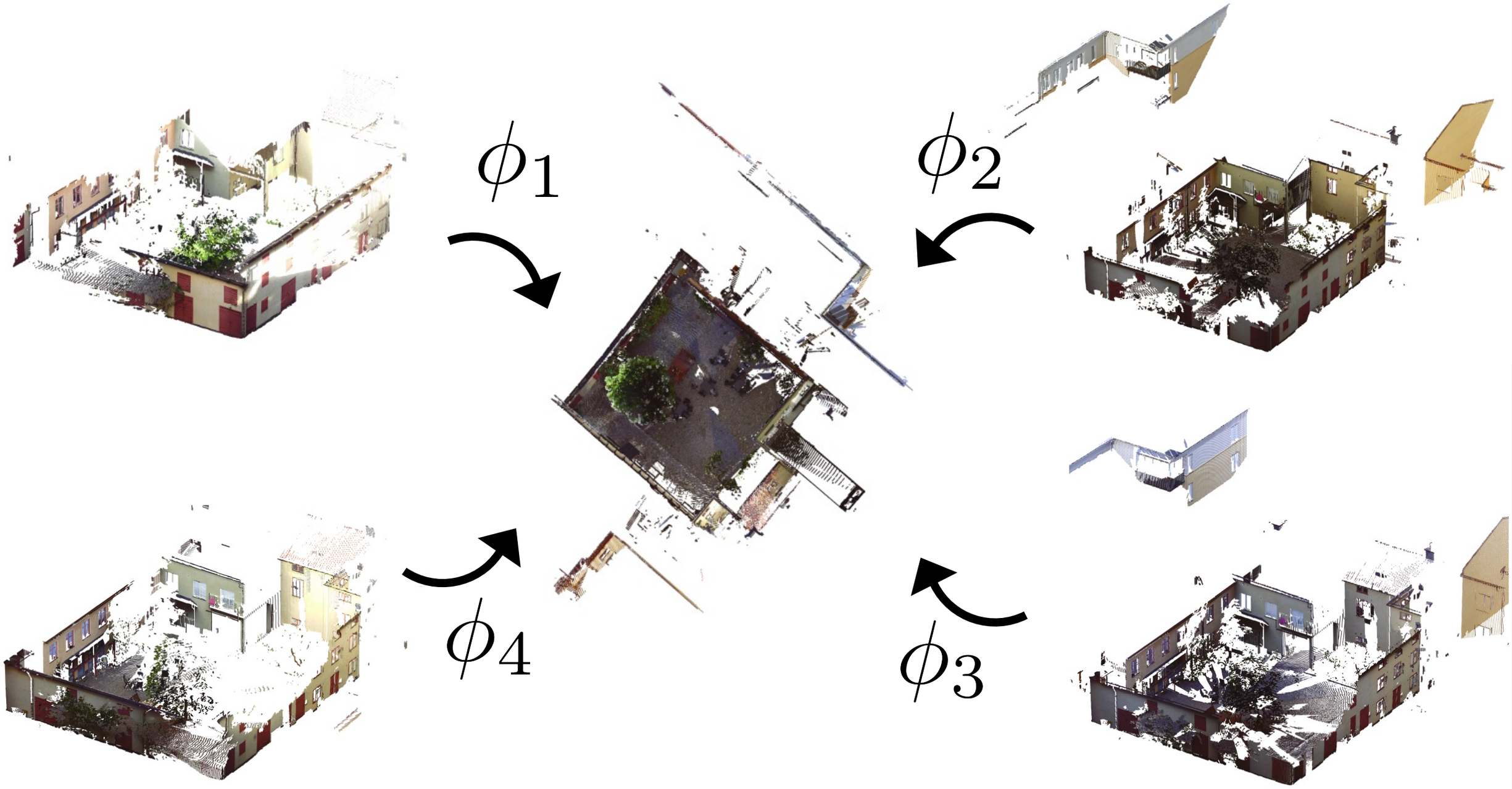


Illustration by Martin Danelljan

Multiple point set registration

- Multiple point set registration can reduce drift compared to adding one point cloud at a time.

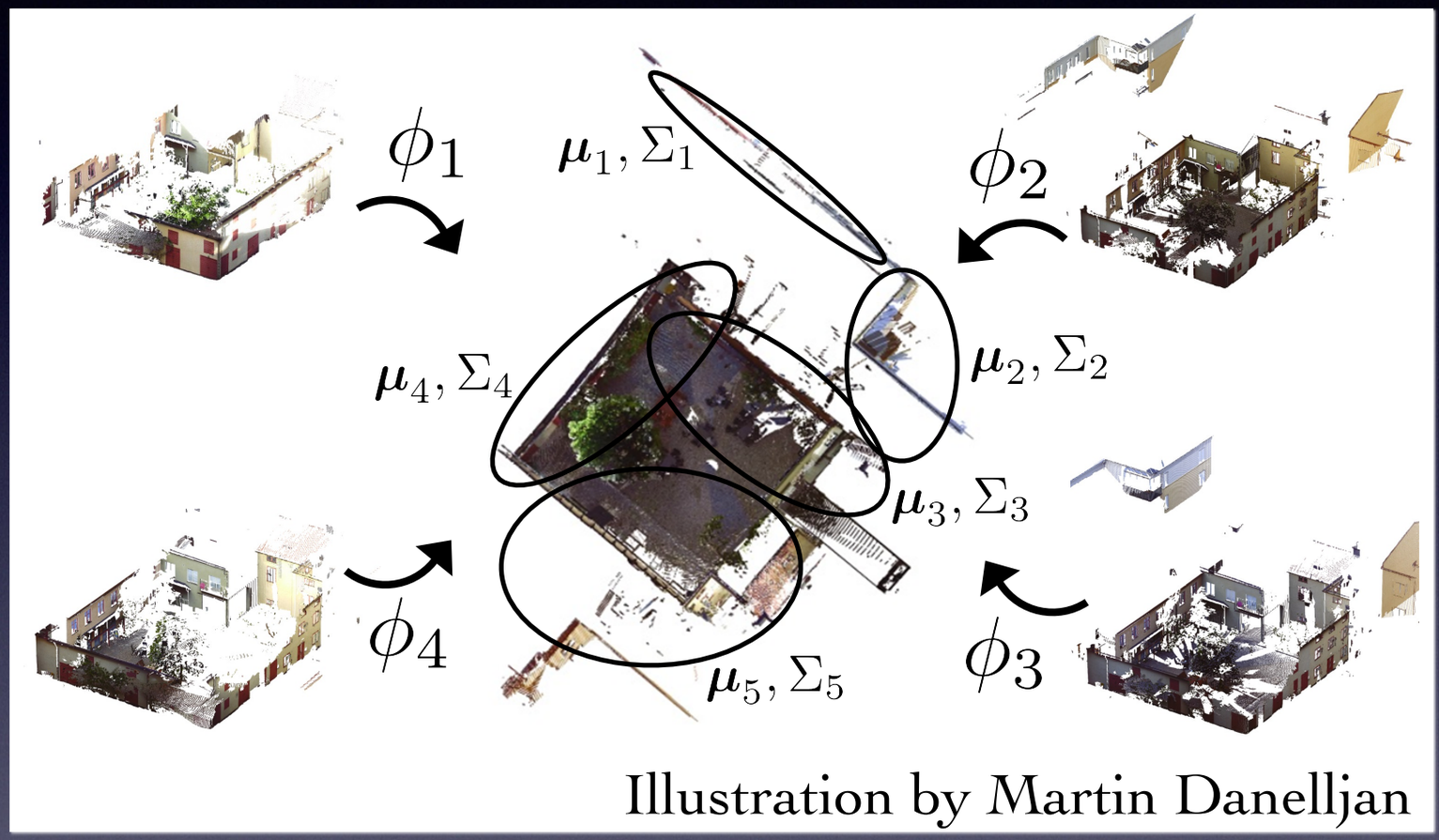
- The complexity of ICP is $O(NM^K)$

  N - #iterations
  M - #points in point-clouds
  K - #point clouds (ICP has K=2)

- If we instead match to a central representation we get $O(NMK)$

JRMPS (Joint Registration of Multiple Point Sets)

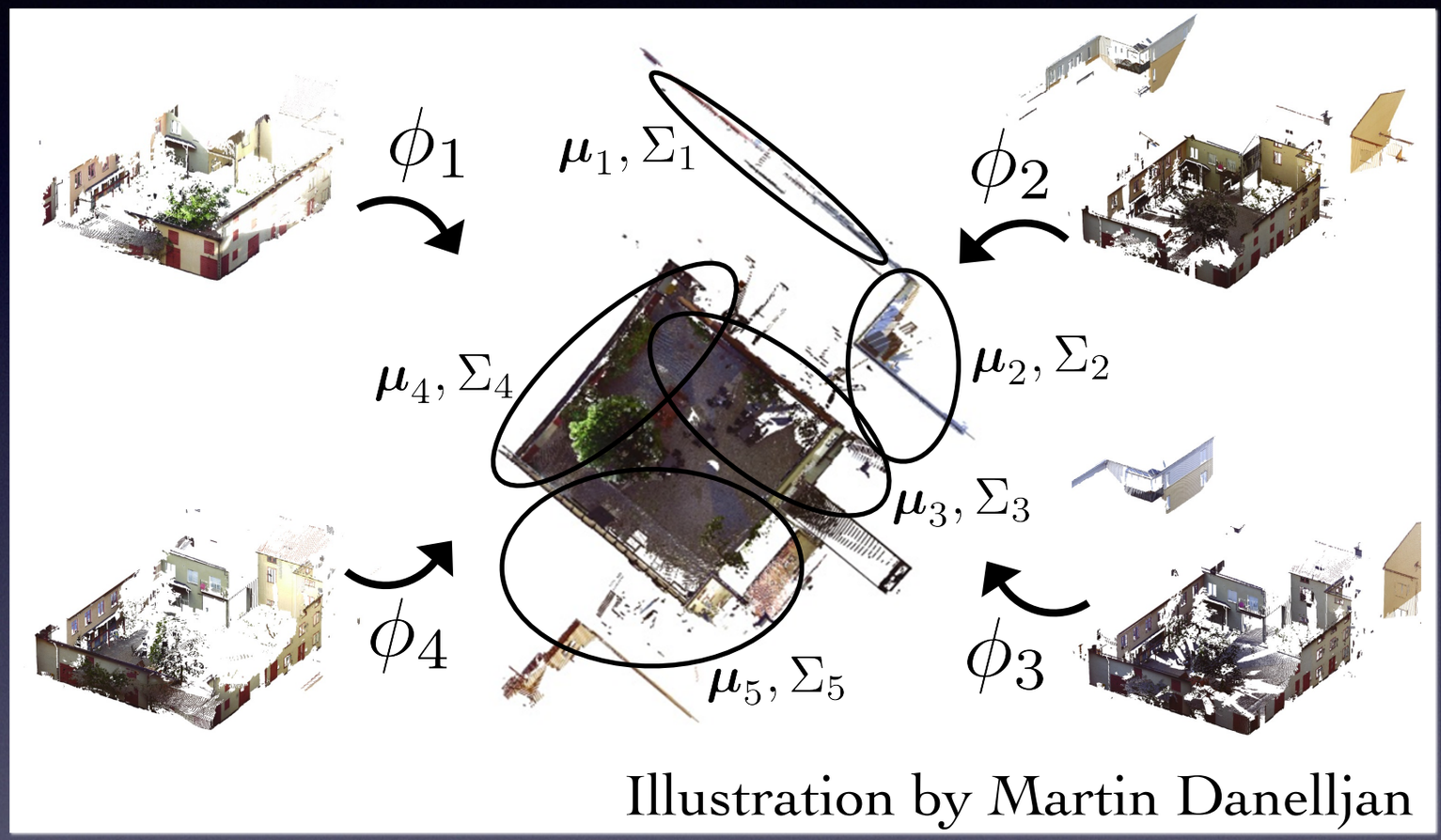
- The joint point cloud is represented as a **Gaussian Mixture Model** (GMM) (a set of Gaussians)



$$p(\mathbf{p}) \approx \sum_k g(\mathbf{p}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

JRMPS (Joint Registration of Multiple Point Sets)

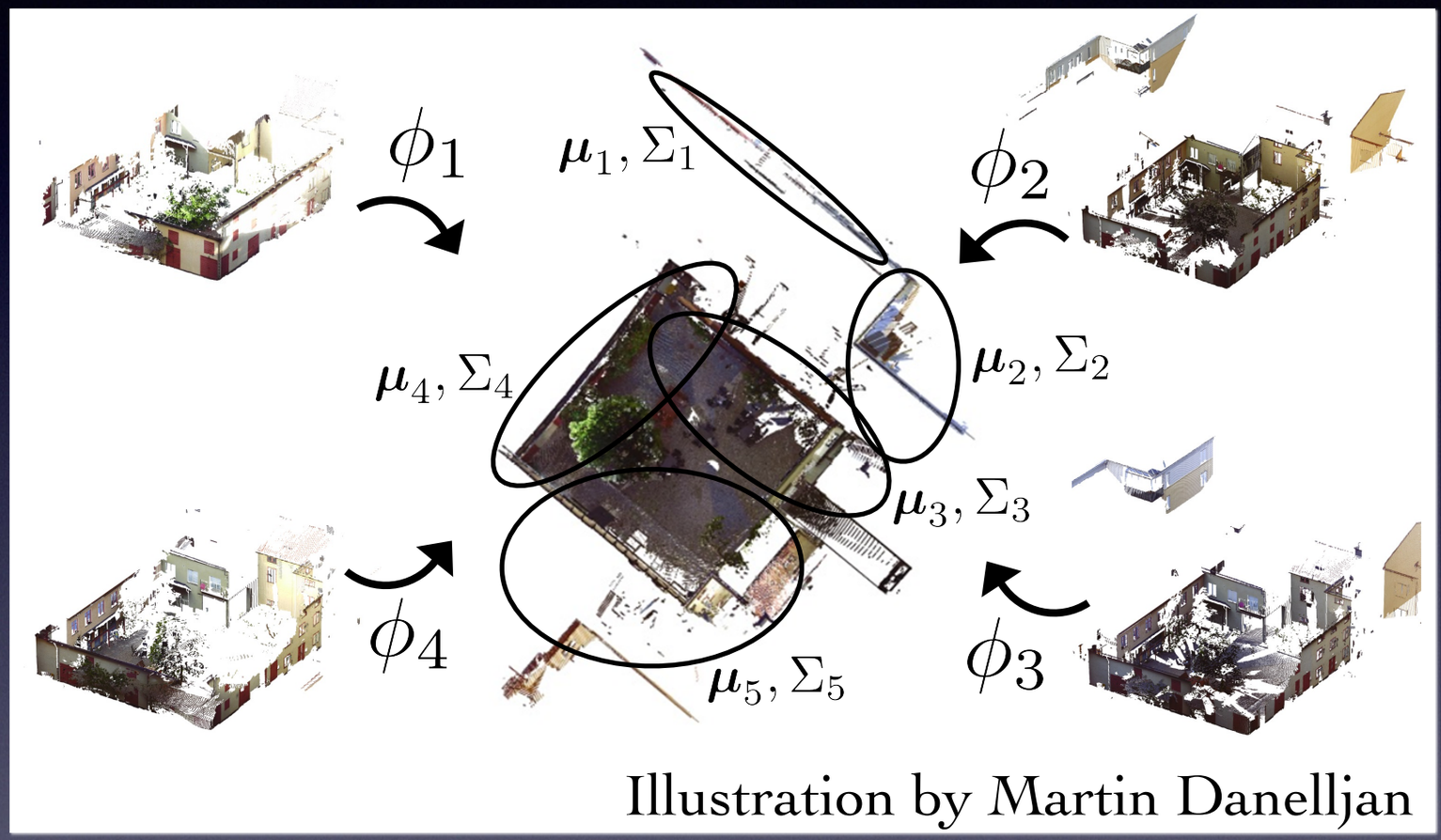
- The joint point cloud is represented as a **Gaussian Mixture Model** (GMM) (a set of Gaussians)
- Matching is done between each point cloud and GMM.



$$p(\mathbf{p}) \approx \sum_k g(\mathbf{p}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

JRMPS (Joint Registration of Multiple Point Sets)

- The joint point cloud is represented as a **Gaussian Mixture Model** (GMM) (a set of Gaussians)
- Matching is done between each point cloud and GMM.



- Advantages:
  - + Multiple point sets are registered in parallel
  - + Symmetric error distribution
  - + Linear complexity in the number of points.

JRMPS (Joint Registration of Multiple Point Sets)

- Joint ML estimation of relative poses and point cloud mixture using ***expectation conditional maximization*** (ECM):

1. Update GMM
2. Update membership weights
3. Update transformations
4. Go to 1.

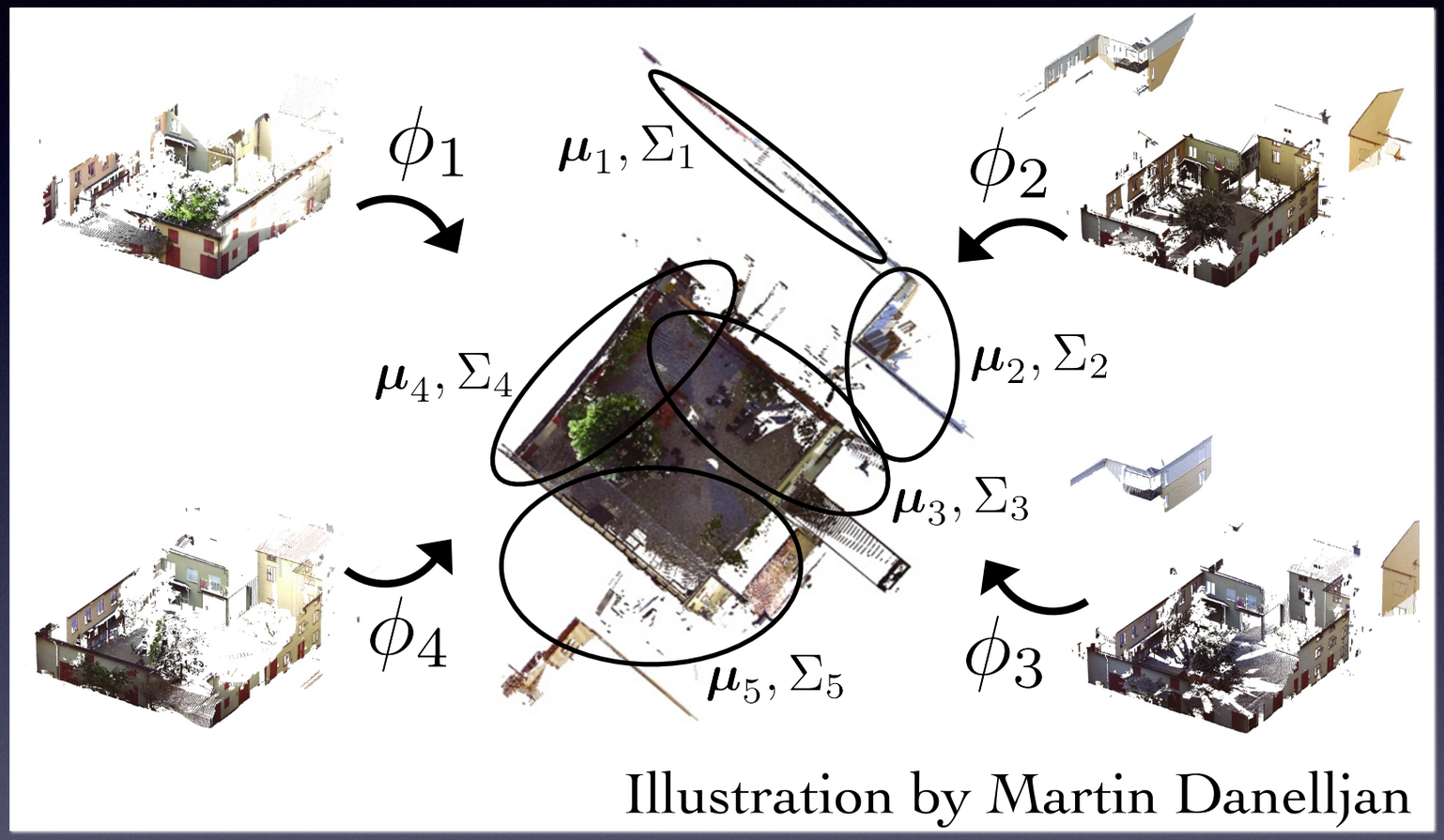
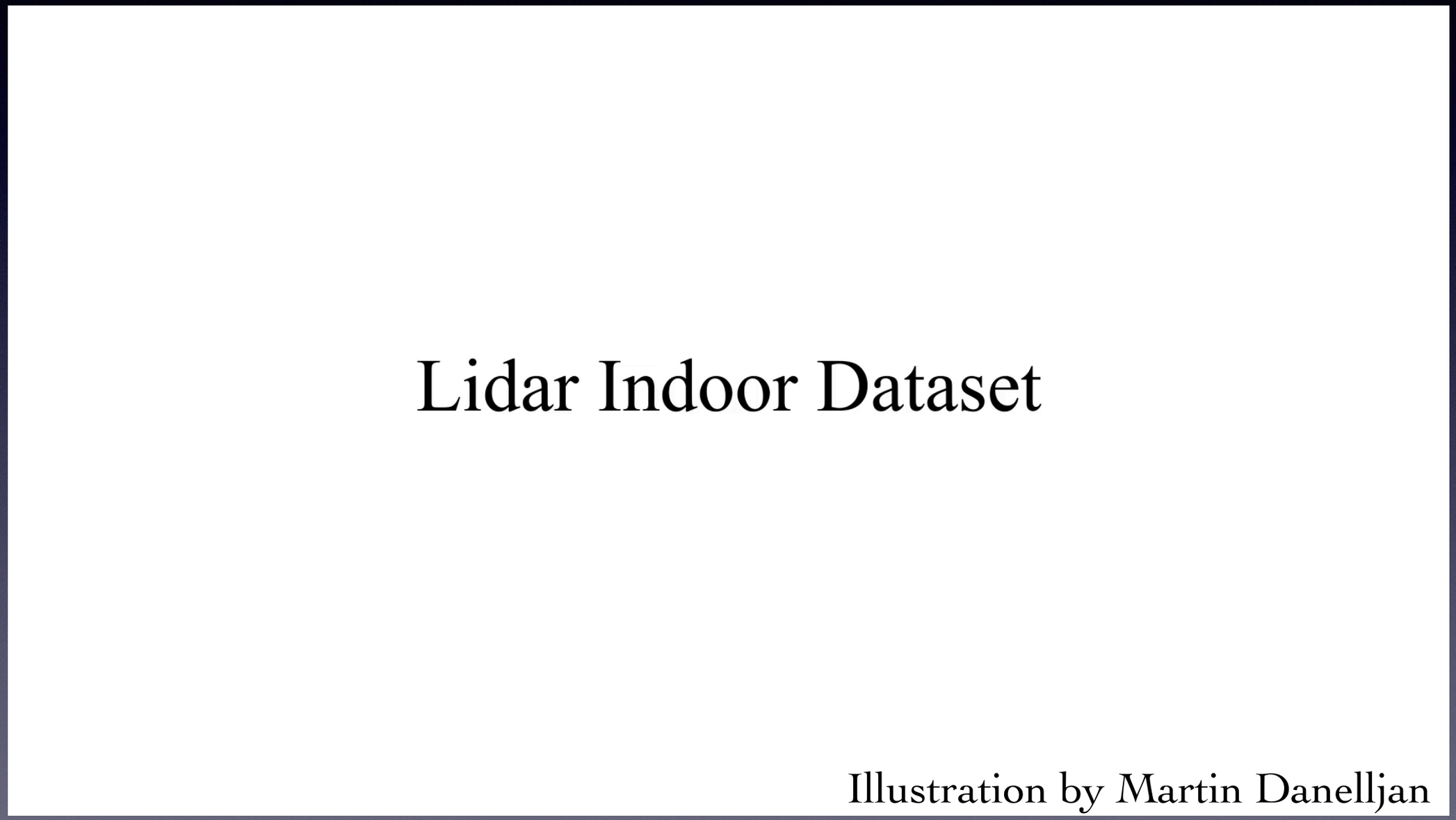


Illustration by Martin Danelljan

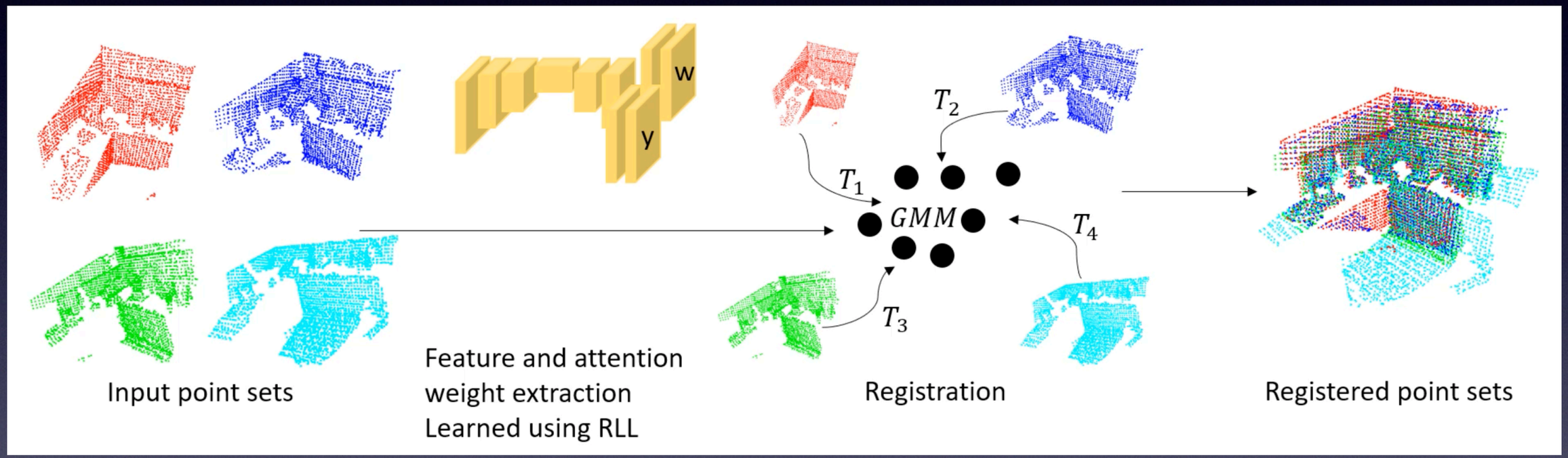
- R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, *Rigid and articulated point registration with expectation conditional maximization*, **PAMI, vol. 33, no. 3, pp. 587–602, 2011**

JRMPS with colour



JRMPS with deep learning

- Registration Loss Learning (RLL)

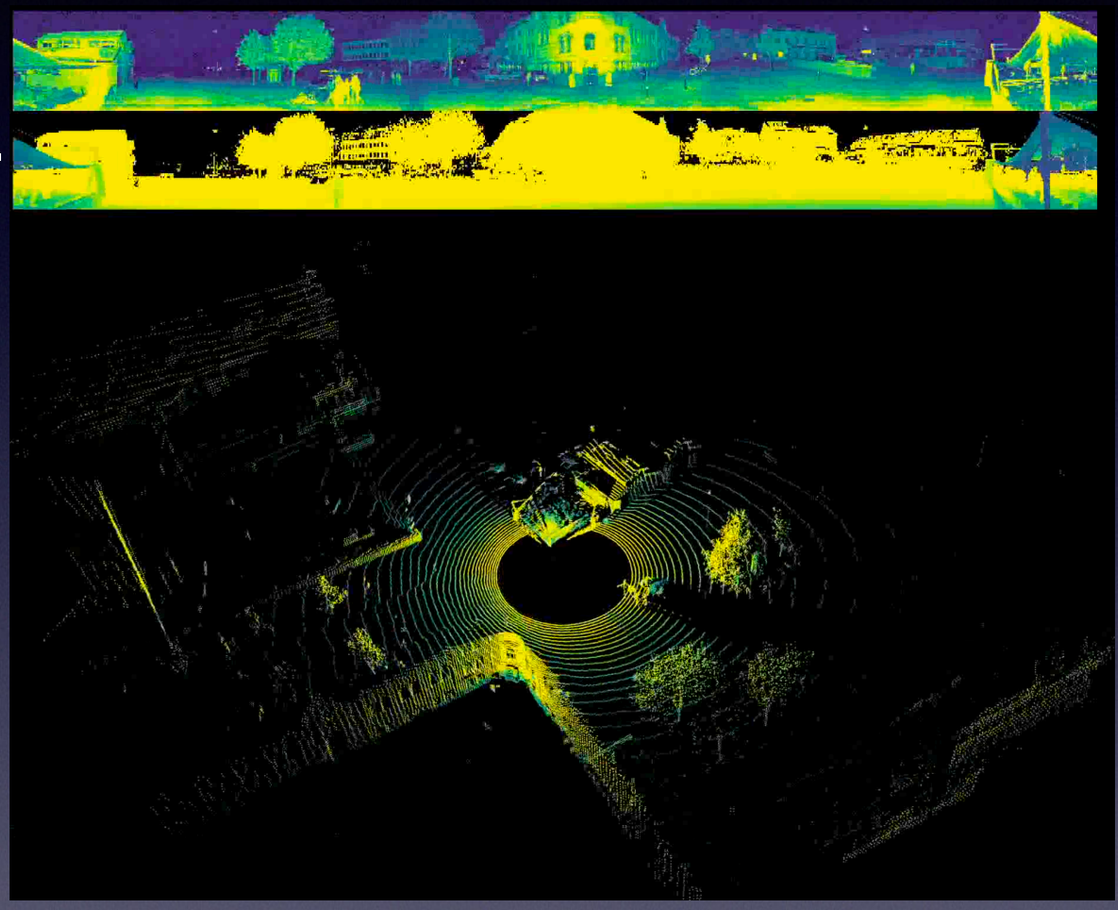


- Joint Gaussian Mixture in 3D and feature space. Learn feature space by back-propagating through the registration loss.

- Felix Järemo Lawin and Per-Erik Forssén, *Registration Loss Learning for Deep Probabilistic Point Set Registration,* **International Conference on 3D Vision (3DV) 2020**

Farthest Point Sampling (FPS)

- Point clouds from a spinning Lidar always have a non-uniform density. Many more points closer to the sensor.
- Point set registration speed is proportional to the number of points.
- Idea: Resample point sets to have more uniform density before registration.



- Y. Eldar et al. *The Farthest Point Strategy for Progressive Image Sampling*, **ICPR'94**

Note: FPS was originally defined for edge maps in 2D images, but nowadays mainly used on point clouds.

Farthest Point Sampling (FPS)

FPS Algorithm

Draws a subset of N farthest points from a set $\mathcal{X}$.

1. Draw a random point $\mathbf{p}_1 \in \mathcal{X}$
   move it to another set $\mathcal{S} = \{\mathbf{p}_1\}$, $\mathcal{X} = \mathcal{X} - \mathbf{p}_1$
2. Draw another point $\mathbf{p}_k \in \mathcal{X}$
   that is as far as possible from all previously drawn points
   $$\mathbf{p}_k^* = \arg \max_{\mathbf{p}_k \in \mathcal{X}} \min_{\mathbf{p}_l \in \mathcal{S}} d(\mathbf{p}_k, \mathbf{p}_l)$$
3. If $|\mathcal{S}| < N$, i.e. number of points drawn is less than required, go to 2.

Farthest Point Sampling (FPS)

[Show FPS demo]

Farthest Point Sampling (FPS)

+ Speeds up registration as there are fewer points to consider

+ As e.g. ICP minimizes the sum of registration errors, FPS also reduces locking effects where only the central circles of a scan are matched.

- May degrade accuracy if *too coarse* downsampling (N is problem dependent)

Farthest Point Sampling (FPS)

FPS is also used in Deep Learning on point clouds, e.g.

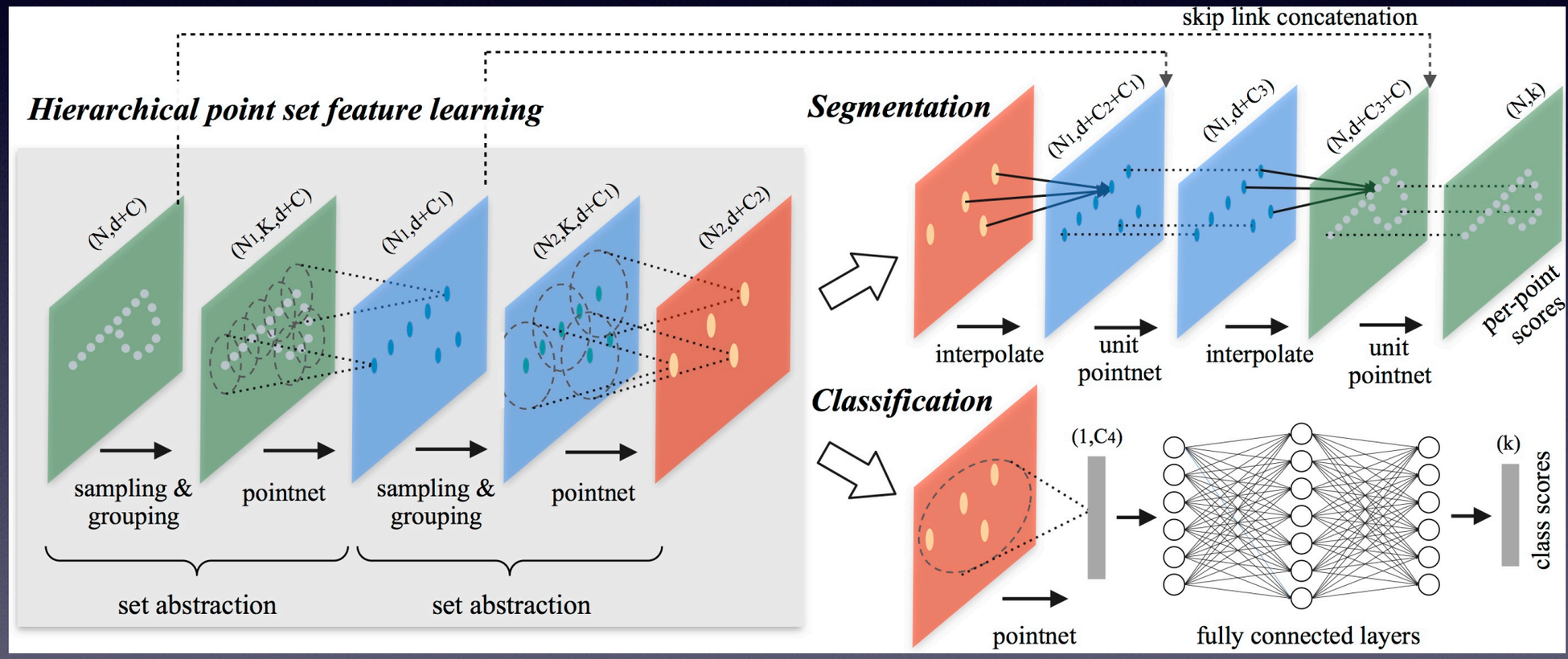


Image source: C. R. Qi et al., PointNet++: *Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, **NeurIPS'17**

C. R. Qi et al., PointNet++: *Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, **NeurIPS'17**

Farthest Point Sampling (FPS)

Alternatives to FPS:

1. Instead of FPS, inverse density weighting can be used, and keeping all samples

2. Alternatively, attention weights can be learned, as in RLL.

3. With random subsampling in inverse proportion to local density, the result will also be similar. (but much faster).

4. Voxel downsampling.

Summary

- ***Point-set registration*** (PSR) is used to create large 3D models from 3D cameras.
- ***Truncated Signed Distance Fields*** (TSDF) allow real-time fusion
- PSR failures can be reduced by also matching ***local features***
- PSR drift can be reduced using ***joint registration of multiple point sets*** (JRMPS)
- ***farthest point sampling*** (FPS) can subsample point clouds to speed up matching and reduce locking effects.
- Important PSR methods: ICP, TSDF/KinectFusion, FGR, JRMPS, and PSR with learned features e.g. RLL and DGR.