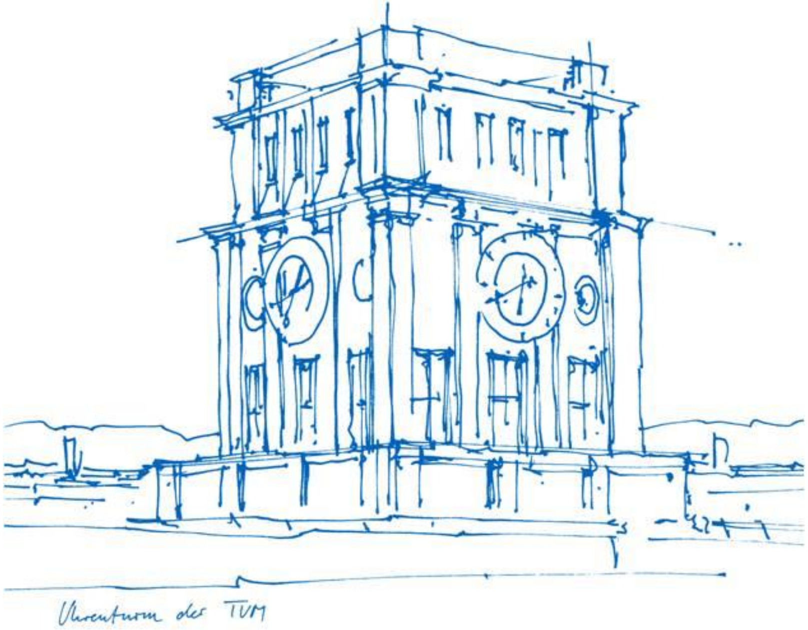
Model globally, match locally: Efficient and robust 3D object recognition

Bertram Drost, Markus Ulrich, Nassir Navab, Slobodan Ilic

Julian Ost

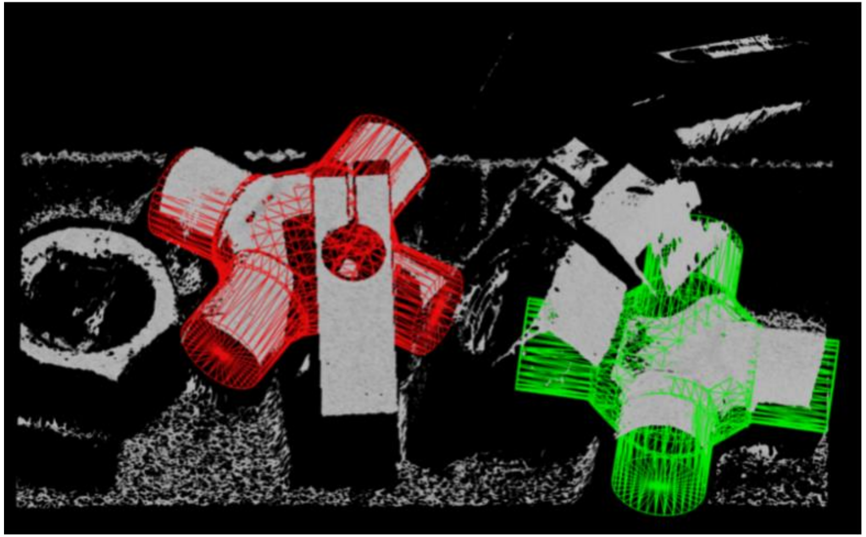
Technische Universität München

Garching, 25. June 2019



Free-Form 3D Object Recognition

Example of two partly occluded instances of an object in a 3D point cloud



Global Approach

Only detect standard shapes e.g. planes, cylinders and spheres;

Or require segmentation of the scene e.g. recover pose from primitives.

Recover 3D pose with 6 degrees of freedom.

- No free form 3D objects
- Low precision
- Computationally expensive

Wahl et al. introduced “surflets” a two-point feature (similar to this method)

Local Approach

1. Identification of possible point to point correspondences of model and scene
2. Grouping correspondences and recover pose

Point descriptors: Describe the surface around a point

Quite efficient, but:

- Depends on local surface information (e.g. clutter, occlusion, ...)
- Is related to data quality and resolution (e.g. noise)

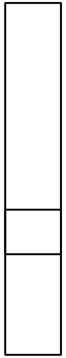
Method Overview

Combining global feature based model description and a local matching

Point Pair Feature

$F_s(\mathbf{s}_r, \mathbf{s}_i)$

Global Model Description



$A = \{(\mathbf{m}_1, \mathbf{m}_2), (\mathbf{m}_3, \mathbf{m}_4), (\mathbf{m}_5, \mathbf{m}_6)\}$

Hash Table

Local Matching

$(\mathbf{s}_r, \mathbf{s}_i)$ and $(\mathbf{m}_r, \mathbf{m}_i)$

$(\mathbf{s}_r, \mathbf{s}_i)$ and $(\mathbf{m}'_r, \mathbf{m}'_i)$

…

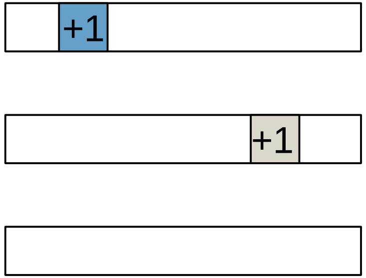
$$\mathbf{s}_i = (T_{s\to g})^{-1} R_x(\alpha) T_{m\to g} \mathbf{m}_i$$

Voting Scheme for local Coordinates

α_1 α_n

$\mathbf{m}_r$

$\mathbf{m}'_r$



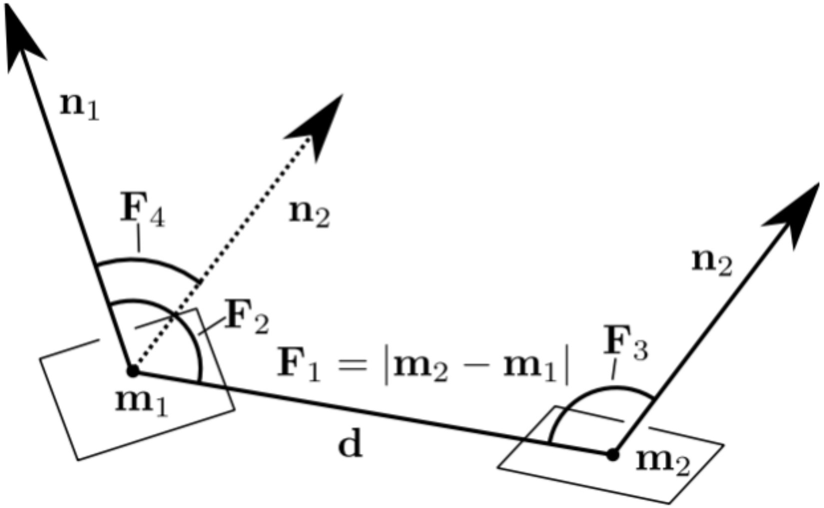
Point Pair Feature

$\mathbf{F}(\mathbf{m}_1, \mathbf{m}_2) = (|\mathbf{d}|_2, \angle(\mathbf{n}_1, \mathbf{d}), \angle(\mathbf{n}_2, \mathbf{d}), \angle(\mathbf{n}_1, \mathbf{n}_2))$

Describes the relative position and orientation of two orientated points

Asymmetric property guarantees uniqueness for sequence of points

Offline: Creating the global model description
Online: Finding the object in the scene



Global Model Description

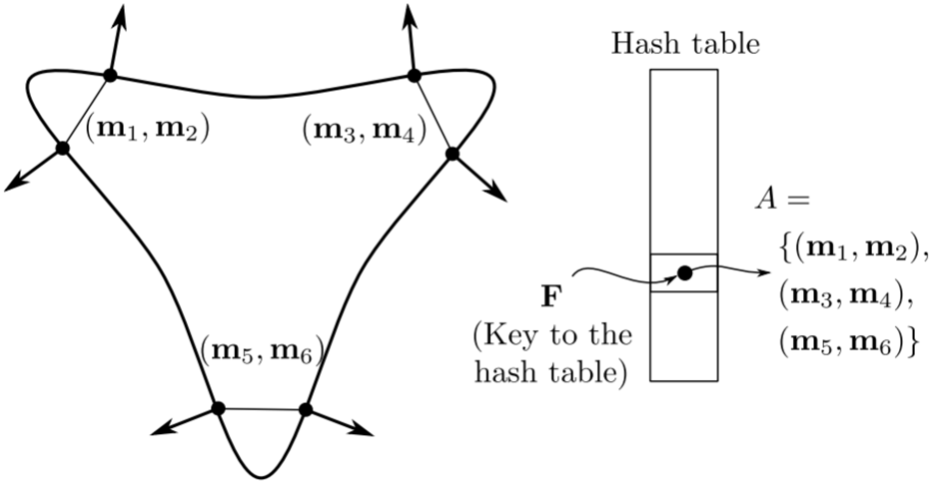
Representation as a set of point pairs ($\mathbf{m}_i$, $\mathbf{m}_j$)

Point pair features **F** for all point pairs on the model surface M

Discretization of feature vectors:

- $d_{dist} = \tau_d \cdot \mathrm{diam}(M)$
- $d_{angle} = 2\pi / n_{angle}$

Equal discrete vectors are grouped in hash table indexed by the feature



Local coordinates

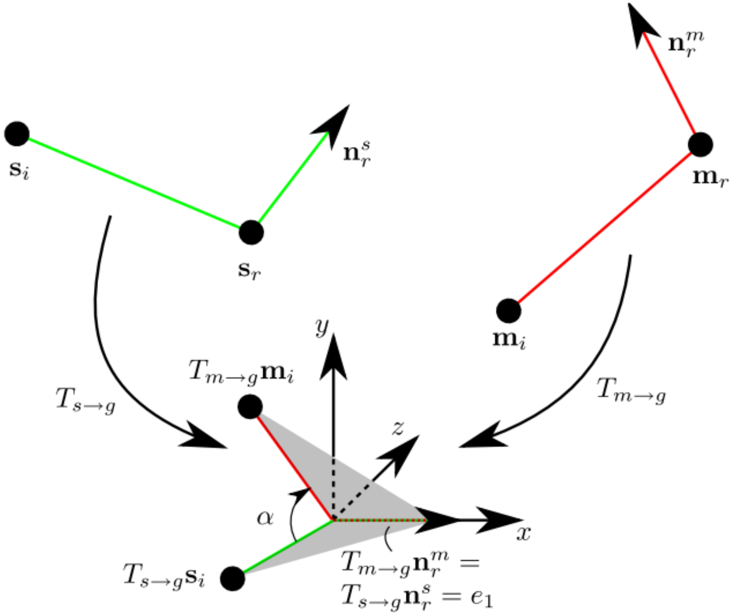
Assumption: Any arbitrary $\mathbf{s}_r$ on the object from the scene corresponds to a point $\mathbf{m}_r$ on the model

Matching model and scene - **local coordinates**:

$\mathbf{m}_r$, α (reduces the problem to three dimension)

Alignment of ($\mathbf{s}_r$, $\mathbf{s}_i$) and ($\mathbf{m}_r$, $\mathbf{m}_i$) with similar **F**

$$\mathbf{s}_i = (T_{s\to g})^{-1} R_x(\alpha) T_{m\to g} \mathbf{m}_i$$



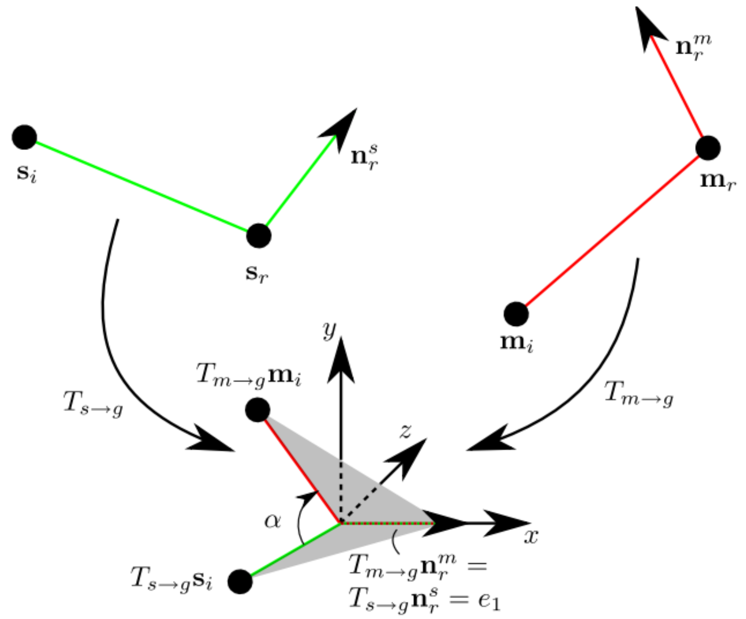
Efficient Looping

$\alpha = \alpha_m - \alpha_s$

α_m can be pre calculated off line

α_s needs to be calculated only once for every ($\mathbf{s}_r$, $\mathbf{s}_i$)

$\mathbf{t} = R_x(\alpha_s)T_{s\rightarrow g}\mathbf{s}_i$
$= R_x(\alpha_m)T_{m\rightarrow g}\mathbf{m}_i \in\in R\mathbf{x} + R_0^+\mathbf{y}$

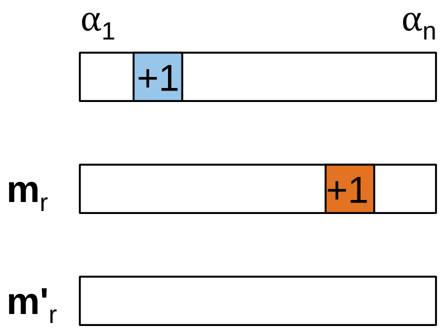


Voting Scheme

Find the best local coordinates for a given $\mathbf{s}_r$

- $\mathbf{F}_s(\mathbf{s}_r, \mathbf{s}_i)$ for every point pair as key to the hash table
- Local coordinates $\mathbf{m}_r$, α of every match $(\mathbf{m}_r, \mathbf{m}_i)$
- Voting Scheme on two dimensional array representing the discrete space for a fixed $\mathbf{s}_r$

→ Peaks in the accumulator array: **Optimal local coordinate**



Pose Clustering

Filter incorrect poses (e.g. $\mathbf{s}_r$ not on object) and increase accuracy

- Optimal poses for multiple $\mathbf{s}_r$ are clustered given translational and rotational threshold
- Per Cluster: Votes are accumulated and poses averaged
- Return (multiple) clusters with the highest score for (multiple) objects in the scene

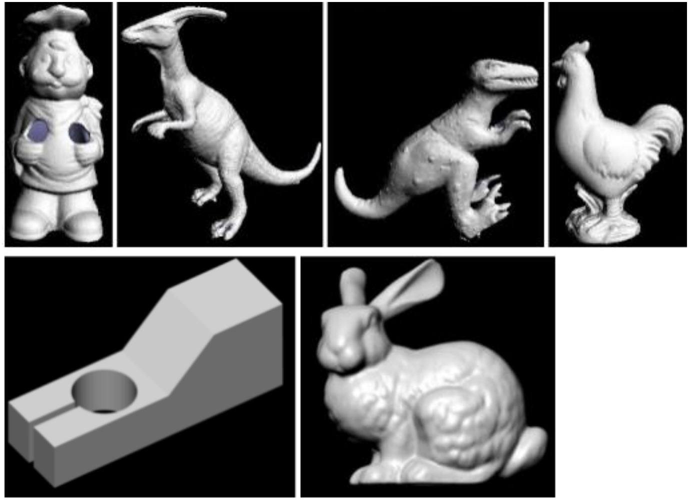
→ Removes isolated poses with low scores and increases acc. by averaging poses

Evaluation

Test performance, efficiency and dependence on parameters:

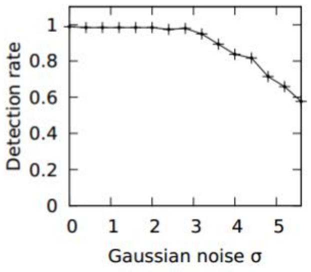
- τ_d
- n_{angle}
- number of reference points (as percentage of subsampled scene points)

Evaluation against synthetic and real Data



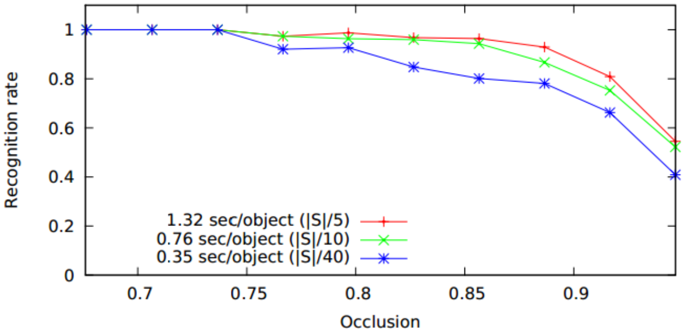
Result Synthetic Datasets

1. Single Object and gaussian noise
   - Distance and angle threshold for detection rate



1. Four to nine randomly placed objects
   - w.r.t. to occlusion
   - and Number of reference points in the scene

→ Trade Off between speed and performance



Real Data - Quantitative

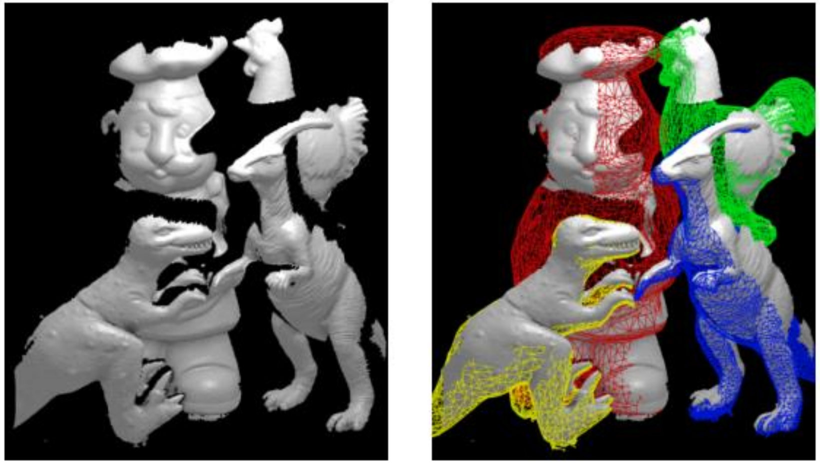
Evaluated on 50 scenes by Mian et al.

Evaluated against:

- *Tensor Matching*: Multidimensional table representations of the model (Mian et al.)
- *Spin images*: Match surfaces represented as surface meshes (Johnson and Herbert)

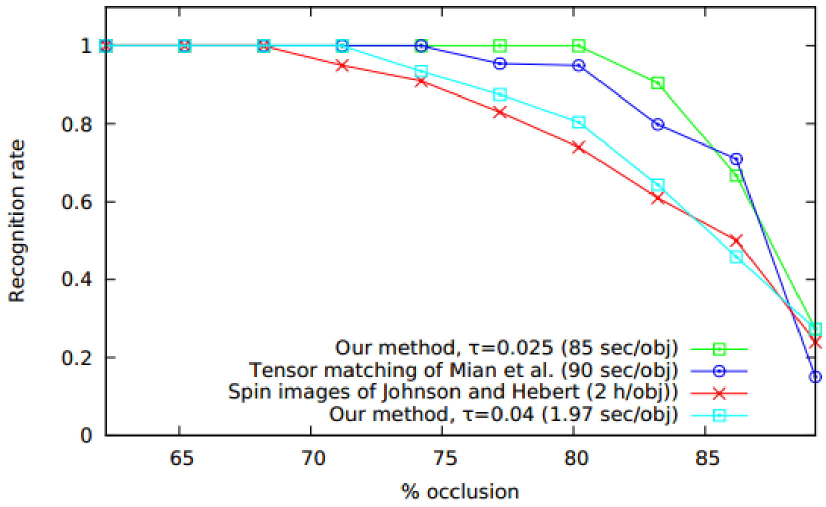
Variation of the sampling rate τ_d

Evaluated w.r.t. clutter and occlusion



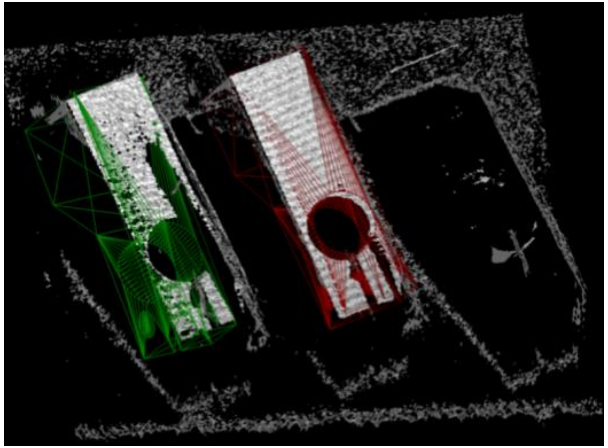
Real Data - Quantitative

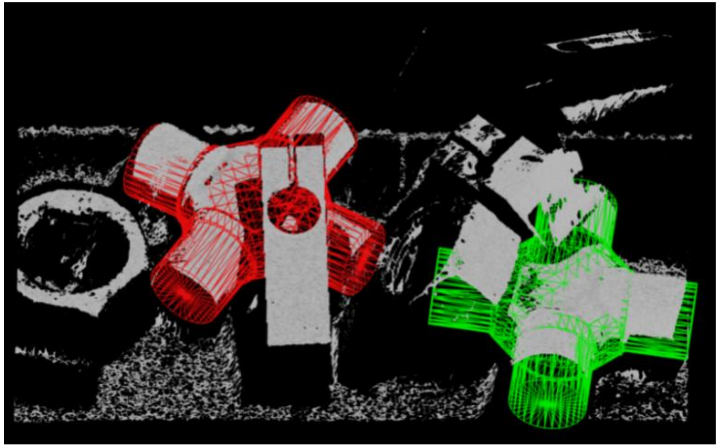
- Slightly increased the recognition rate at same speed
- For lower sampling rate: A bit worse recognition rate but 40 times faster matching
- Main Advantage of this method:
  Trade off between speed and recognition rate



Real Data - Qualitative

Experiments on self-build laser scanning setup showed accurate enough recognition for object manipulation despite a lot of clutter and occlusion





Conclusion

- Efficient, stable and accurate method to find free-form 3D objects in point clouds
- Independent from local surface information
- Very fast matching through locally reduced search space
- Better recognition rate in comparison to traditional approaches

- C++ implementation and parallelization could speed up matching times
- Refinement of poses with e.g. ICP could increase detection rate